AI 330: Machine Learning

Fall 2023

Dr. Wessam EL-Behaidy

Associate Professor, Computer Science Department,

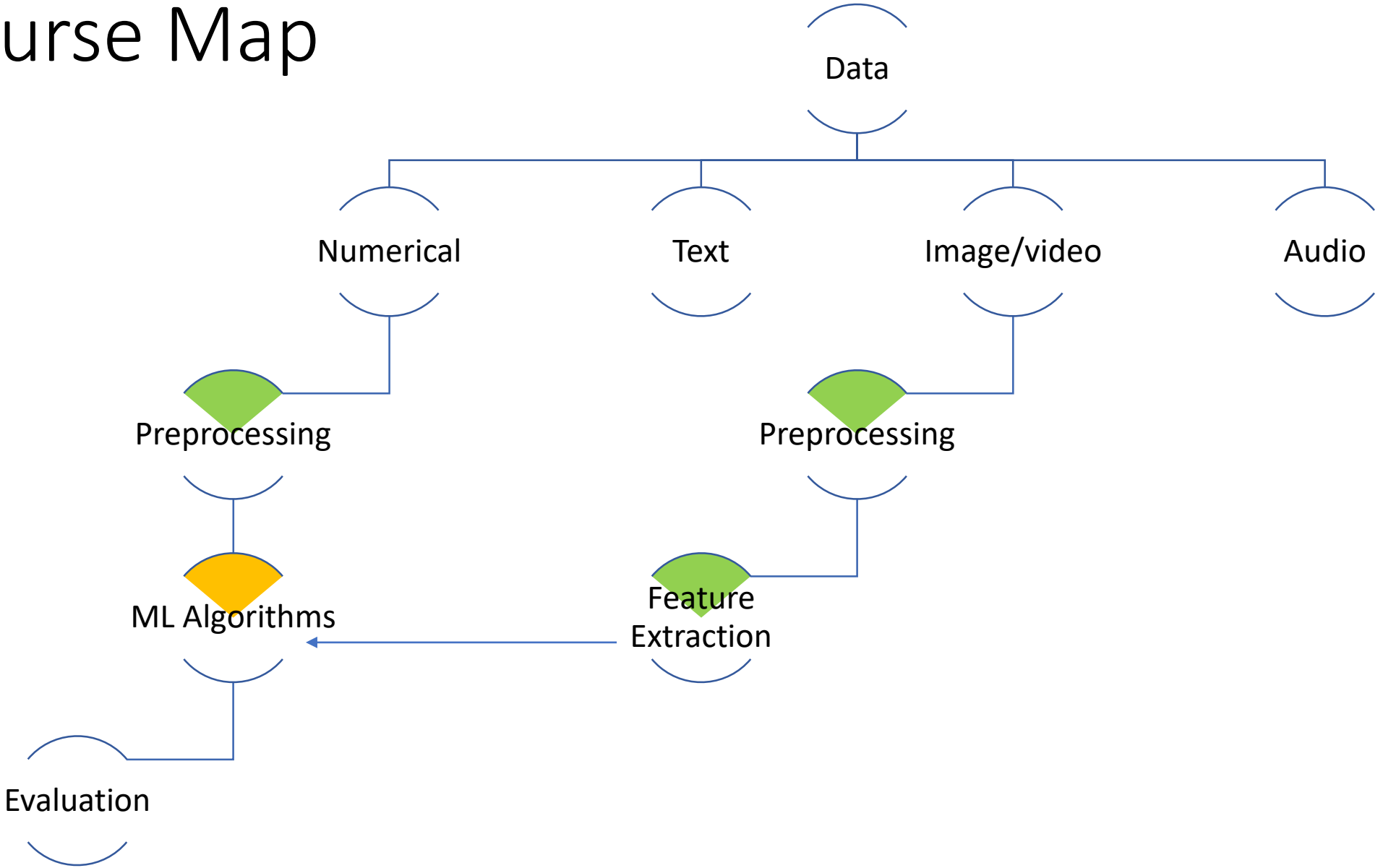
Faculty of Computers and Artificial Intelligence,

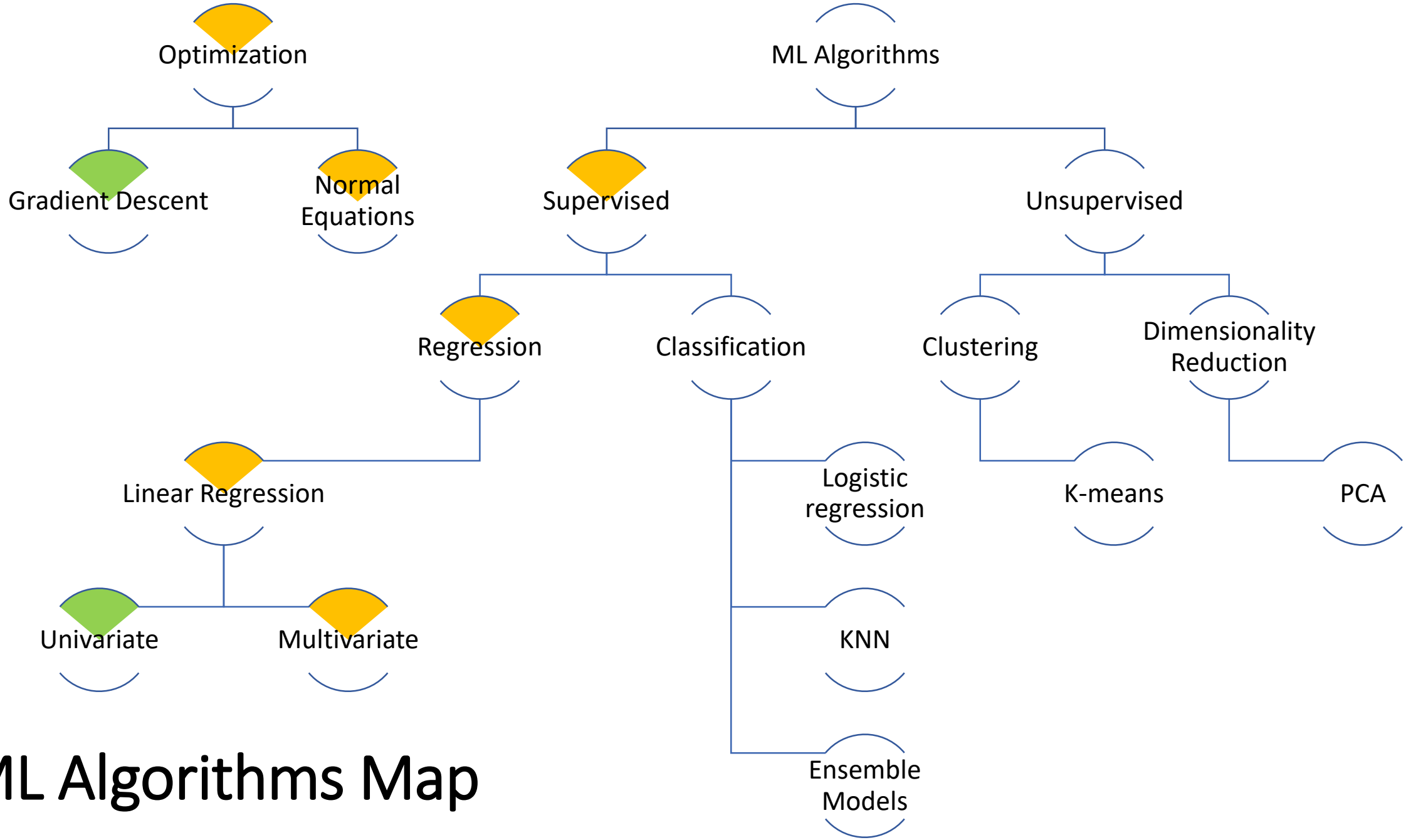
Helwan University.

