

COMP4434 Big Data Analytics

Lecture 2 Gradient Descent

HUANG Xiao

xiaohuang@comp.polyu.edu.hk



Definition of Machine Learning

- H. Simon

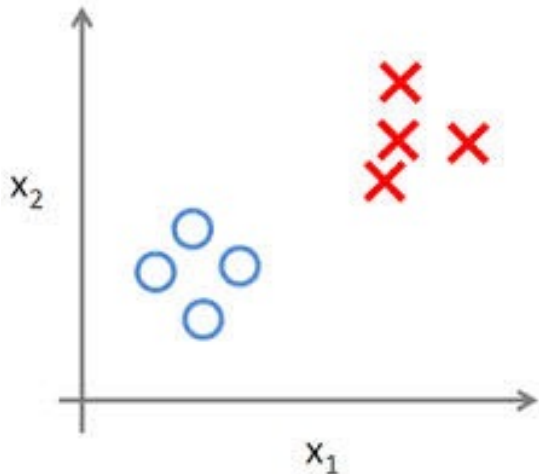
Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the task or tasks drawn from the same population more efficiently and more effectively the next time.

- T. Mitchell: Well posed machine learning –
Improving performance via experience

Formally, a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T as measured by P improves with experience E .

Machine Learning Algorithms

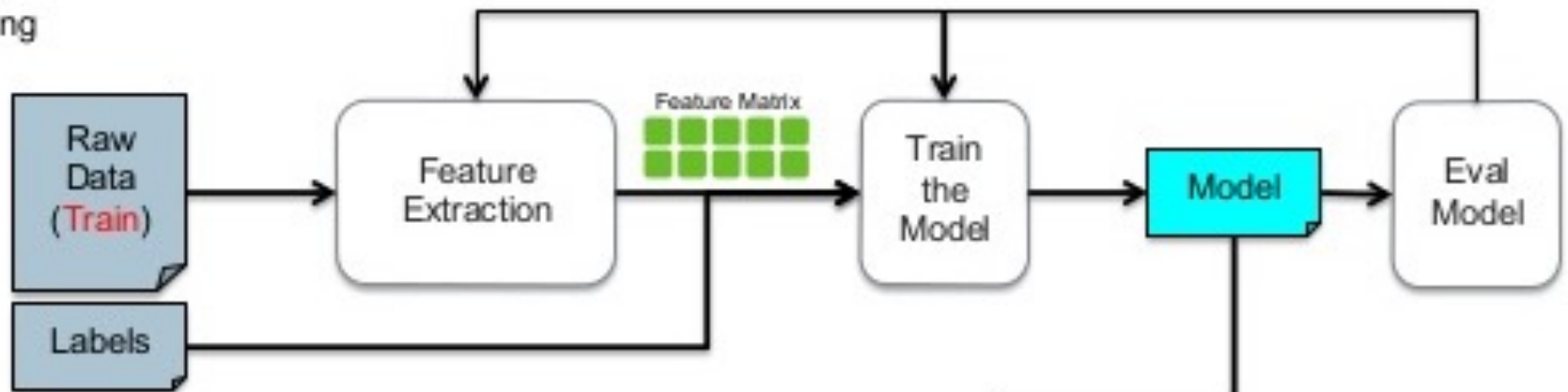
- Supervised Learning
 - Training data includes desired outputs



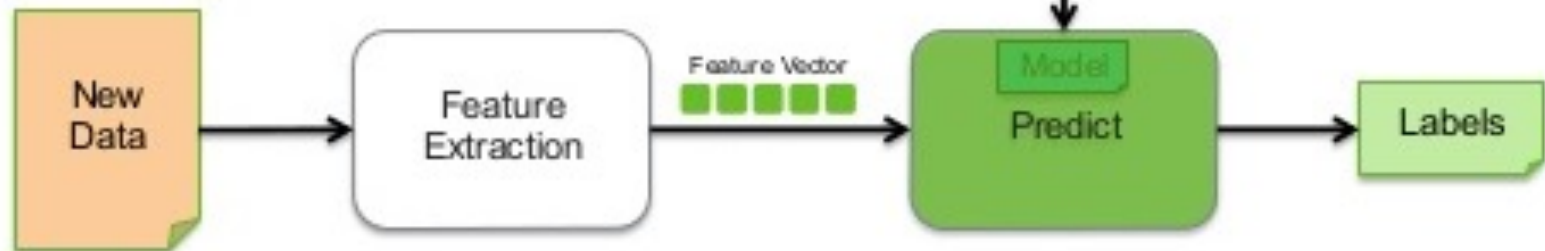
- Unsupervised Learning
 - Training data does not include desired outputs
 - Find hidden structure in data
- Semi-supervised Learning
- Reinforcement Learning

Supervised Learning Workflow

Training

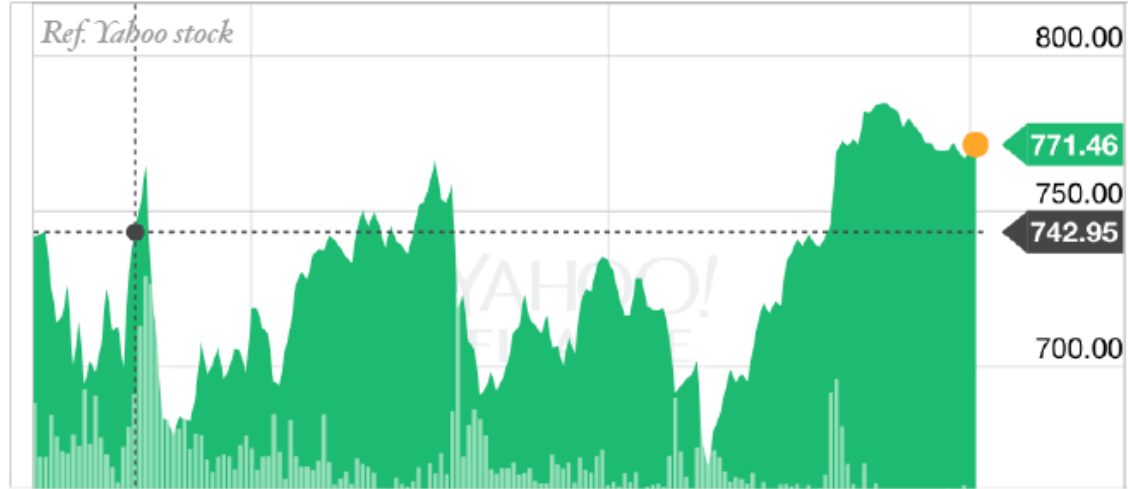


Predicting



Supervised Learning Task - Regression

- Regression
 - Learning to predict a continuous/real value
 - Ex: housing price, gold price, stock price



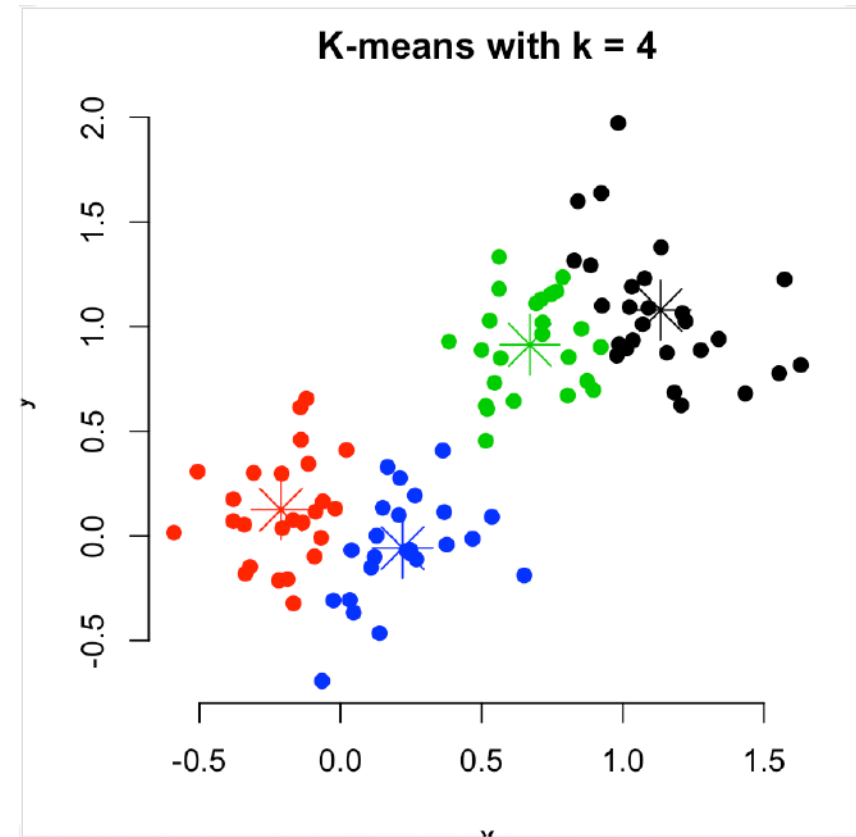
Supervised Learning Task - Classification

- Classification
 - Learning to predict a discrete value from a predefined set of values
 - Ex. weather prediction, spam email filtering, product categorization, object detection, medical diagnose



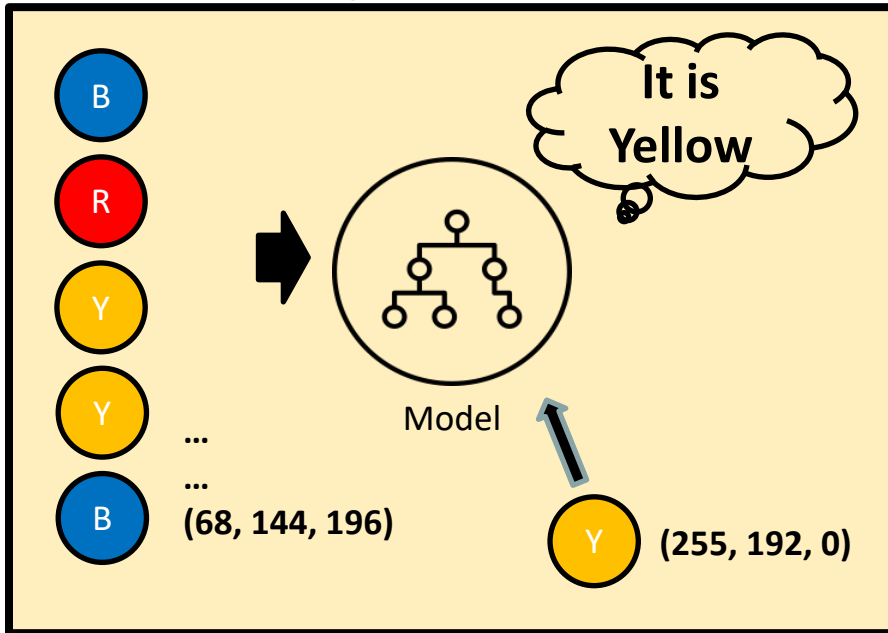
Unsupervised Learning Task - Clustering

- Clustering
 - Determine the intrinsic grouping in a set of unlabeled data
 - Ex. clustering in networking, image clustering



