KB-74
Applied Data Science

Math behind Machine Learning 4

Gradient descent, derivative, learning rate, update rules, batch gradient descent

THE HAGUE
UNIVERSITY OF
APPLIED SCIENCES
FACULTY IT & DESIGN

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

Parameters: θ_0, θ_1

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

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Idea behind gradient descent

Have some function $J(\theta_0, \theta_1)$ $J(\theta_0, \theta_1, \theta_2, \ldots, \theta_n)$

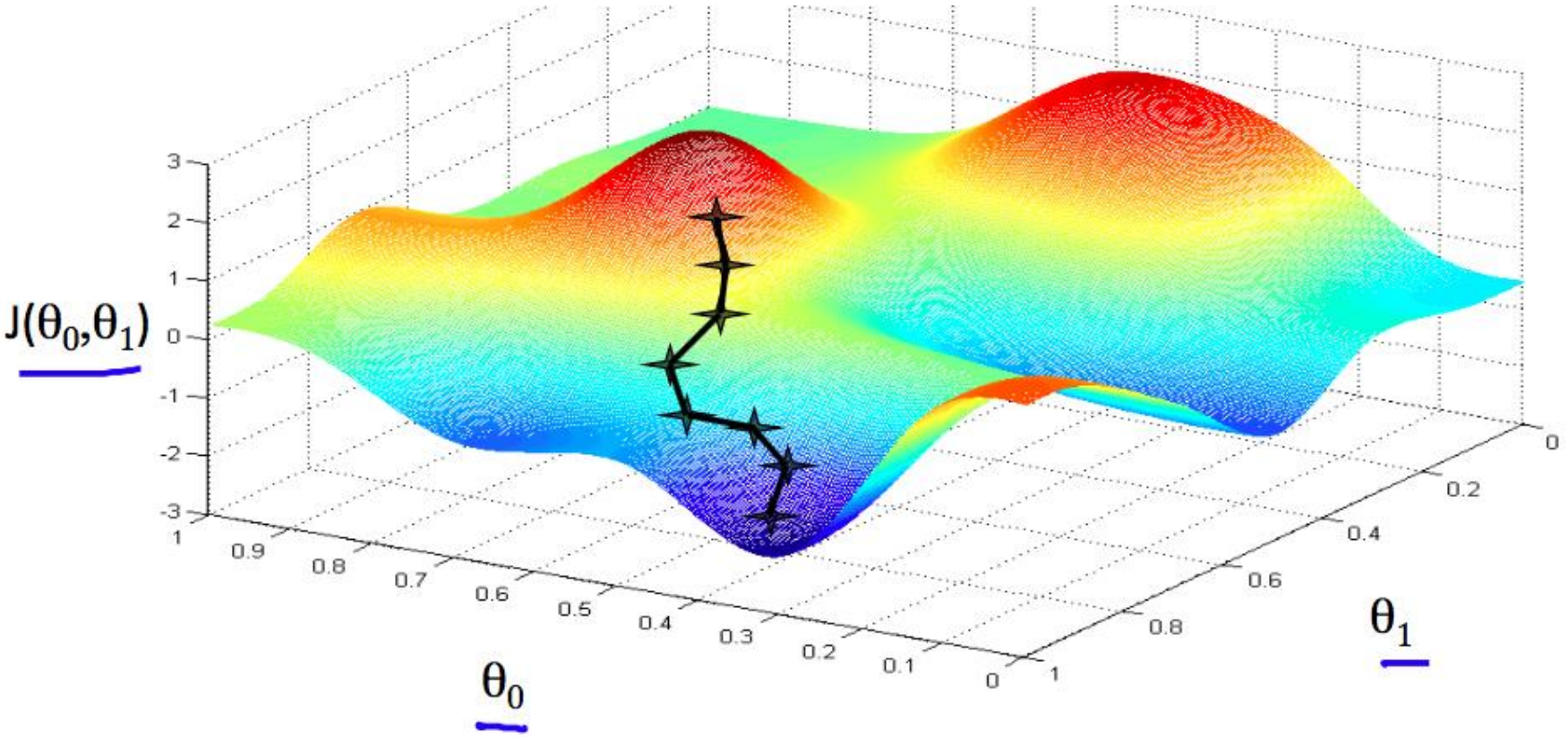
Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$ $\min_{\theta_0 \ldots \theta_n} J(\theta_0, \ldots, \theta_n)$

Outline:

- Start with some θ_0, θ_1 (say $\theta_0 = 0, \theta_1 = 0$)
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$

  until we hopefully end up at a minimum

Idea behind gradient descent



Idea behind gradient descent

Gradient descent algorithm

repeat until convergence {

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

}

(for $j = 0$ and $j = 1$)

learning rate

Simultaneously update θ_0 and θ_1

Correct: Simultaneous update

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

Incorrect:

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

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

$$\frac{\delta}{\delta \theta_j} J(\theta_0, \theta_1)$$

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

$$\theta_j :=$$

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

α:

Batch gradient descent

"Batch" Gradient Descent

"Batch": Each step of gradient descent uses all the training examples.