Dr. Ensaf Hussein

Associate Professor, Computer Science Department,
Faculty of Computers and Artificial Intelligence,
Helwan University.

Course Map


Data
Numerical
Text
Image/video
Audio
Preprocessing
Preprocessing
ML Algorithms
Feature Extraction
Evaluation



ML Algorithms Map

Lecture 5
Multivariate Linear Regression
&
Normal Equation

Slides of:
Machine Learning Specialization https://www.coursera.org/specializations/machine-learning-introduction
at Stanford University (**Andrew Ng**)

Review: Matrices and vectors

Vector: Special case of the matrix

n x 1 matrix

1 dimention
1 vector

$\mathbb{R}^4$

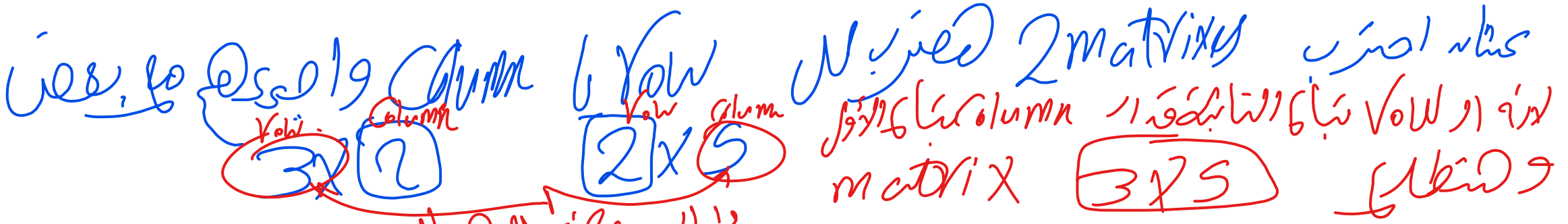
- $y = \begin{bmatrix} 4 \\ 50 \\ 3 \\ 25 \end{bmatrix}$ $4 - dimentional\ vector\ n = 4$

1-indexed vs 0-indexed:

$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$ $y = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}$

$y_i = i^{\text{th}}$ element.

$y_2 = 50$ $y_2 = 3$



Matrix Multiplication

Column's row ضرب

$$A \times x = y$$

$$\begin{bmatrix} 5 & 1 \\ 2 & 6 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 15 \\ 34 \\ 26 \end{bmatrix}$$

3 × 2 **2 × 1** **3 × 1 matrix → 3 dim vector**

To get y_i, *multiply* $A's\ i^{th}$ *row with elements of vector* x, *and add them up.*

- $y_1 = 5\times2 + 1\times5 = 15$
- $y_2 = 2\times2 + 6\times5 = 34$
- $y_3 = 3\times2 + 4\times5 = 26$

An **m x n matrix** multiplied by an **n x 1 vector** results in an **m x 1 vector**.

$$A \times B = C$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 1 & 2 \\ 0 & 5 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 7 & 15 \\ 16 & 39 \end{bmatrix}$$

2 × 3 **3 × 2** **2 × 2 matrix**

- C_{11}=1+0+6 = 7
- C_{21} =4 + 0+ 12 = 16
- C_{12} = 2+10+3= 15
- C_{22} =8+25+6= 39

An **m x n matrix** multiplied by an **n x o matrix** results in an **m x o** matrix.

Matrix Multiplication Properties

- Matrices are **not commutative**: $A \times B \neq B \times A$
- Matrices are **associative**: $(A \times B) \times C = A \times (B \times C)$
- Identity Matrix $I, or\ I_{n \times n}$ for any A, $A.\ I = I.\ A = A$

- $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Inverse and Transpose

- The **inverse** of a matrix A $(m \times m)$ is denoted A^{-1} . Multiplying by the inverse results in the identity matrix. $A \times A^{-1} = 1$

  A non square matrix does not have an inverse matrix.

$$\begin{bmatrix} 3 & 4 \\ 2 & 16 \end{bmatrix} \times \begin{bmatrix} 0.4 & -0.1 \\ -0.05 & 0.075 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

identity

- The **transposition** $B_{ij} = A_{ji}$ of a matrix is like rotating the matrix 90° in clockwise direction and then reversing it.
- $A_{ij} = A_{ij}^{T}$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}^{T} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

m x n matrix

n x m matrix

Regression Problem:
Univariate Linear Regression Example

Linear Regression with One Variable

Recap:

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

Recap: Gradient Descent

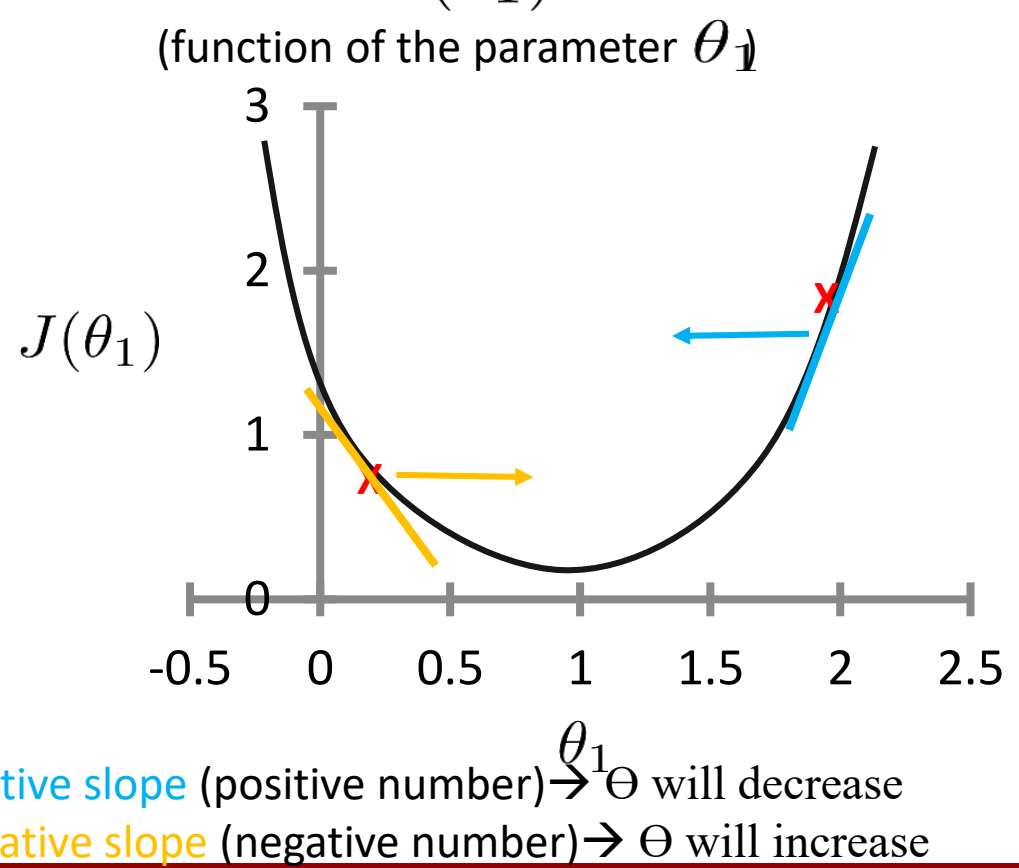
- We make steps down the cost function in the direction with the steepest descent, and the size of each step is determined by the parameter **α**, which is called the **learning rate.**
- The gradient descent algorithm is:
- Repeat until **convergence** {

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

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

}

Learning rate (step size)



Positive slope (positive number)→ Θ will decrease

Negative slope (negative number)→ Θ will increase

Recap: Gradient Descent for Linear Regression

- When specifically applied to the case of **linear regression**, a new form of the gradient descent equation can be derived. We can substitute our actual cost function and our actual hypothesis function and modify the equation to:

- Start by initializing the parameters θ_0, θ_1 **randomly**

- Repeat until convergence { // until error is small

- Predicted values with linear regression hypothesis.

- Calculate the cost function.

- If cost is large, update parameters using GD

Should be done simultaneously

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

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \cdot x^{(i)} \quad \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

}

Hypothesis using Matrix Product

- $h_\theta(x) = \theta_0 + \theta_1 x$

$$x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \in \mathbb{R}^2 \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} \in \mathbb{R}^2$$

For convenience of notation, define $x_0 = 1$

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 = [\theta_0 \theta_1] \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \theta^T x$$

Linear Regression with one variable Example

- Suppose we are given a dataset, and we need to find the hypothesis of linear regression:

Experience (x)	Salary (y) (in lakhs)
2	3
6	10
5	4
7	13

https://www.geeksforgeeks.org/mathematical-explanation-for-linear-regression-working/

Iteration 1

Experience (x)	Salary (y) (in lakhs)
2	3
6	10
5	4
7	13

a) In the start, θ_0 and θ_1 values are randomly chosen.

- Let us suppose, $\theta_0 = 0$ and $\theta_1 = 0$.

b) Predicted values after iteration 1 with Linear regression hypothesis.

$$h_\theta = \theta_0 x_0 + \theta_1 x_1 = [\theta_0 \theta_1] \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$$

$$h_\theta = [\theta_0 \theta_1] \begin{bmatrix} x_0 & x_0 & x_0 & x_0 \\ x_1 & x_2 & x_3 & x_4 \end{bmatrix}$$

$$h_\theta = [0 \;\; 0] \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 6 & 5 & 7 \end{bmatrix}$$

$$h_\theta = [0 \; 0 \; 0 \; 0]$$

One training example

Iteration 1

Experience (x)	Salary (y) (in lakhs)
2	3
6	10
5	4
7	13

c) Cost function Error (m=No. of samples = 4)

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}[h_\theta(x_i) - y_i]^2$$

$$= \frac{1}{2*4}[(0-3)^2+(0-10)^2+(0-4)^2+(0-13)^2]$$

$$= \frac{1}{8}[9+100+16+169]$$

$$= 36.75$$

Iteration 1

Experience (x)	Salary (y) (in lakhs)
2	3
6	10
5	4
7	13

d) Gradient Descent – Updating θ_0 value

Here, j=0 , α = 0.001

$$\theta_0 = \theta_0 - \frac{\alpha}{m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)$$

$$= 0 - \frac{0.001}{4}[(0-3) + (0-10) + (0-4) + (0-13)]$$

$$= -\frac{0.001}{4}[(-3) + (-10) + (-4) + (-13)]$$

$$= -\frac{0.001}{4}[-30]$$

$$= 0.0075$$

new θ_0

Algorithm GD for linear regression:

repeat until convergence {

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

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

}

Iteration 1

Experience (x)	Salary (y) (in lakhs)
2	3
6	10
5	4
7	13

e) Gradient Descent – Updating θ_1 value

Here, j=1

$$\theta_1 = \theta_1 - \frac{\alpha}{m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i).x_i$$

$$= 0 - \frac{0.001}{4}[(0-3)2 + (0-10)6 + (0-4)5 + (0-13)7]$$

$$= -\frac{0.001}{4}[(-6) + (-60) + (-20) + (-91)]$$

$$= -\frac{0.001}{4}[-177]$$

$$= 0.04425$$

Algorithm GD for linear regression:

repeat until convergence {

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

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

}

Iteration 2

Experience (x)	Salary (y) (in lakhs)
2	3
6	10
5	4
7	13

Iteration 2 – $\boldsymbol{\theta_0 = 0.0075}$ $\boldsymbol{and}$ $\boldsymbol{\theta_1 = 0.04425}$