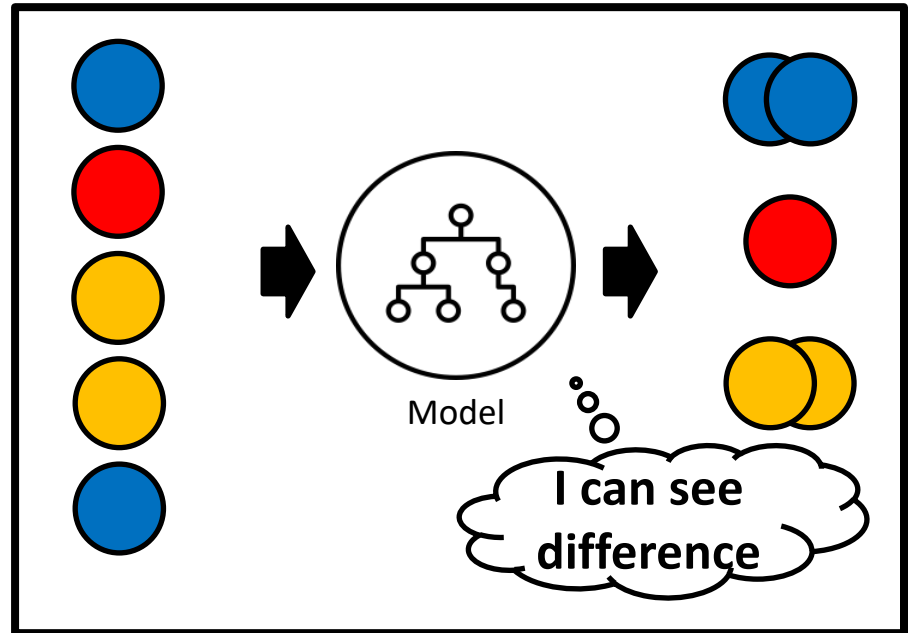
Supervised vs Unsupervised

Supervised



- Labeled Data
- Direct Feedback
- Predict Output

Unsupervised

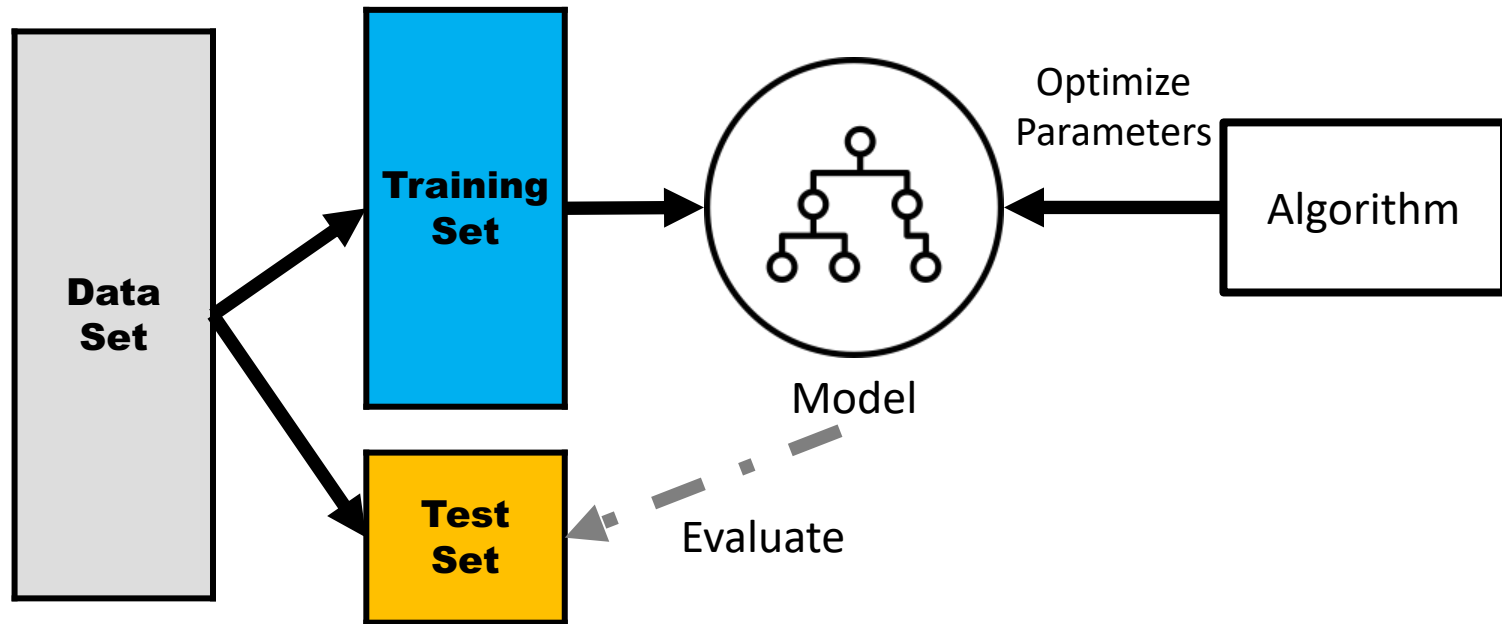


- Non-labeled Data
- No Feedback
- Find Hidden Structure in Data

Supervised learning tasks

- Classification
 - predicts **categorical** class labels
 - classifies data (constructs a model) based on the training set and the values (class labels) and uses the trained model to classify new data
 - return a discrete-value (label) as output, e.g., classifying Hang Seng Index (HSI)'s trend as Up, Down, Level
- Regression
 - models **continuous-valued** functions, i.e., predicts unknown or missing values
 - Return a real-value as output, e.g., predicting HSI's future values

Two-Step Process



1. Model Construction: describing a set of samples

- Each sample is associate with a label attribute
- The set of samples used for model construction: **training set**
- The model is represented as mathematical formulae

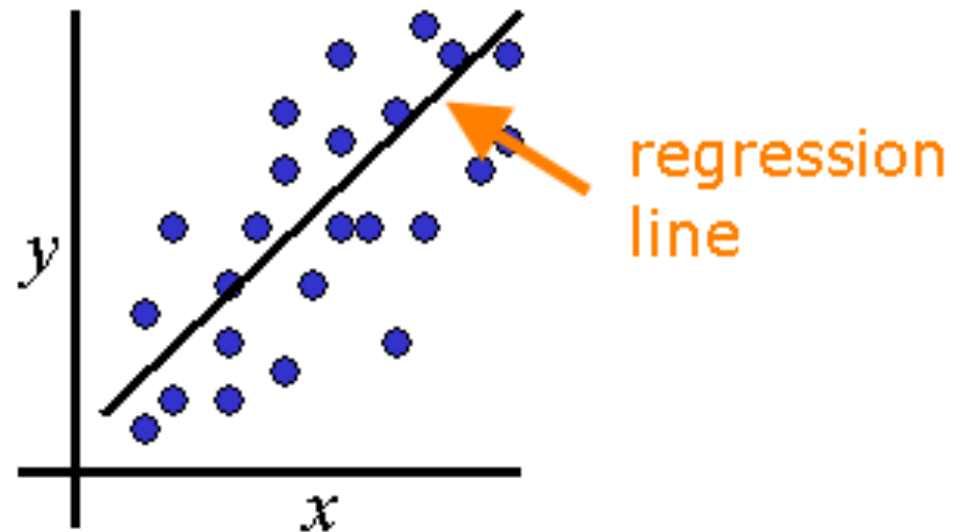
2. Model Usage: for future or unknown objects

- The known label of test sample is compared with the result from the model
- Test set is **independent** of training set

Supervised Learning - Linear Regression

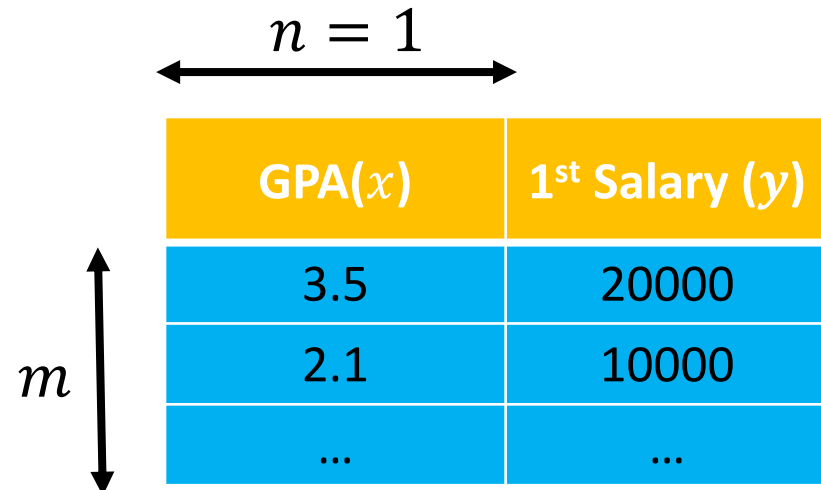
- A Regression Problem - Line Fitting
 - E.g., “My GPA is 2.9, what will be my salary?”

GPA (x)	1 st Salary (y)
3.5	20000
2.1	10000
...	...



Notations

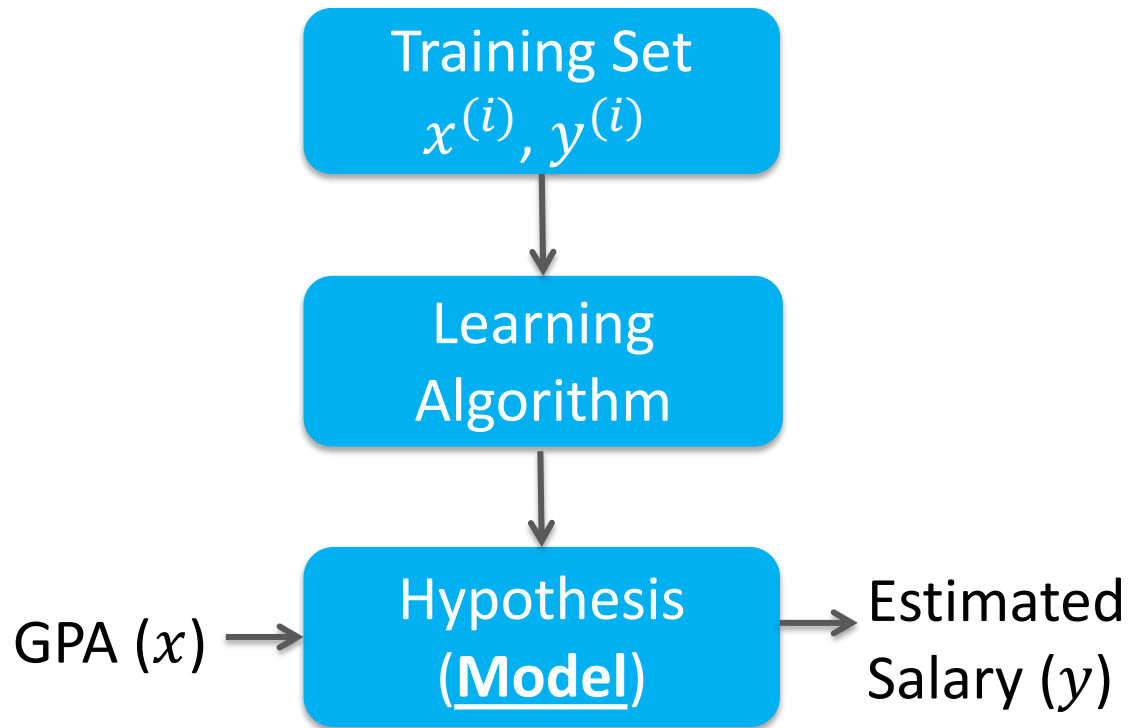
- x : input variables/attributes/features
- y : output variable/attribute/target variable
- m : number of training examples
- n : number of input variables
 - Univariate: $n = 1$
 - Multivariate: $n > 1$



A diagram illustrating a data matrix. It consists of a table with 3 rows and 2 columns. The first column is labeled 'GPA(x)' and the second column is labeled '1st Salary (y)'. The first two rows contain the values 3.5 and 20000, and 2.1 and 10000 respectively. The third row contains ellipses (...). To the left of the table, a vertical double-headed arrow is labeled 'm', indicating the number of training examples (rows). Above the table, a horizontal double-headed arrow is labeled 'n = 1', indicating the number of input variables (columns).

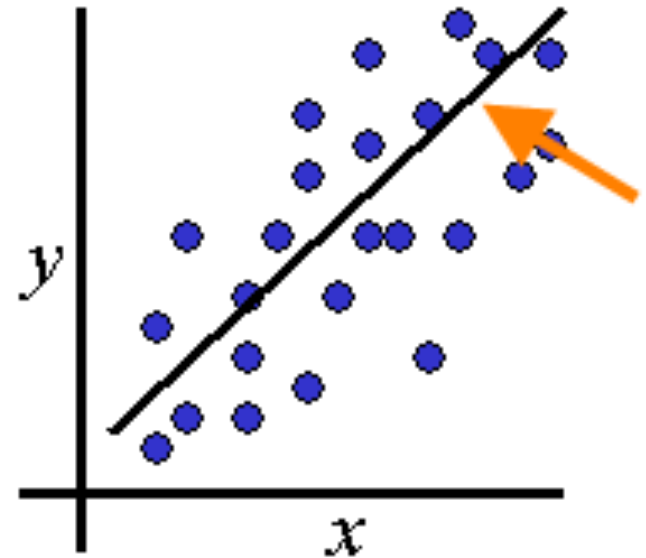
GPA(x)	1 st Salary (y)
3.5	20000
2.1	10000
...	...