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

Derivation of update rules

$$\theta_j := \theta_j - \alpha \cdot \sum_{i=1}^{m} J'_j(\theta)$$

$$J'_j(\theta) = \frac{\delta}{\delta \theta_j} J(\theta)$$

$$J'_0(\theta) = h_\theta(x^{(i)}) - y^{(i)}$$

$$J'_1(\theta) = \left(h_\theta(x^{(i)}) - y^{(i)}\right) \cdot x^{(i)}$$

Batch gradient descent update rules

$$\theta_j := \theta_j - \alpha \cdot \sum_{i=1}^{m} J_j'(\theta)$$

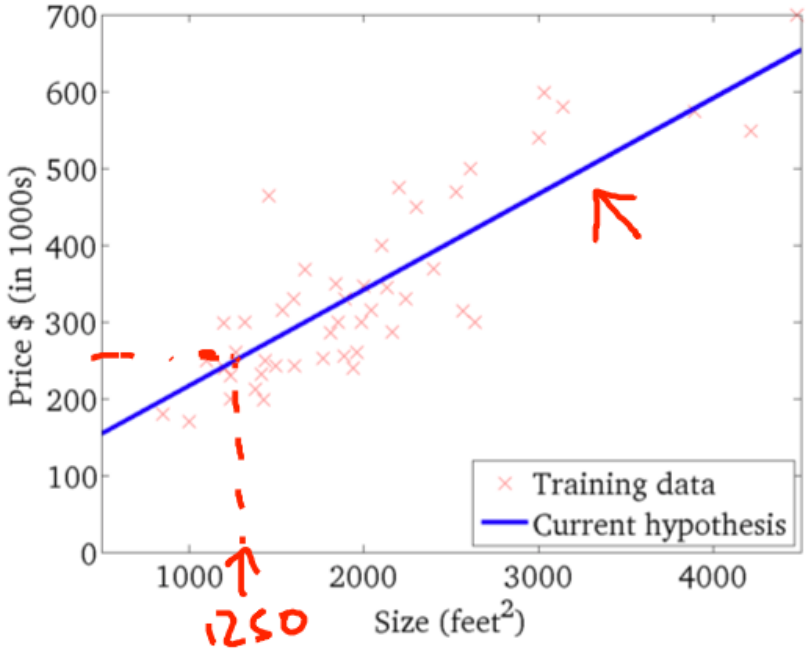
$$\theta_0 := \theta_0 - \alpha \cdot \sum_{i=1}^{m} h_\theta\left(x^{(i)}\right) - y^{(i)}$$

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

Gradient descent in action

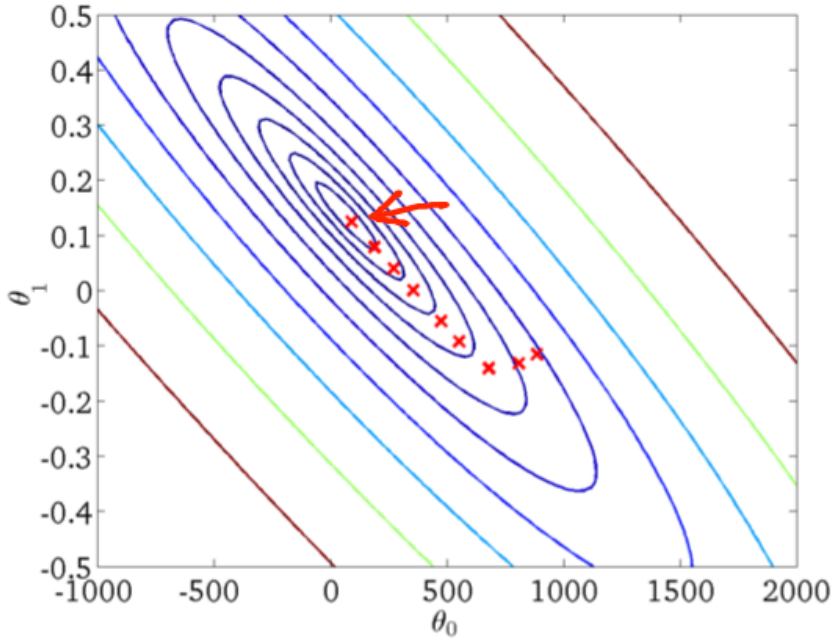
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



KB-74: Applied Data Science

THE HAGUE
UNIVERSITY OF
APPLIED SCIENCES
FACULTY IT & DESIGN