- Predicting values after iteration 1 with linear regression hypothesis

$$h_\theta = [\theta_0 \quad \theta_1] \begin{bmatrix} x_0 & x_0 & x_0 & x_0 \\ x_1 & x_2 & x_3 & x_4 \end{bmatrix}$$

$$= [0.0075 \quad 0.04425] \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 6 & 5 & 7 \end{bmatrix}$$

$$= [0.096 \quad 0.273 \quad 0.22875 \quad 0.31725]$$

Iteration 2

- Similar to iteration no. 1 performed above
  - we will again calculate Cost function and update θ_j values using Gradient Descent.
  - We will keep on iterating until Cost function doesn't reduce further **(converge)**. At that point, model achieves best θ values.
  - Using these θ values in the model hypothesis will give the best prediction results.

Testing

- If we have the experience years and the hypotheses (h_θ), and you need to predict the salaries of these persons.

$$h_\theta(x) = -7 + 3\,x$$

Experiences:
21
31
45
11

Matrix **4x2** :

$$\begin{bmatrix} 1 & 21 \\ 1 & 31 \\ 1 & 45 \\ 1 & 11 \end{bmatrix} \times \begin{bmatrix} -7 \\ 3 \end{bmatrix} =$$

2x1

Prediction **4x1 matrix**: $h_\theta(21)$

$$\begin{bmatrix} -7 \times 1 + 3 \times 21 \\ -7 \times 1 + 3 \times 31 \\ -7 \times 1 + 3 \times 45 \\ -7 \times 1 + 3 \times 11 \end{bmatrix} = \begin{bmatrix} 56 \\ 86 \\ 128 \\ 26 \end{bmatrix}$$

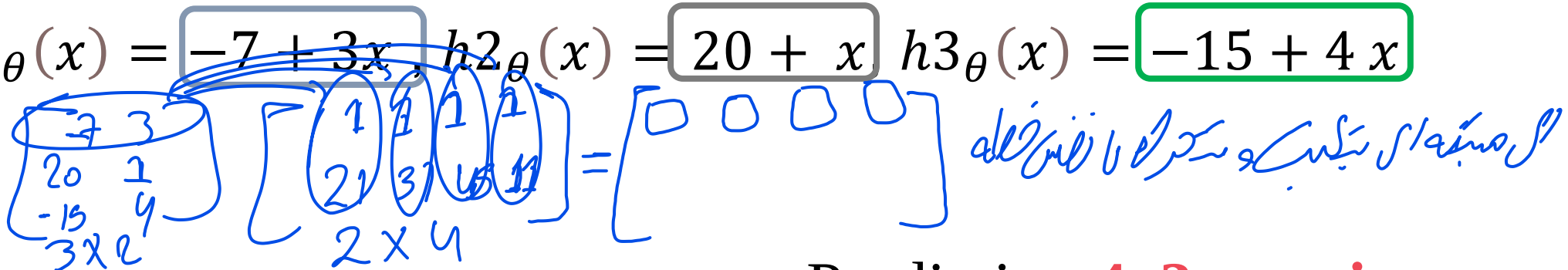
Testing

Experiences:

21
31
45
11

- If we have the experiences and more than one hypotheses (h_θ):

$$h1_\theta(x) = -7 + 3x \quad h2_\theta(x) = 20 + x \quad h3_\theta(x) = -15 + 4x$$



Matrix **4x2**:

$$\begin{bmatrix} 1 & 21 \\ 1 & 31 \\ 1 & 45 \\ 1 & 11 \end{bmatrix} \times \begin{bmatrix} -7 & 20 & -15 \\ 3 & 1 & 4 \end{bmatrix} =$$

Prediction **4x3 matrix**:

$$\begin{bmatrix} 56 & 41 & 69 \\ 86 & 51 & 109 \\ 128 & 65 & 165 \\ 26 & 31 & 29 \end{bmatrix}$$

2x3

$h1_\theta$

Regression Problem:

Multivariate Linear Regression

Linear Regression with multiple Variable

multiple Variables

more than one Feature

Multiple variables (Features)

- Linear regression with multiple variables is also known as "multivariate linear regression".

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

$\boldsymbol{Size\ (feet)^2}$ $\boldsymbol{x}$	$\boldsymbol{Price\ (\$1000)}$ $\boldsymbol{y}$
2104	460
1416	232
1534	315
852	178

Could the prediction be more accurate if we add #of rooms?

Multiple variables (Features)

$Size\ (feet)^2$ 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
…	…	…	…	…

$\mathbf{m}$, $number\ of\ training\ examples$

$\mathbf{n}$, number of features

$x^{(i)}$, the input (features) of the i^{th} training example

$x_j^{(i)}$, value of feature j in the i^{th} training example

row

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

row

column

$$x_4^{(3)} = 30$$

Hypothesis

Size $(feet)^2$ ⇧ 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
...	...	...	...	...

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

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$$

$$h_\theta(x) = 80 \quad 0.1x_1 \quad 0.01x_2 \quad 3x_3 \quad 2x_4$$

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

$For\ convenience\ of\ notation, define\ x_0 = 1$

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ . \\ . \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ . \\ . \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

Hypothesis for Multiple Features

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ . \\ . \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ . \\ . \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

$$h_\theta = \sum_{j=0}^{n} \theta_j x_j = \theta^T x = \begin{bmatrix} \theta_0 & \theta_1 \ldots\ldots & \theta_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ . \\ . \\ x_n \end{bmatrix}$$

Cost Function for Multiple Variables

$x_0 = 1$ ← always

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

$Parameters: \theta_0, \quad \theta_1, \ldots\ldots, \quad \theta_n$

θ is $n+1$ dimension vector

from 1 — Hidden — Real

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

Hypothesis Function

والتعديل الى الجمع

$$= \frac{1}{2m}\sum_{i=1}^{m}\left(\theta^T x^{(i)} - y^{(i)}\right)^2$$

$$= \frac{1}{2m}\sum_{i=1}^{m}\left(\left(\sum_{j=0}^{n}\theta_j x_j^{(i)}\right) - y^{(i)}\right)^2$$

GD for Multiple Variables

- The gradient descent equation itself is generally the same form; we just have to repeat it for our 'n' features