Model $h_{\theta}(x) = \theta_0 + \theta_1 x$

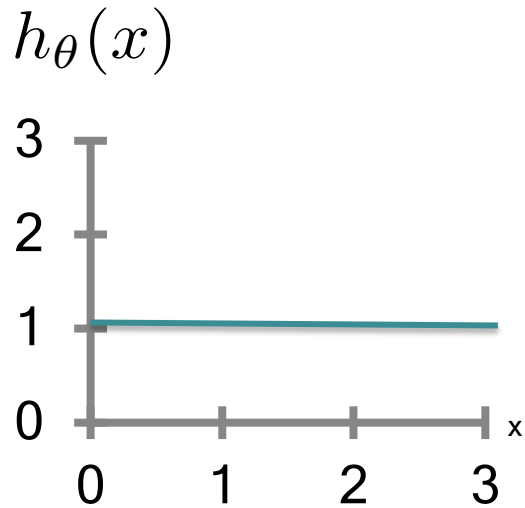


How do we represent h ?

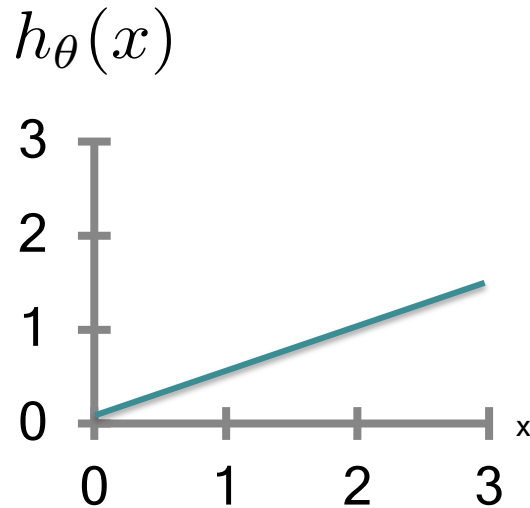
$$h_{\theta}(x) = \theta_0 + \theta_1 x$$



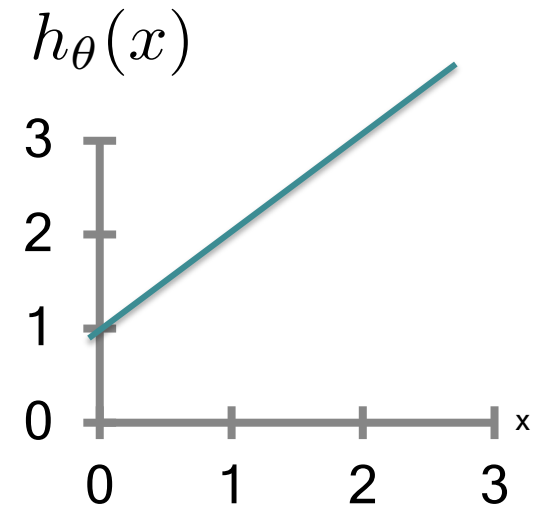
How to Set θ



$$\theta_0 = 1$$
$$\theta_1 = 0$$



$$\theta_0 = 0$$
$$\theta_1 = 0.5$$



$$\theta_0 = 1$$
$$\theta_1 = 1$$

What is the Best Fitting Line?

- Finding θ , which makes $h_{\theta}(x)$ closest to y for all training data $(x^{(i)}, y^{(i)})$
- Mathematical definition: **Cost Function** $J(\theta_0, \theta_1)$

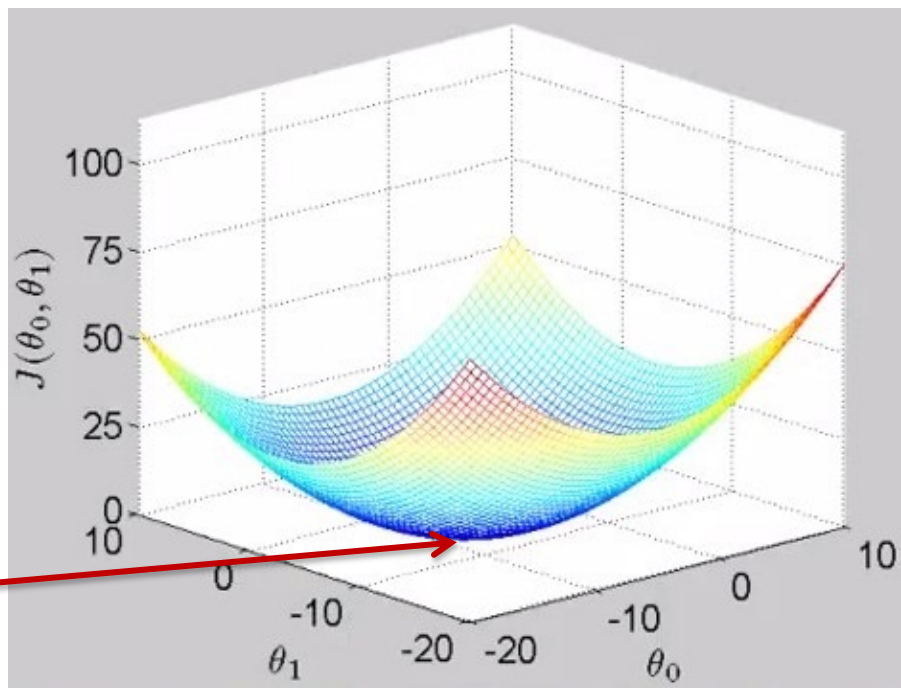
$$\min_{\theta_0, \theta_1} \frac{1}{2m} \sum_{i=1}^m \left(\underline{h_{\theta}(x^{(i)}) - y^{(i)}} \right)^2$$

Error: (Estimated – Actual)
Squared Error: to be positive

Cost Function $J(\theta_0, \theta_1)$

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

Our Target



- Input: $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$
- Start with some θ_0, θ_1 (e.g., $\theta_0 = 0, \theta_1 = 0$)
- Keep refining θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until we hopefully end up at a minimum

Gradient Descent Algorithm

Repeat until convergence {

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

}

**Learning Rate
(or Step Size)**

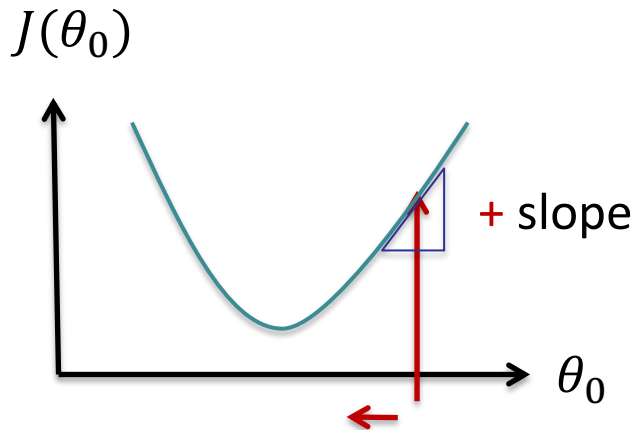
**Derivative
Slope
Gradient**

Three Problems:

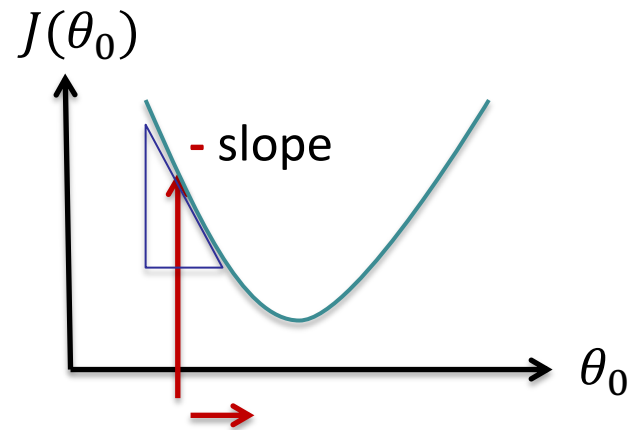
1. How to compute the derivative?
2. How to set the learning rate?
3. What is the convergence criteria?

Derivative

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



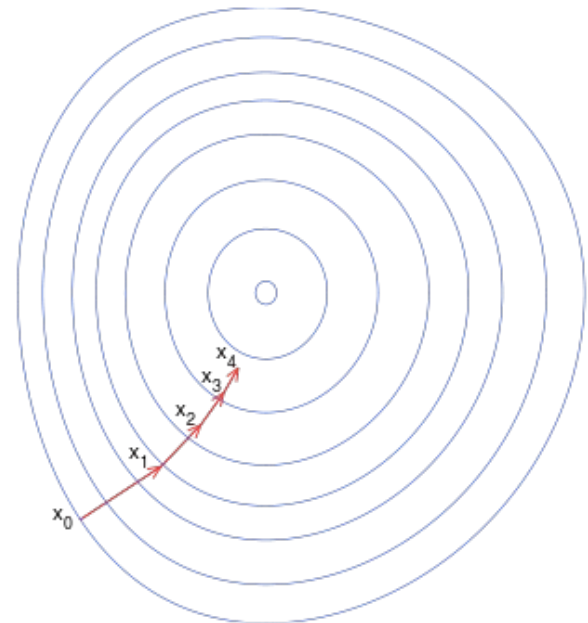
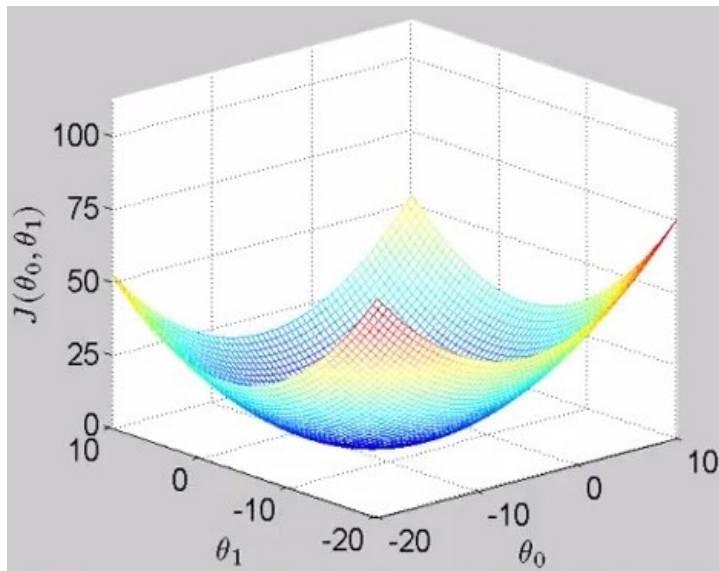
$$\theta_0 := \theta_0 - \alpha \text{ (+ number)}$$



$$\theta_0 := \theta_0 - \alpha \text{ (- number)}$$

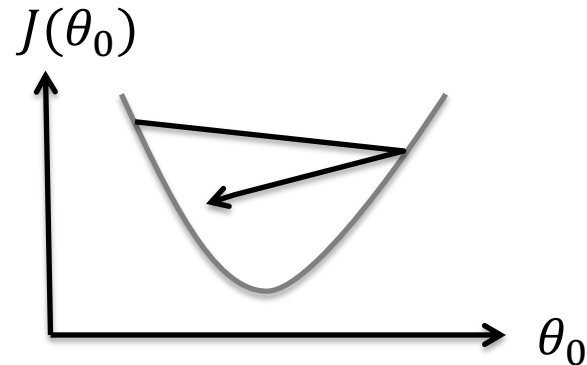
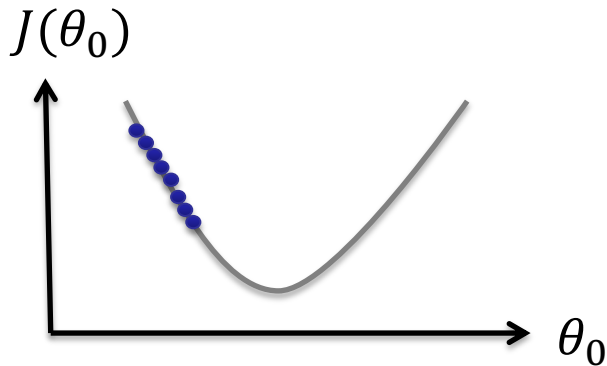
Geometric Interpretation: Gradient Decent

- The gradient can be interpreted as the **direction and rate of fastest increase**
- Parameters update in reverse direction of gradient

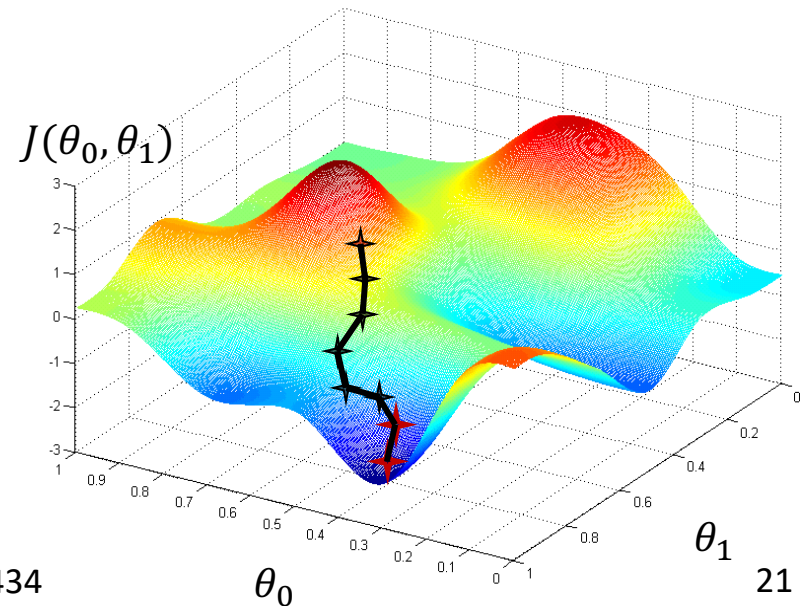


Step Size

- Learning rate / Step size α is a user parameter.



Small $\alpha \Rightarrow$ slow convergence Large $\alpha \Rightarrow$ fail to converge



Gradient

- Gradient of J is the vector

$$\nabla J(\theta_0, \theta_1) = \begin{bmatrix} \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0} \\ \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1} \end{bmatrix}$$

- Partial derivative ∂ of a function $J(\theta_0, \theta_1)$ of several variables θ_0, θ_1 is its derivative with respect to one of those variables, with the others held constant

Gradient $\partial J(\theta_0, \theta_1) / \partial \theta_0$

$$\begin{aligned}\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0} &= \frac{\partial}{\partial \theta_0} \left(\frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \right) \\&= \frac{\partial}{\partial \theta_0} \left(\frac{1}{2m} \sum_{i=1}^m (\theta_0 + \theta_1 x^{(i)} - y^{(i)})^2 \right) \\&= \frac{1}{2m} \sum_{i=1}^m \frac{\partial}{\partial \theta_0} (\theta_0 + \theta_1 x^{(i)} - y^{(i)})^2 \\&= \frac{1}{2m} \sum_{i=1}^m 2(\theta_0 + \theta_1 x^{(i)} - y^{(i)}) \\&= \frac{1}{m} \sum_{i=1}^m (\theta_0 + \theta_1 x^{(i)} - y^{(i)})\end{aligned}$$

$$\begin{aligned}y &= f(u) = f(g(x)) \\ \frac{dy}{dx} &= \frac{dy}{du} \frac{du}{dx} = f'(u)g'(x)\end{aligned}$$

Chain Rule

Gradient $\partial J(\theta_0, \theta_1) / \partial \theta_1$

$$\begin{aligned}\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1} &= \frac{\partial}{\partial \theta_1} \left(\frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \right) \\&= \frac{\partial}{\partial \theta_1} \left(\frac{1}{2m} \sum_{i=1}^m (\theta_0 + \theta_1 x^{(i)} - y^{(i)})^2 \right) \\&= \frac{1}{2m} \sum_{i=1}^m \frac{\partial}{\partial \theta_1} (\theta_0 + \theta_1 x^{(i)} - y^{(i)})^2 \\&= \frac{1}{2m} \sum_{i=1}^m 2(\theta_0 + \theta_1 x^{(i)} - y^{(i)}) \cdot x^{(i)} \\&= \frac{1}{m} \sum_{i=1}^m (\theta_0 + \theta_1 x^{(i)} - y^{(i)}) x^{(i)}\end{aligned}$$

Gradient Descent Algorithm

$$n = 0, \theta_0[n] = \theta_1[n] = 0$$

REPEAT {

$$\theta_0[n + 1] = \theta_0[n] - \frac{\alpha}{m} \sum_{i=1}^m (\theta_0[n] + \theta_1[n]x^{(i)} - y^{(i)})$$

$$\theta_1[n + 1] = \theta_1[n] - \frac{\alpha}{m} \sum_{i=1}^m (\theta_0[n] + \theta_1[n]x^{(i)} - y^{(i)})x^{(i)}$$

$$n = n + 1$$

} UNTIL $J(\theta_0[n - 1], \theta_1[n - 1]) - J(\theta_0[n], \theta_1[n]) < \varepsilon$ OR $n > X$

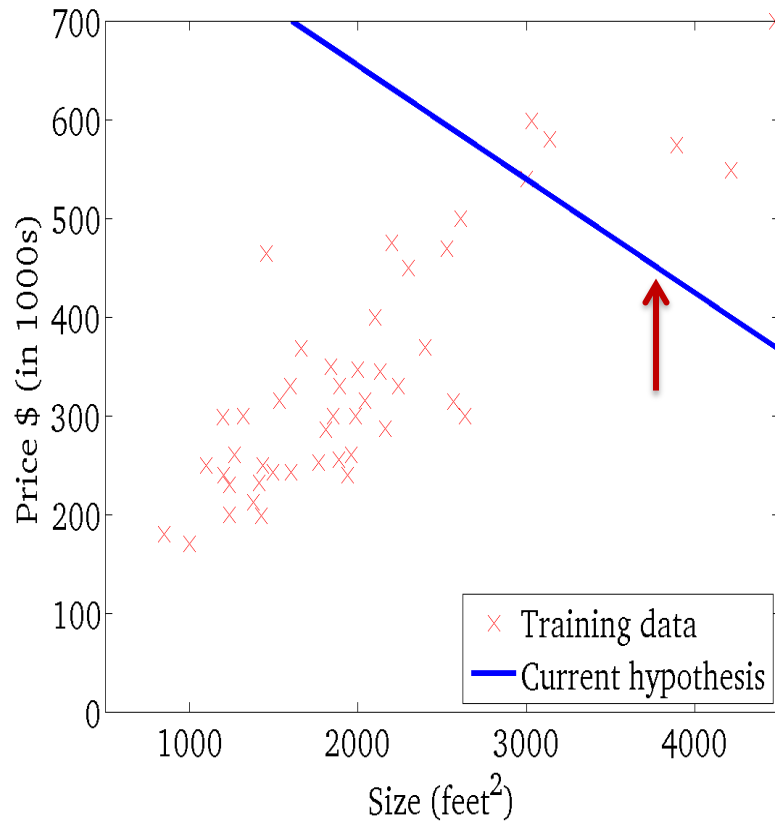
Stop Criteria

- If $J(\theta_0, \theta_1)$ decreases by less than a threshold ε (e.g., 10^{-3}) in one iteration
 - Each iteration, we use **all** m training examples to update θ_0, θ_1
 - Use updated θ_0, θ_1 to recalculate $J(\theta_0, \theta_1)$ and find the decrease
- Or, after X (e.g., **5000**) number of iterations

Gradient Descent in Action

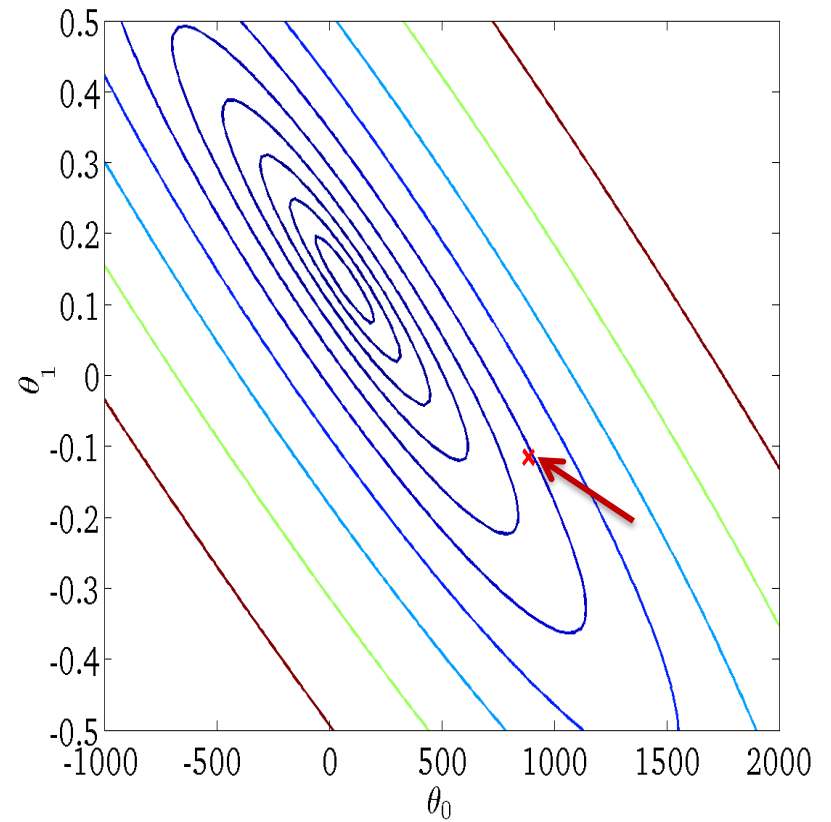
$$h_{\theta}(x)$$

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



$$J(\theta_0, \theta_1)$$

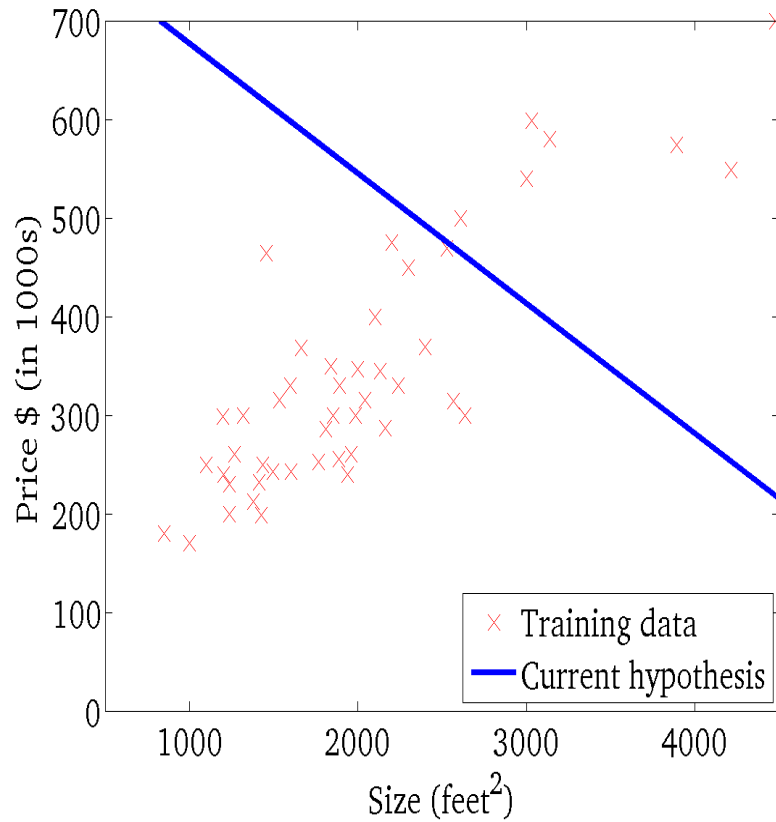
(function of the parameters θ_0, θ_1)