Repeat until convergence: {

$$\theta_0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_j} J(\theta), \quad \text{simultaneously updates for every } j = 0,1,2, \dots, n$$

}

For n=1:

Repeat {

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

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

(simultaneously update θ_0, θ_1)

}

For (n>1):

Repeat until convergence: {

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

$\theta_1 := \dots$

$\theta_2 := \dots$, simultaneously updates θ_j

$\theta_3 := \dots$ for every $j = 0,1,2, \dots, n$

$\theta_n := \dots$

}

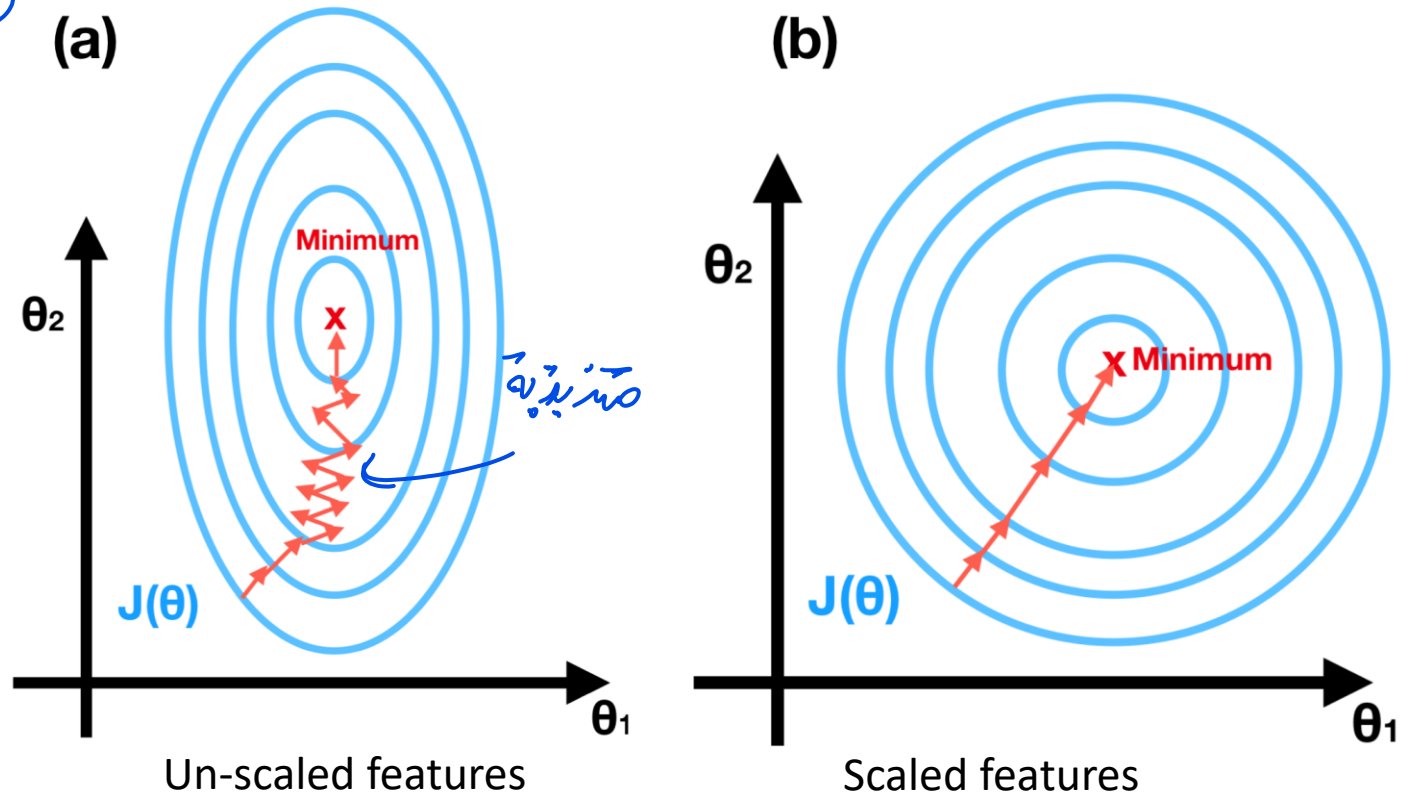
Gradient Descent in Practice I - Feature Scaling

- We can speed up gradient descent by having each of our input values in roughly the **same range**.
- Because θ will:
  - descend quickly on small ranges
  - descend slowly on large ranges, and
  - oscillate inefficiently down to the optimum when the variables are very uneven.
- The way to prevent this is to modify the ranges of our input variables so that they are all roughly the same.

Gradient Descent in Practice I - Feature Scaling

Make sure features are on same scale using:

- Mean Normalization
- Z-Score Normalization



Un-scaled features

Scaled features

https://www.blog.nipunarora.net/ml_multi_variate_linear_regression/

Gradient Descent in Practice I - Feature Scaling

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

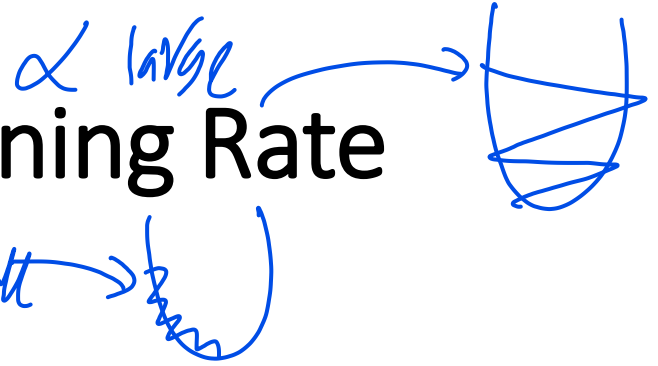
(**Do not apply to $x_0 = 1$**)

Rule a thumb regarding acceptable ranges

- -3 to +3 is generally fine - any bigger bad
- -1/3 to +1/3 is ok - any smaller bad

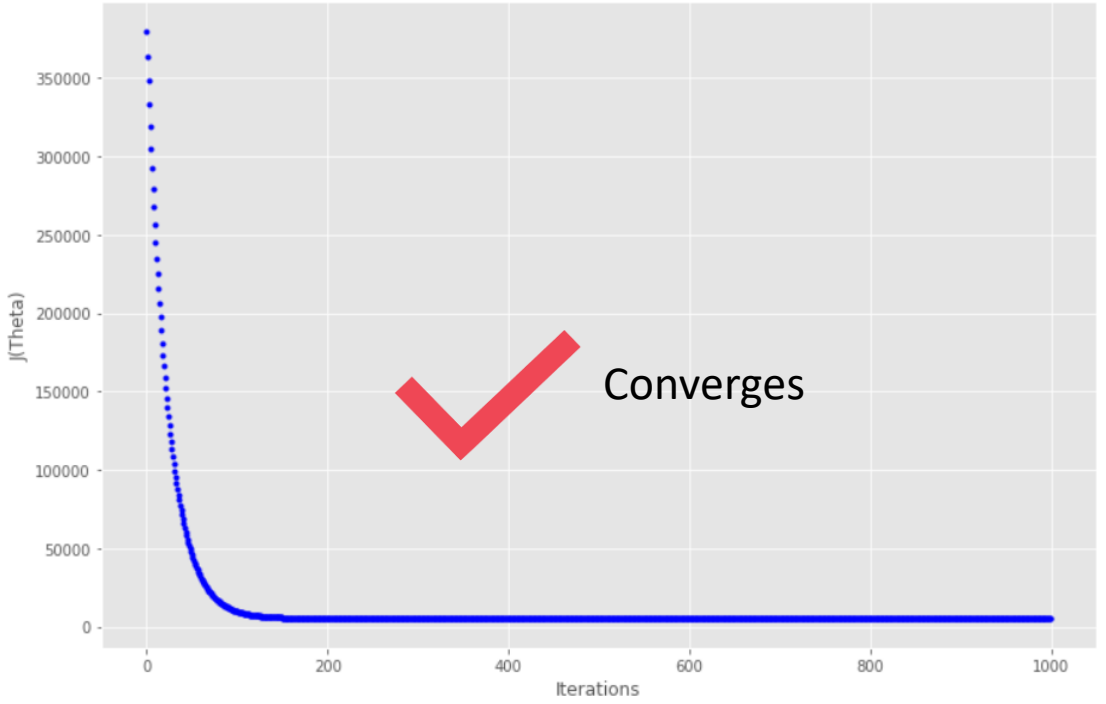
https://www.blog.nipunarora.net/ml_multi_variate_linear_regression/

Gradient Descent in Practice II - Learning Rate



- **Debugging gradient descent.**
- Make a plot with *number of iterations* of gradient descent on the $x - axis$ and the cost function $J(\theta)$ on the $y - axis$.
- $J(\theta)$ should decrease after each iteration else decrease α

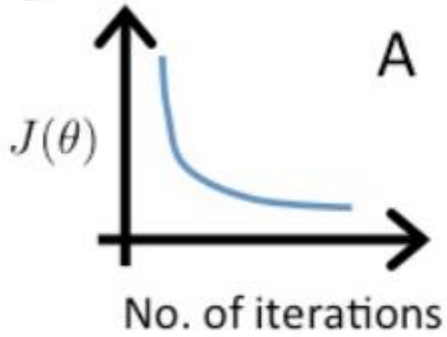
Automatic convergence test. Declare convergence if $J(\theta)$ decreases by less than ε in one iteration, where ε is some small value such as 10^{-3}. However in practice it's difficult to choose this threshold value.



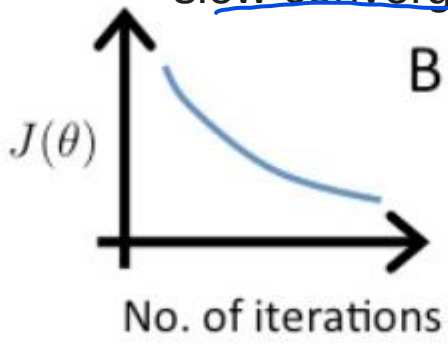
Gradient Descent in Practice II - Learning Rate

It has been proven that if learning rate α is sufficiently small, then $J(\theta)$ will decrease on every iteration. When Gradient Descent can't decrease the cost-function anymore and remains more or less on the same level, we say it has converged.

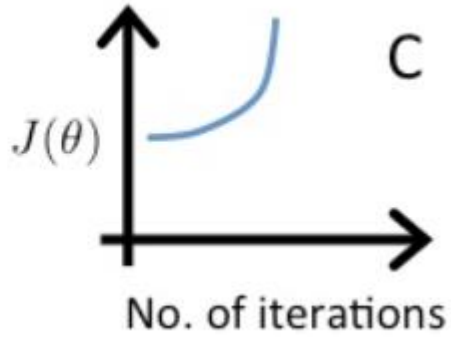
Note:
If you see in the plot that your learning curve is just going up and down, without really reaching a lower point, you also should try to decrease the learning rate.



If α is too small:
slow convergence.



If α is too high:
the cost function is increasing



Features choice /selection

- We can improve our features and the form of our hypothesis function.
- We can **combine** multiple features into one. For example, we can combine x_1 and x_2 into a new feature x_3 by taking $x_1 . x_2$
- Ex: x_1 is length and x_2 is width of a house, we can combine them into a new feature area= length x width

Detecting Redundancies using Correlation Analysis

- *Consider two attributes A and B coming from two different data sources. The correlation between attributes A and B can be measured by the correlation coefficient $r_{A,B}$:*

$$r_{A,B} = \sum_{i=1}^{n} (A_i - \overline{A})(B_i - \overline{B}) / \{(n-1)\sigma_A \sigma_B\}$$

Where

$$\overline{A} = \left(\frac{\sum_{i=1}^{n} A_i}{n} \right) \qquad \overline{B} = \left(\frac{\sum_{i=1}^{n} B_i}{n} \right)$$

$$\sigma_A = \sqrt{\frac{\sum_{i=1}^{n} (A_i - \overline{A})^2}{n-1}} \quad \text{and} \quad \sigma_B = \sqrt{\frac{\sum_{i=1}^{n} (B_i - \overline{B})^2}{n-1}}$$

- Where **n** is the number of data items in A or B (assuming they have the same number of data items), $\overline{A}$ and $\overline{B}$ are the mean values of A and B, and σ_A and σ_B are the standard deviations of A and B