Gradient Descent in Action

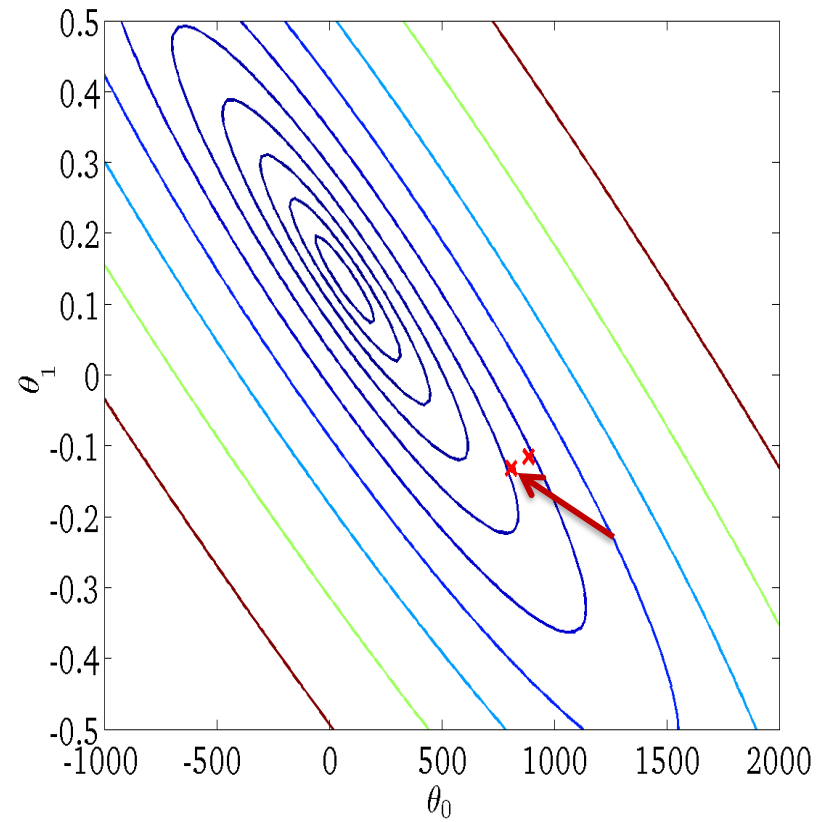
$$h_{\theta}(x)$$

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



$$J(\theta_0, \theta_1)$$

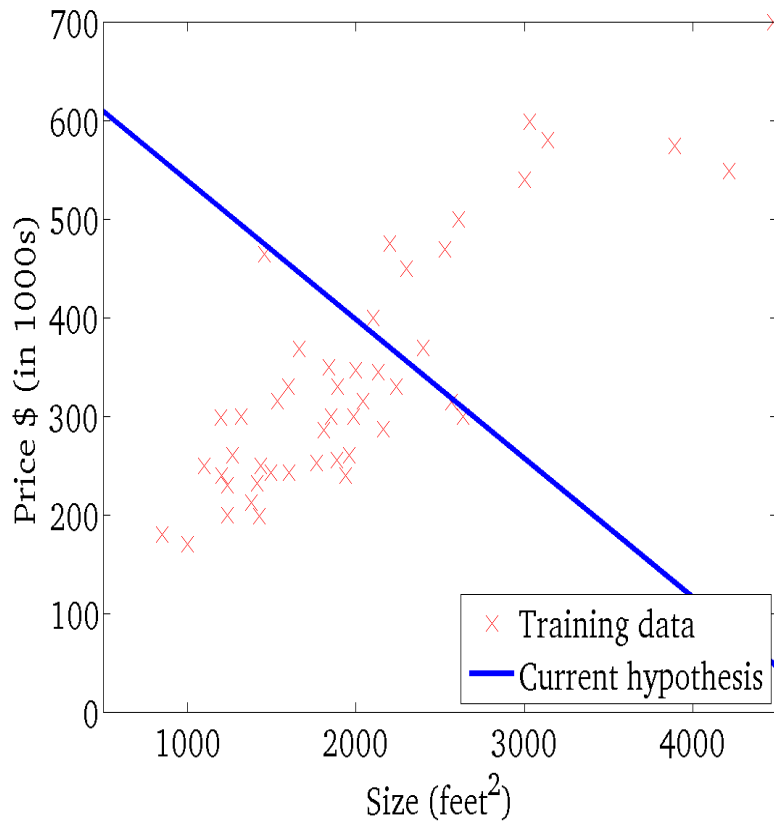
(function of the parameters θ_0, θ_1)



Gradient Descent in Action

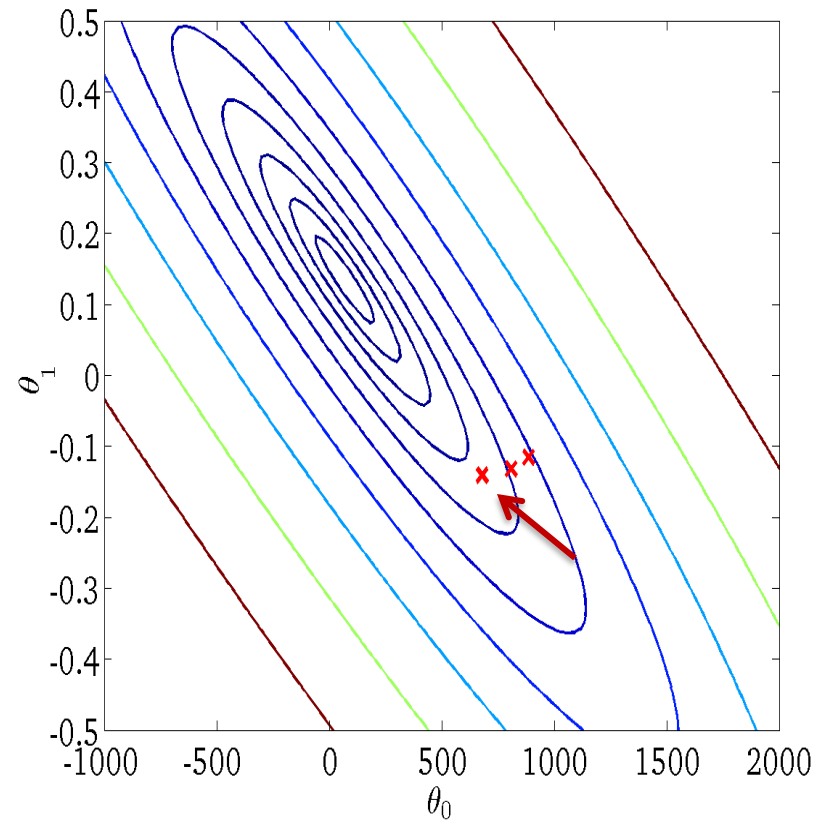
$$h_{\theta}(x)$$

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



$$J(\theta_0, \theta_1)$$

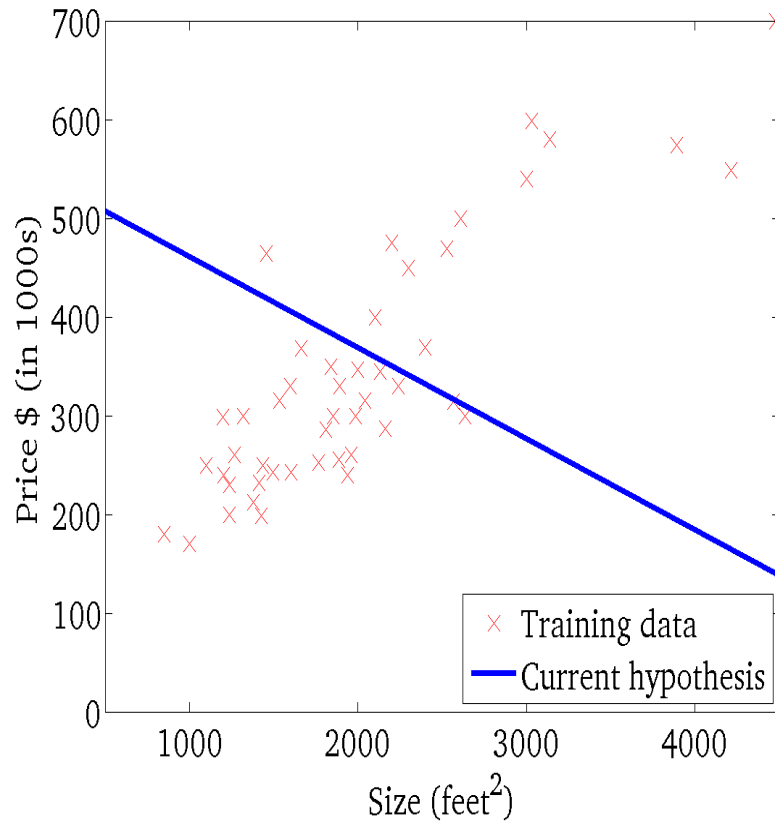
(function of the parameters θ_0, θ_1)



Gradient Descent in Action

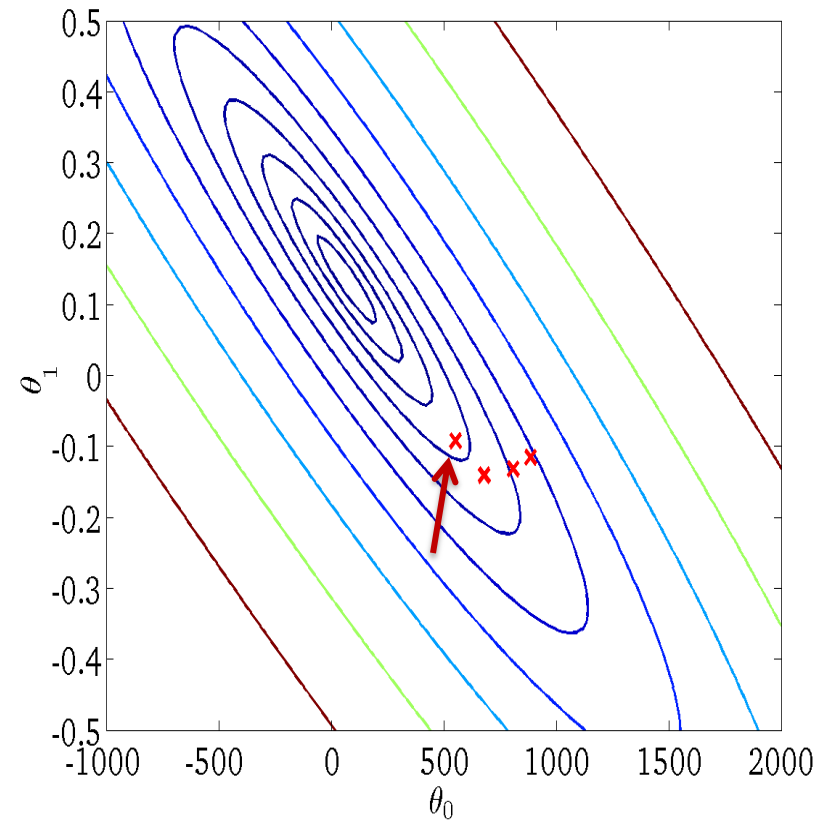
$$h_{\theta}(x)$$

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



$$J(\theta_0, \theta_1)$$

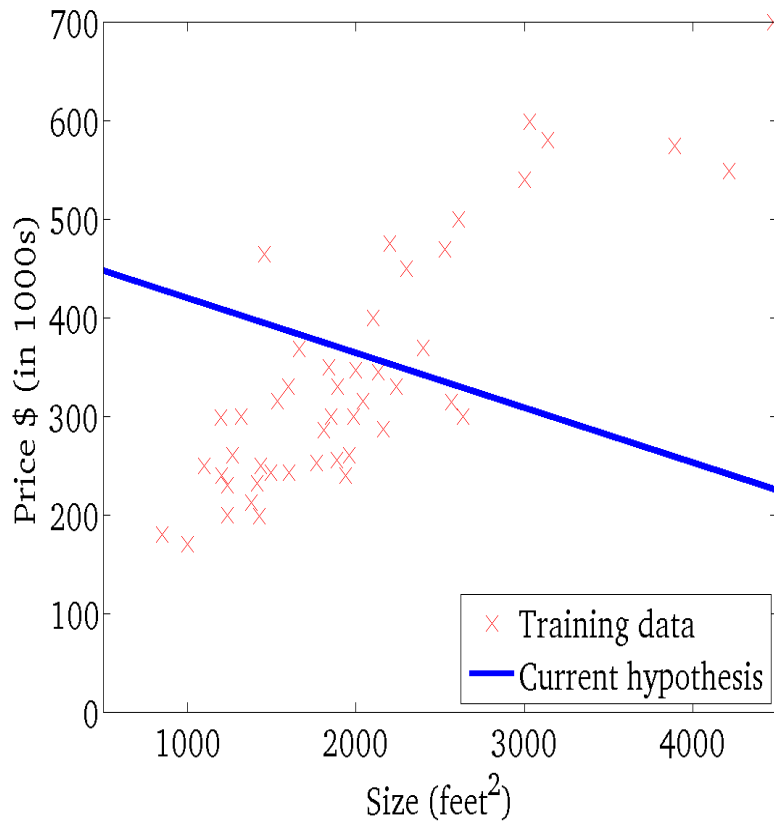
(function of the parameters θ_0, θ_1)