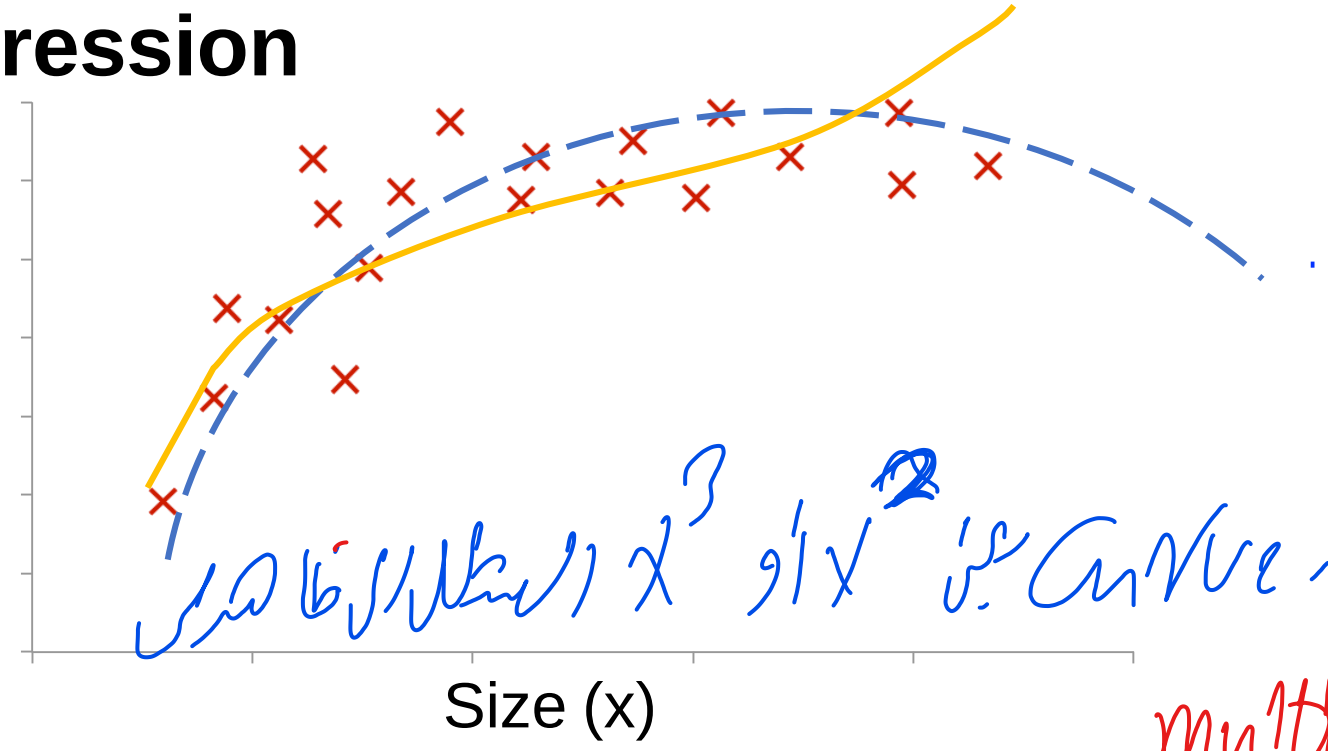
Detecting Redundancies using Correlation Analysis

- If the $r_{A,B} > 0$ ➔ A and B are positively correlated
  - ➔ the values of A increase as the values of B increase
    - ➔ **either A or B** may be removed as a redundancy but not both of them

- If the $r_{A,B} = 0$ ➔ A and B are independent (no redundancy)
  - ➔ we cannot remove any of them

- If the $r_{A,B} < 0$ ➔ A and B are negatively correlated
  - ➔ the values of one attribute increase as the values of the other decrease
  - ➔ we cannot remove any of the attributes

Polynomial regression



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$
$$= \theta_0 + \theta_1 (size) + \theta_2 (size)^2 + \theta_3 (size)^3$$