Gradient Descent in Action

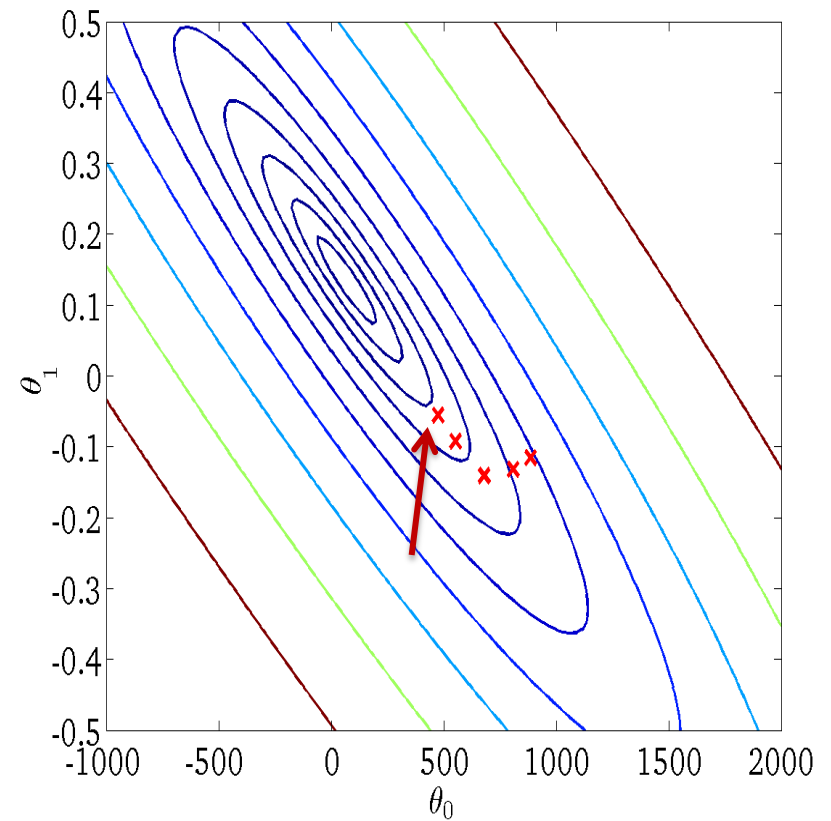
$$h_{\theta}(x)$$

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



$$J(\theta_0, \theta_1)$$

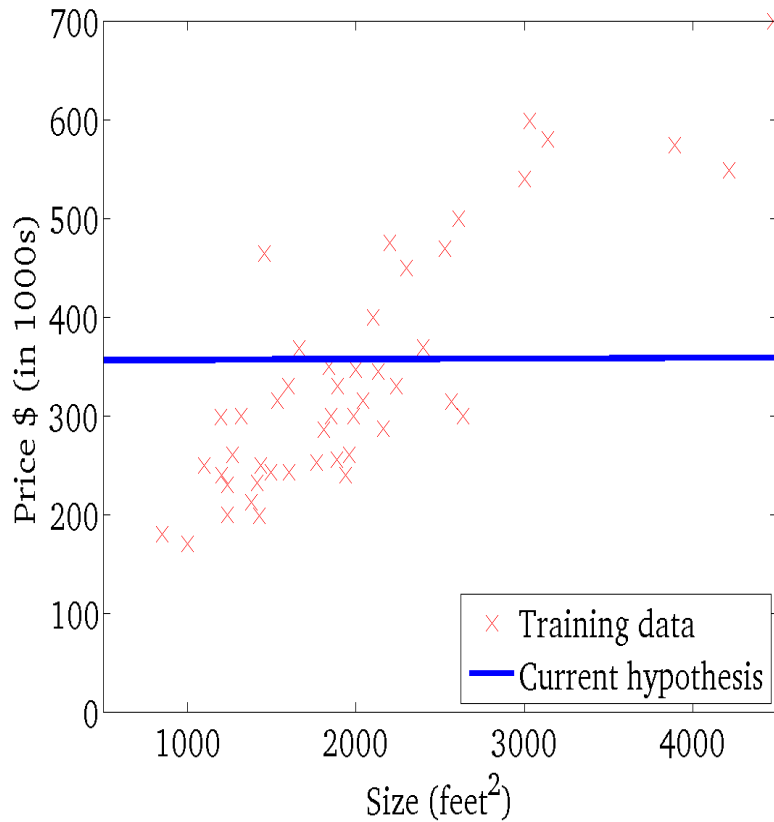
(function of the parameters θ_0, θ_1)



Gradient Descent in Action

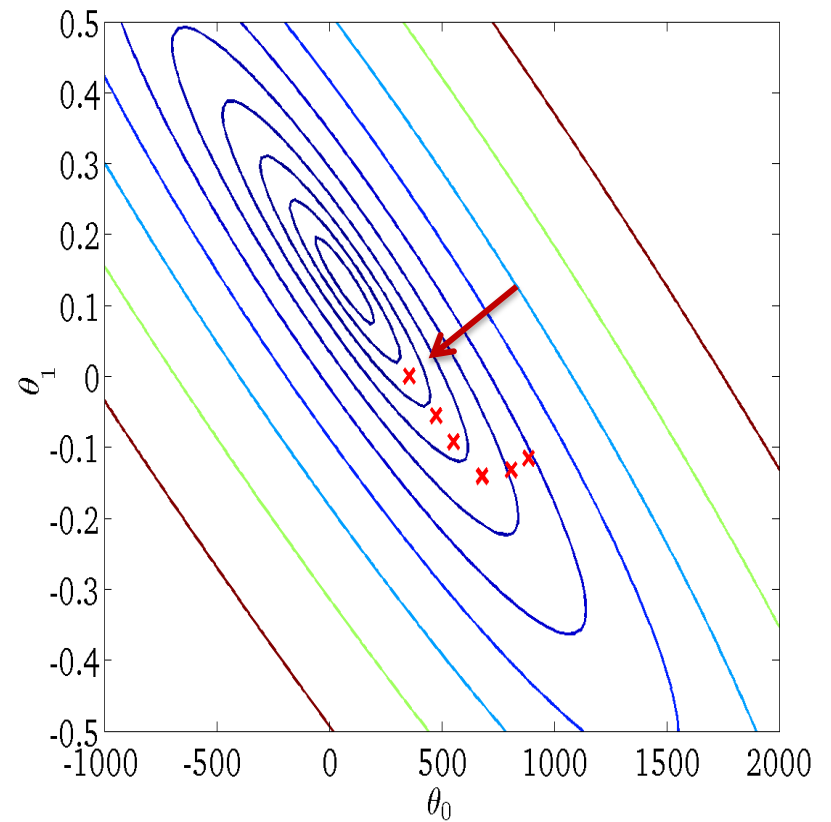
$$h_{\theta}(x)$$

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



$$J(\theta_0, \theta_1)$$

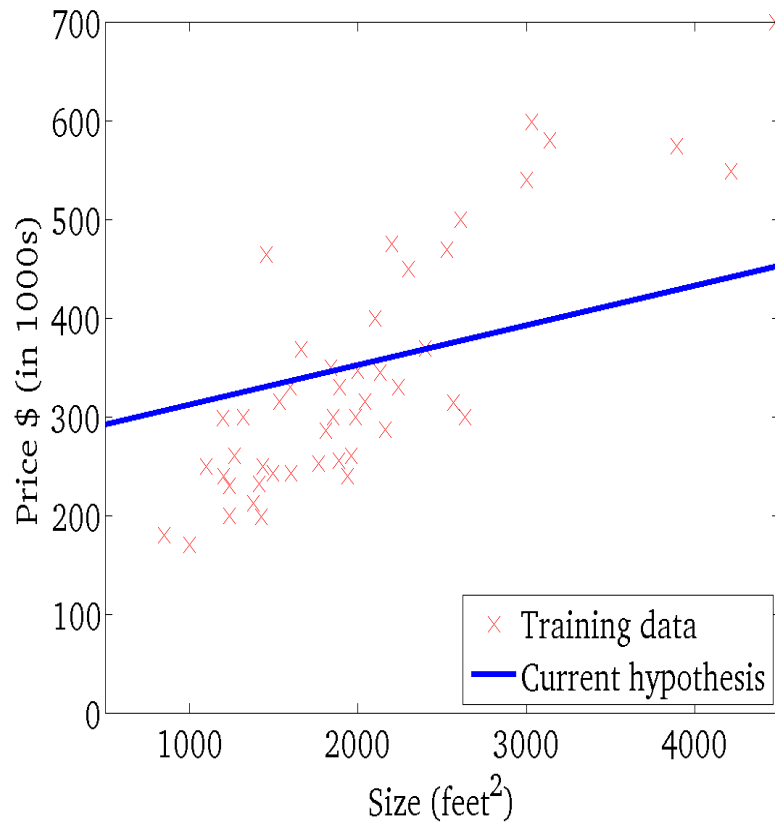
(function of the parameters θ_0, θ_1)



Gradient Descent in Action

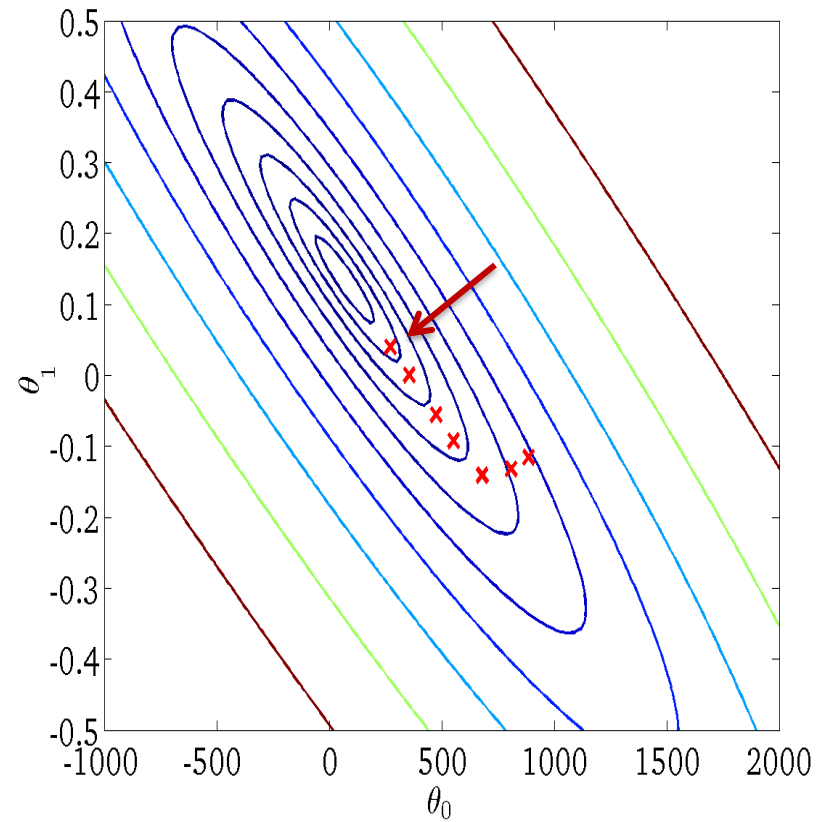
$$h_{\theta}(x)$$

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



$$J(\theta_0, \theta_1)$$

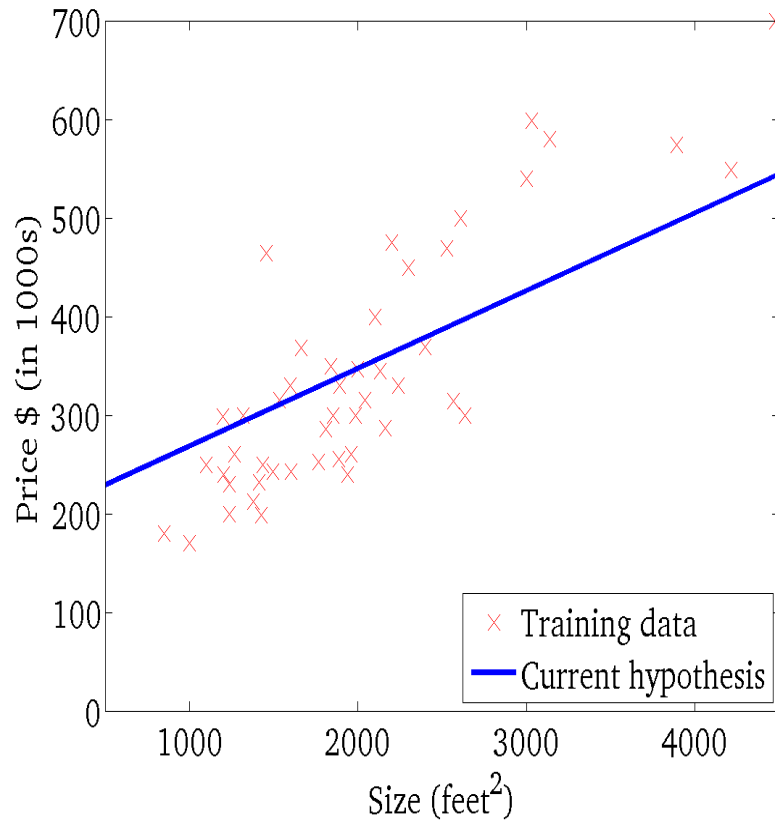
(function of the parameters θ_0, θ_1)



Gradient Descent in Action

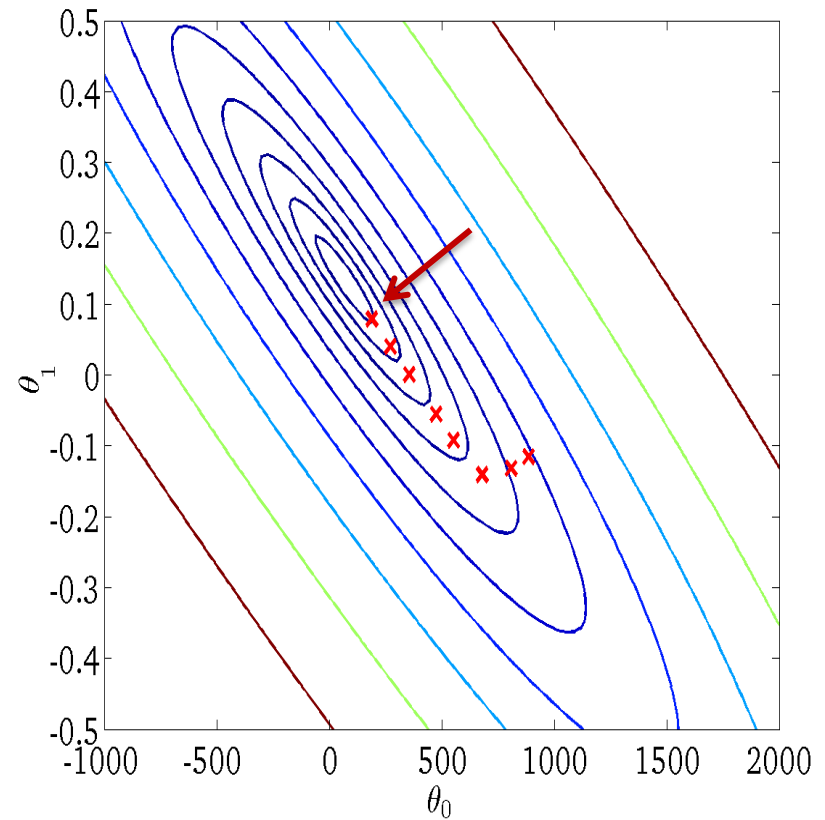
$$h_{\theta}(x)$$

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



$$J(\theta_0, \theta_1)$$

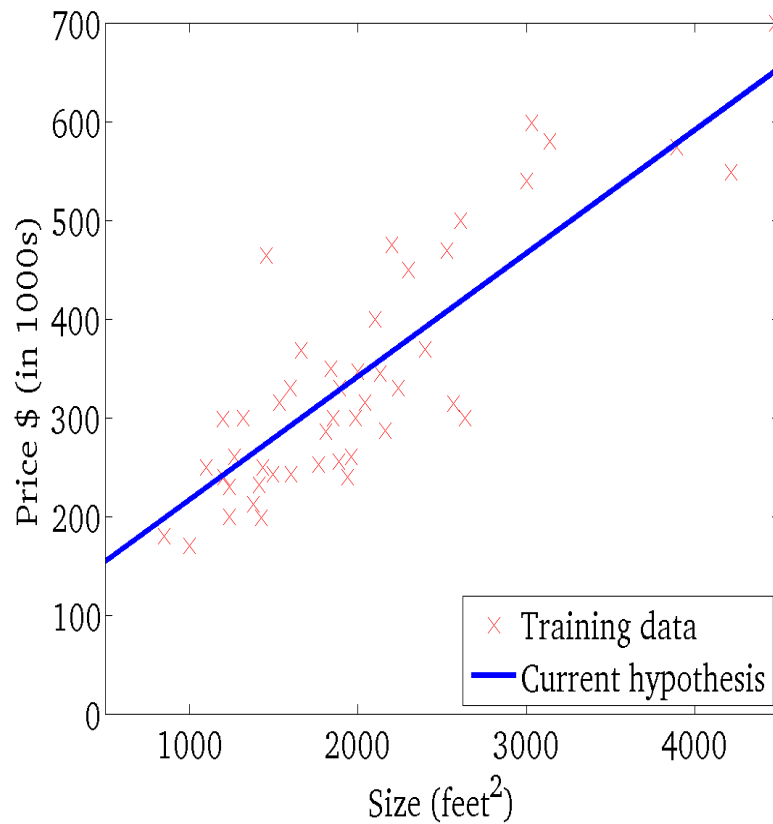
(function of the parameters θ_0, θ_1)



Gradient Descent in Action

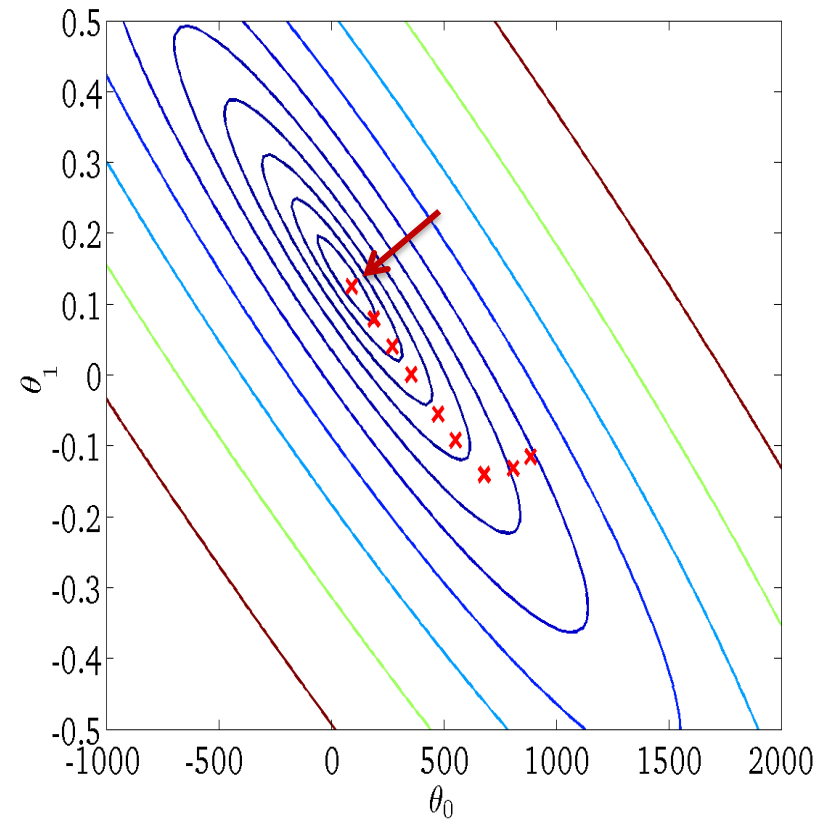
$$h_{\theta}(x)$$

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

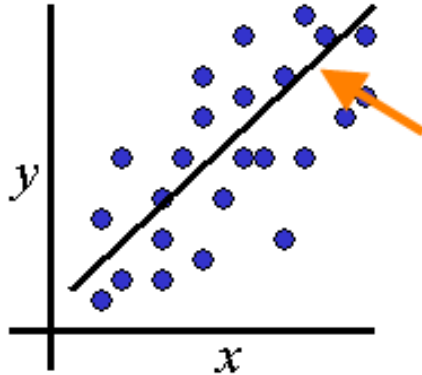


$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



Summary: Univariate Linear Regression



Target: Find best fitting line $h_{\theta}(x) = \theta_0 + \theta_1 x$

$$\text{minimize } J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m \underbrace{(h_{\theta}(x^{(i)}) - y^{(i)})^2}_{\text{Squared Error: (Estimated - Actual)}^2}$$

Squared Error: (Estimated – Actual)²

Repeat until convergence {

$$\theta_0 = \theta_0 - \alpha \cdot \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0} \quad \text{Derivative (or Gradient)}$$

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

}

Learning Rate
(or Step Size)

Gradient Descent Algorithm

Multivariate Linear Regression

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

$n = 1$		$n = 3$			
GPA (x)	1 st Salary (y)				
3.5	20000	GPA (x_1)	# of Exchanges (x_2)	Age (x_3)	1 st Salary (y)
2.1	10000	3.5	1	23	20000
...	...	2.1	2	22	10000
	

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

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

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

Example with Training data

The training data contain some example measurements of the profit gained by opening an outlet in the cities with the population ranging between 30,000 and 100,000. The y-values are the profit measured in USD, and the x-values are the populations of the city. Each city population and profit tuple constitutes one training example in training dataset.