$$x_1 = (size)$$
$$x_2 = (size)^2$$
$$x_3 = (size)^3$$

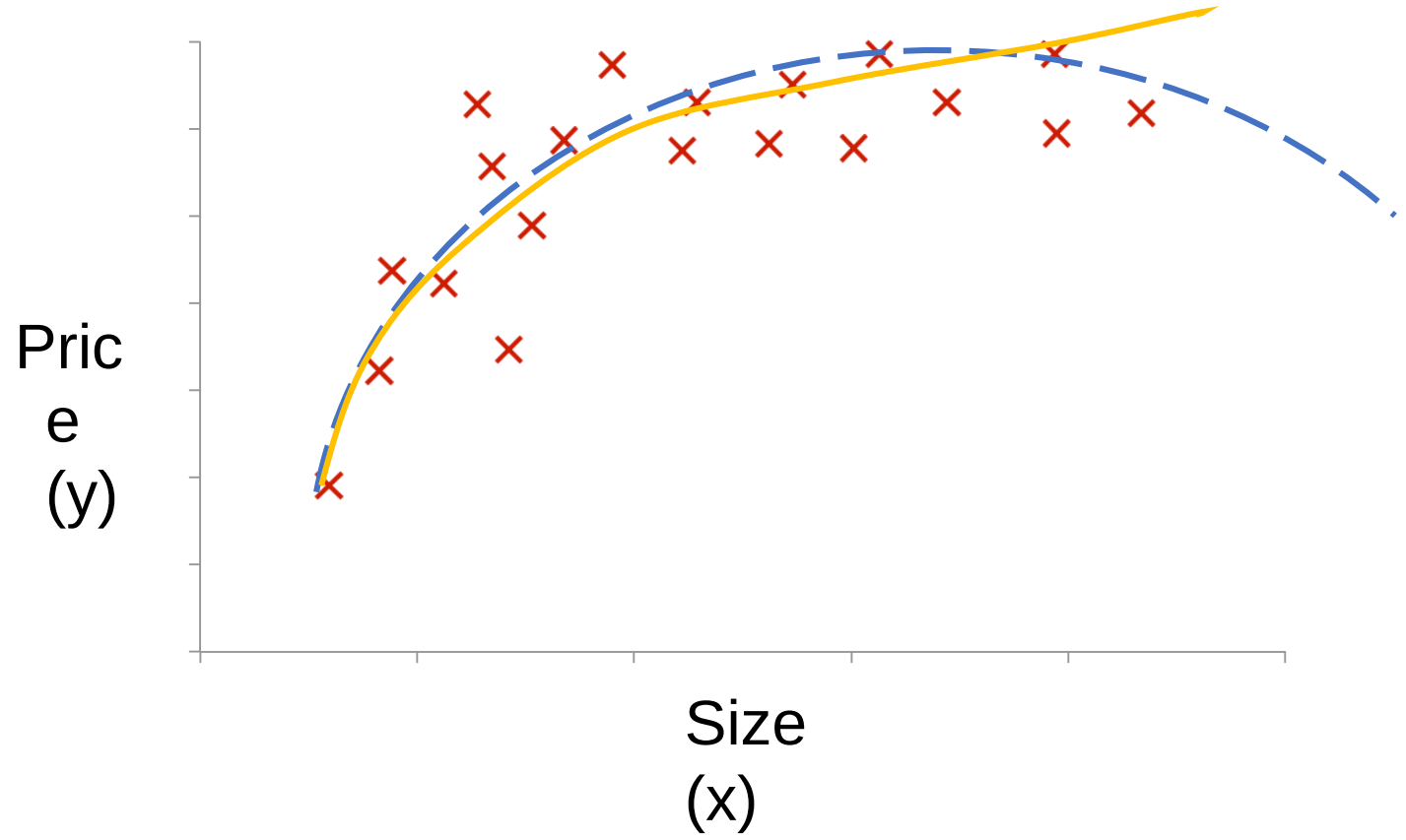
Size: 1- 1000

Size^2: 1- 10^6

Size^3: 1- 10^9

feature scaling becomes very important.

Choice of features



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

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

Optimization Technique:
Normal Equations

Better values for θ

- To solve for θ analytically → normal equation
- $\theta \in \mathbb{R}^{n+1}$

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

Minimize $J(\theta)$

$\frac{\partial}{\partial \theta_j} J(\theta)$= …=0 (set to zero)

for $every\ j$ solve for $\theta_0, \theta_1, \dots, \theta_n$

Example: m=4

x_0	$Size\ (feet)^2$ x_1	$Number\ of\ bedrooms$ x_2	$Number\ of\ floors$ x_3	$Age\ of\ home\ (years)$ x_4	$Price\ (\$1000)$ 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

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}$$

$$y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

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

Features matrix $m \times (n+1)$

$m - \dim vector$

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

$m\ samples\ , n\ features$

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

$$X = \begin{bmatrix} \cdots (x^{(1)})^T \cdots \\ (x^{(2)})^T \\ \cdot \\ \cdots \quad \cdot \quad \cdots \\ \cdot \\ \\ \cdots (x^{(m)})^T \cdots \end{bmatrix}$$

Design matrix
m x (n+1)

$Example: one\ feature$

$$x^{(i)} = \begin{bmatrix} 1 \\ x_1^{(i)} \end{bmatrix}$$

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

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

Gradient Descent Vs. Normal Equation

Gradient Descent	Normal Equation
Need to choose α	No need to choose α
Needs many iterations	No need to iterate
$O\ (kn^2)$	$O\ (n^3)$
Works well when n is large	Slow if n is very large

Example 2:

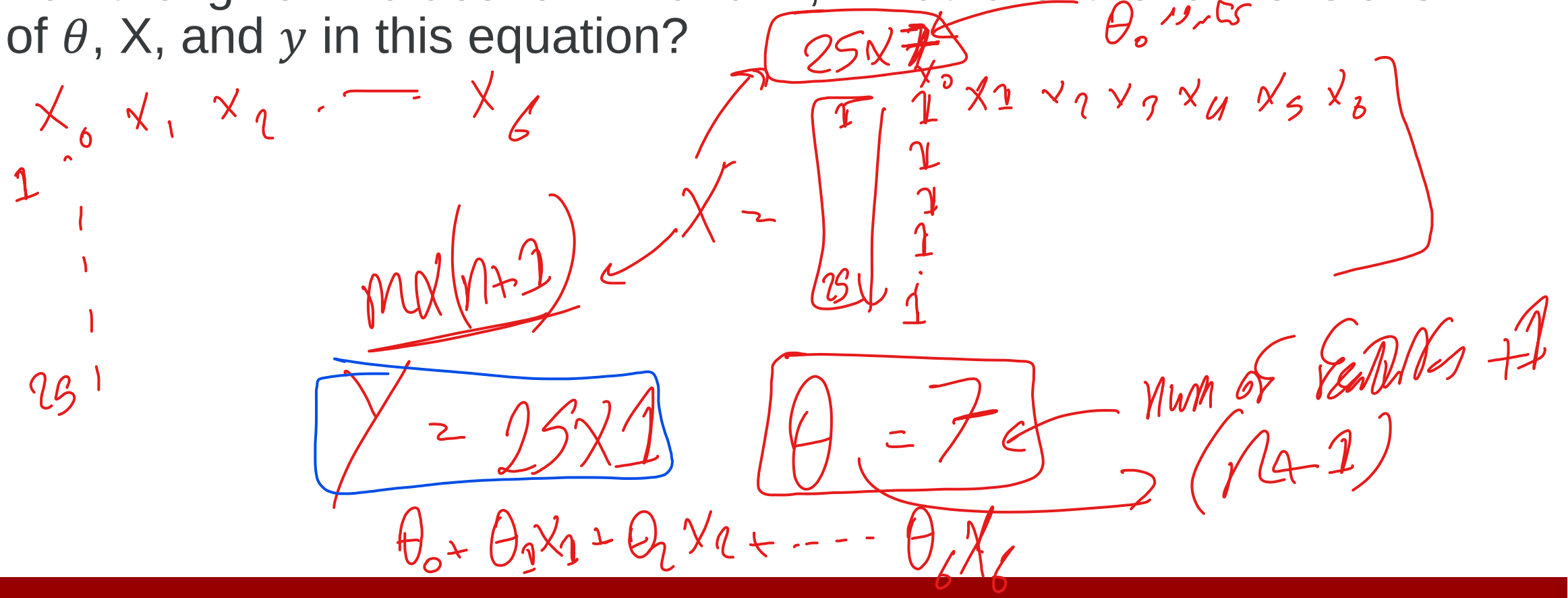
x_0	Age (x1)	Height in cm (x2)	Weight in Kg (y)
1	4	89	16
1	9	124	28
1	5	103	20

- What is X (design matrix) and y?

$$X=\begin{bmatrix}1 & 4 & 89\\ 1 & 9 & 124\\ 1 & 5 & 103\end{bmatrix} \qquad y=\begin{bmatrix}16\\ 28\\ 20\end{bmatrix}$$

Check?

- Suppose you have $m = 25$ training examples with $n = 6$ features. The normal equation is $\theta = (X^T X)^{-1} X^T y$
- For the given values of m and n, what are the dimensions of θ, X, and y in this equation?



Thanks