City Population (10^4) x	Profit (10^4) y
6.4862	6.5987
5.5277	9.1302
8.5186	13.662
7.0032	11.854

Initialize Gradient Descent

Use training dataset to develop a linear regression model and solve it by using gradient descent algorithm. Find the values of θ_0 , θ_1 , and cost function J in the first two iterations. ($\theta_0 = 0, \theta_1 = 0, \alpha = 0.01$)

Initial setting:

$$\theta_0[0] = \theta_1[0] = 0$$

Iteration 1

■ Update θ_0 :

$$\begin{aligned}\frac{\partial J}{\partial \theta_0}[0] &= \frac{1}{4} \sum_{i=1}^4 (\theta_0[0] + \theta_1[0]x^{(i)} - y^{(i)}) \\ &= 1/4 [-y^{(1)} - y^{(2)} - y^{(3)} - y^{(4)}] \\ &= 1/4 [-6.5987 - 9.1302 - 13.662 - 11.854] = -10.311 \\ \theta_0[1] &= \theta_0[0] - 0.01 \frac{\partial J}{\partial \theta_0}[0] = 0.10311\end{aligned}$$

■ Update θ_1 :

$$\begin{aligned}\frac{\partial J}{\partial \theta_1}[0] &= \frac{1}{4} \sum_{i=1}^4 (\theta_0[0] + \theta_1[0]x^{(i)} - y^{(i)})x^{(i)} \\ &= 1/4 [-y^{(1)}x^{(1)} - y^{(2)}x^{(2)} - y^{(3)}x^{(3)} - y^{(4)}x^{(4)}] \\ &= 1/4 [-6.5987 \times 6.4862 - 9.1302 \times 5.5277 - 13.662 \times 8.5186 \\ &\quad - 11.854 \times 7.0032] = -73.167 \\ \theta_1[1] &= \theta_1[0] - 0.01 \frac{\partial J}{\partial \theta_1}[0] = 0.73167\end{aligned}$$

Cost Update

- Update $J(\theta_0, \theta_1)$:

$$\begin{aligned} J(\theta_0[0], \theta_1[0]) &= \frac{1}{8} \sum_{i=1}^4 (\theta_0[0] + \theta_1[0]x^{(i)} - y^{(i)})^2 \\ &= 1/8 [(y^{(1)})^2 + (y^{(2)})^2 + (y^{(3)})^2 + (y^{(4)})^2] \\ &= 1/8 [(6.5987)^2 + (9.1302)^2 + (13.662)^2 + (11.854)^2] = 56.759 \end{aligned}$$

$$\begin{aligned} J(\theta_0[1], \theta_1[1]) &= \frac{1}{8} \sum_{i=1}^4 (\theta_0[1] + \theta_1[1]x^{(i)} - y^{(i)})^2 \\ &= 1/8 [(0.10311 + 0.73167 \times 6.4862 - 6.5987)^2 \\ &\quad + (0.10311 + 0.73167 \times 5.5277 - 9.1302)^2 \\ &\quad + (0.10311 + 0.73167 \times 8.5186 - 13.662)^2 \\ &\quad + (0.10311 + 0.73167 \times 7.0032 - 11.854)^2] = 15.864 \end{aligned}$$

Iteration 2

- Update θ_0 :

$$\frac{\partial J}{\partial \theta_0}[1] = \frac{1}{4} \sum_{i=1}^4 (\theta_0[1] + \theta_1[1]x^{(i)} - y^{(i)}) = \dots$$

$$\theta_0[2] = \theta_0[1] - 0.01 \frac{\partial J}{\partial \theta_0}[1] = \dots$$

- Update θ_1 :

$$\frac{\partial J}{\partial \theta_1}[1] = \frac{1}{4} \sum_{i=1}^4 (\theta_0[1] + \theta_1[1]x^{(i)} - y^{(i)})x^{(i)} = \dots$$

$$\theta_1[2] = \theta_1[1] - 0.01 \frac{\partial J}{\partial \theta_1}[1] = \dots$$

- Update $J(\theta_0, \theta_1)$:

$$J(\theta_0[2], \theta_1[2]) = \frac{1}{8} \sum_{i=1}^4 (\theta_0[2] + \theta_1[2]x^{(i)} - y^{(i)})^2 = \dots$$

Linear Regression by Linear Algebra

- Minimizing function:

$$\begin{aligned} & \min_{\theta_0, \theta_1, \dots} J(\theta_0, \theta_1, \dots) \\ &= \min_{\theta_0, \theta_1, \dots} \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \\ &= \min_{\theta_0, \theta_1, \dots} \frac{1}{2m} \sum_{i=1}^m (\theta_0 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \dots - y^{(i)})^2 \end{aligned}$$

- Necessary Condition: $\frac{\partial J(\theta_0, \theta_1, \dots)}{\partial \theta_i} = 0, 0 \leq i \leq n$

Normal Equation

$$\left\{ \begin{array}{l} \frac{\partial J(\theta_0, \theta_1, \dots)}{\partial \theta_0} = \frac{1}{m} \sum_{i=1}^m \left(\theta_0 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \dots + \theta_n x_n^{(i)} - y^{(i)} \right) = 0 \\ \frac{\partial J(\theta_0, \theta_1, \dots)}{\partial \theta_1} = \frac{1}{m} \sum_{i=1}^m \left(\theta_0 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \dots + \theta_n x_n^{(i)} - y^{(i)} \right) x_1^{(i)} = 0 \\ \dots \\ \frac{\partial J(\theta_0, \theta_1, \dots)}{\partial \theta_n} = \frac{1}{m} \sum_{i=1}^m \left(\theta_0 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \dots + \theta_n x_n^{(i)} - y^{(i)} \right) x_n^{(i)} = 0 \end{array} \right.$$

Normal Equation for Univariate Case

- Notions:

$$\bar{x} = \frac{1}{m} \sum_{i=1}^m x^{(i)}, \bar{y} = \frac{1}{m} \sum_{i=1}^m y^{(i)}, \overline{xy} = \frac{1}{m} \sum_{i=1}^m x^{(i)} y^{(i)}, \overline{x^2} = \frac{1}{m} \sum_{i=1}^m (x^{(i)})^2$$

- Equations:
$$\begin{cases} \frac{1}{m} \sum_{i=1}^m (\theta_0 + \theta_1 x^{(i)} - y^{(i)}) = \theta_0 + \bar{x} \theta_1 - \bar{y} = 0 \\ \frac{1}{m} \sum_{i=1}^m (\theta_0 + \theta_1 x^{(i)} - y^{(i)}) x^{(i)} = \theta_0 \bar{x} + \overline{x^2} \theta_1 - \overline{xy} = 0 \end{cases}$$

- Solutions:
$$\begin{cases} \theta_0 = \frac{\bar{y} * \overline{x^2} - \bar{x} * \overline{xy}}{\overline{x^2} - (\bar{x})^2} \\ \theta_1 = \frac{\overline{xy} - \bar{x} * \bar{y}}{\overline{x^2} - (\bar{x})^2} \end{cases}$$

Same Example

The training data contain some example measurements of the profit gained by opening an outlet in the cities with the population ranging between 30,000 and 100,000. The y-values are the profit measured in USD, and the x-values are the populations of the city. Each city population and profit tuple constitutes one training example in training dataset.

City Population (10^4) x	Profit (10^4) y
6.4862	6.5987
5.5277	9.1302
8.5186	13.662
7.0032	11.854

Solution with Normal Equation

$$\bar{x} = \frac{1}{4}(6.4862+5.5277 + 8.5186 + 7.0032) = 6.8839$$

$$\bar{y} = \frac{1}{4}(6.5987+9.1302 + 13.662 + 11.854) = 10.311$$

$$\begin{aligned}\overline{xy} &= \frac{1}{4}(6.4862 \times 6.5987 + 5.5277 \times 9.1302 + 8.5186 \times 13.662 + 7.0032 \times 11.854) \\ &= 73.167\end{aligned}$$

$$\begin{aligned}\overline{x^2} &= \frac{1}{4}(6.4862 \times 6.4862 + 5.5277 \times 5.5277 + 8.5186 \times 8.5186 + 7.0032 \times 7.0032) \\ &= 48.559\end{aligned}$$

$$\begin{cases} \theta_0 + \bar{x}\theta_1 - \bar{y} = 0 = \theta_0 + 6.8839\theta_1 - 10.311 = 0 \\ \bar{x}\theta_0 + \overline{x^2}\theta_1 - \overline{xy} = 6.8839\theta_0 + 48.55\theta_1 - 73.167 = 0 \end{cases} \Rightarrow \begin{cases} \theta_0 = -2.5471 \\ \theta_1 = 1.8679 \end{cases}$$

$$J(\theta_0, \theta_1) = 1/8 [(-2.5471 + 1.8679 \times 6.4862 - 6.5987)^2 + \dots] = 1.5598$$

Basic Knowledge about Matrix

$$\begin{array}{cccccc} a_{11}x_1 & + & a_{12}x_2 & + \cdots + & a_{1n}x_n & = b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + \cdots + & a_{2n}x_n & = b_2 \\ \vdots & & \vdots & & \vdots & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + \cdots + & a_{mn}x_n & = b_m \end{array}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

$$\begin{aligned} A\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &= A^{-1}\mathbf{b} \end{aligned}$$

Multivariate Linear Regression

$$\left\{ \begin{array}{l} \frac{\partial J(\theta_0, \theta_1, \dots)}{\partial \theta_0} = \frac{1}{m} \sum_{i=1}^m \left(\theta_0 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \dots + \theta_n x_n^{(i)} - y^{(i)} \right) = 0 \\ \frac{\partial J(\theta_0, \theta_1, \dots)}{\partial \theta_1} = \frac{1}{m} \sum_{i=1}^m \left(\theta_0 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \dots + \theta_n x_n^{(i)} - y^{(i)} \right) x_1^{(i)} = 0 \\ \dots \\ \frac{\partial J(\theta_0, \theta_1, \dots)}{\partial \theta_n} = \frac{1}{m} \sum_{i=1}^m \left(\theta_0 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + \dots + \theta_n x_n^{(i)} - y^{(i)} \right) x_n^{(i)} = 0 \end{array} \right.$$

$$\begin{aligned} \left(\sum_{i=1}^m 1 \cdot 1 \right) \theta_0 + \left(\sum_{i=1}^m x_1^{(i)} \cdot 1 \right) \theta_1 + \left(\sum_{i=1}^m x_2^{(i)} \cdot 1 \right) \theta_2 + \dots + \left(\sum_{i=1}^m x_n^{(i)} \cdot 1 \right) \theta_n &= \sum_{i=1}^m y^{(i)} \cdot 1 \\ \left(\sum_{i=1}^m 1 \cdot x_1^{(i)} \right) \theta_0 + \left(\sum_{i=1}^m x_1^{(i)} x_1^{(i)} \right) \theta_1 + \left(\sum_{i=1}^m x_2^{(i)} x_1^{(i)} \right) \theta_2 + \dots + \left(\sum_{i=1}^m x_n^{(i)} x_1^{(i)} \right) \theta_n &= \sum_{i=1}^m y^{(i)} x_1^{(i)} \\ \dots & \\ \left(\sum_{i=1}^m 1 \cdot x_n^{(i)} \right) \theta_0 + \left(\sum_{i=1}^m x_1^{(i)} x_n^{(i)} \right) \theta_1 + \left(\sum_{i=1}^m x_2^{(i)} x_n^{(i)} \right) \theta_2 + \dots + \left(\sum_{i=1}^m x_n^{(i)} x_n^{(i)} \right) \theta_n &= \sum_{i=1}^m y^{(i)} x_n^{(i)} \end{aligned}$$

Matrix Form of Normal Equation

$$A = \begin{bmatrix} \sum_{i=1}^m 1 \cdot 1 & \sum_{i=1}^m x_1^{(i)} \cdot 1 & \sum_{i=1}^m x_2^{(i)} \cdot 1 & \dots & \sum_{i=1}^m x_n^{(i)} \cdot 1 \\ \sum_{i=1}^m 1 \cdot x_1^{(i)} & \sum_{i=1}^m x_1^{(i)} x_1^{(i)} & \sum_{i=1}^m x_2^{(i)} x_1^{(i)} & \dots & \sum_{i=1}^m x_n^{(i)} x_1^{(i)} \\ \sum_{i=1}^m 1 \cdot x_n^{(i)} & \sum_{i=1}^m x_1^{(i)} x_n^{(i)} & \sum_{i=1}^m x_2^{(i)} x_n^{(i)} & \dots & \sum_{i=1}^m x_n^{(i)} x_n^{(i)} \end{bmatrix} \quad b = \begin{bmatrix} \sum_{i=1}^m y^{(i)} \cdot 1 \\ \sum_{i=1}^m y^{(i)} x_1^{(i)} \\ \dots \\ \sum_{i=1}^m y^{(i)} x_n^{(i)} \end{bmatrix}$$

$$\begin{aligned} A\theta &= \mathbf{b} \\ \theta &= A^{-1}\mathbf{b} \end{aligned}$$

Matrix Form of Normal Equation

- Matrix Form of solution θ in terms of matrices X and y ?

$$X = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_j^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & \dots & x_j^{(2)} & \dots & x_n^{(2)} \\ & & \ddots & & & \\ 1 & x_1^{(m)} & \dots & x_j^{(m)} & \dots & x_n^{(m)} \end{bmatrix}, \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}, \quad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

Features

Target variables

$$\begin{aligned} (X^T X) \theta &= X^T y \\ \theta &= (X^T X)^{-1} X^T y \end{aligned}$$