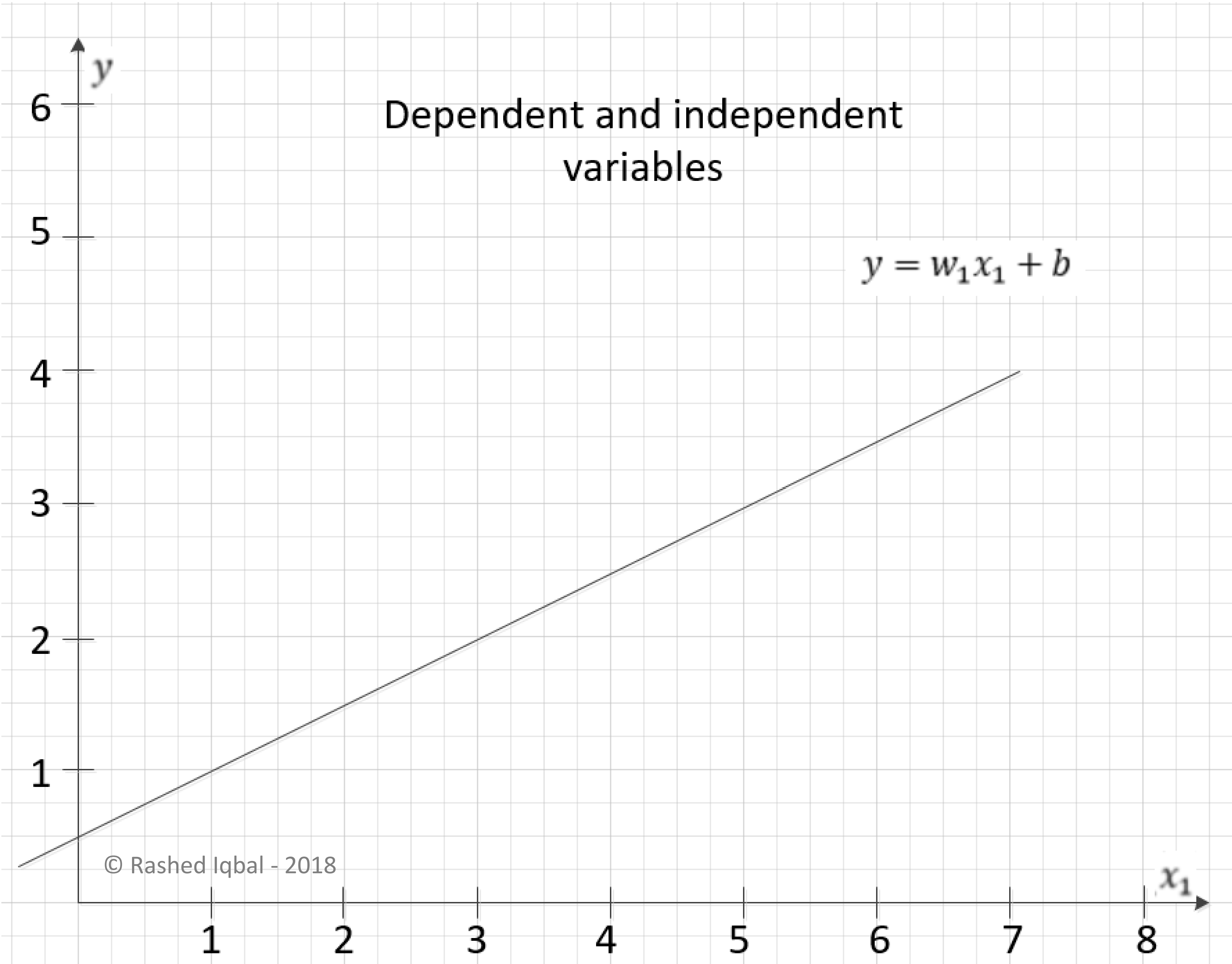
Linear Regression – An Introduction

Rashed Iqbal, Ph.D.

We often make predictions in life based upon our past experience. For example, if for 35 houses sold in a town in the last month, we know their sold prices and their features (bedrooms, bathrooms, lot size, living space, age, quality of construction, etc.), we may be able to predict sold price of another house that just came on for sale with some confidence. In this case, the data that we have is called Experience that consists of multiple training examples.

Let us assume we have *m* training examples $[y, X]$, $X = \{x_1, x_2, x_3, \dots, x_n\}$. Here the variable y is dependent on a set of *n* independent features where, by way of the example, y can be price of a house, and X represents its features such as square footage, lot size, number of bedrooms, number of bathrooms, age, quality of construction, etc. The problem of Linear Regression consists of finding a line that fits best to *m* points (or training examples) in the *n*-dimensional hyperspace which, in other words, means finding *n* + 1 parameters $w_1, w_2, w_3, \dots, w_n$ and b given a set of *m* training examples $[y^{(i)}, X^{(i)}]$ $i \in \{1,2, \dots, m\}$ or Experience.

Sq. Feet	BR	Lot (sq. ft.)	Stories	Sold Price
1000	2	5000	1	$300,000.00
2000	3	5000	1	$400,000.00
2500	4	7500	1	$700,000.00
3500	5	7500	1	$800,000.00
3250	5	5000	2	$900,000.00


Dependent and independent variables
$y = w_1 x_1 + b$
y
x_1
6
5
4
3
2
1
1
2
3
4
5
6
7
8
© Rashed Iqbal - 2018

Linear Regression with one Feature:

Let us consider a simple problem first with a feature x_1 and the dependent variable y. In this case, by a best line we mean a line with slope w_1 and the intercept b in such a way that the line minimizes sum of squared vertical distances from it. This line can be written as:

$$y = w_1 x_1 + b$$

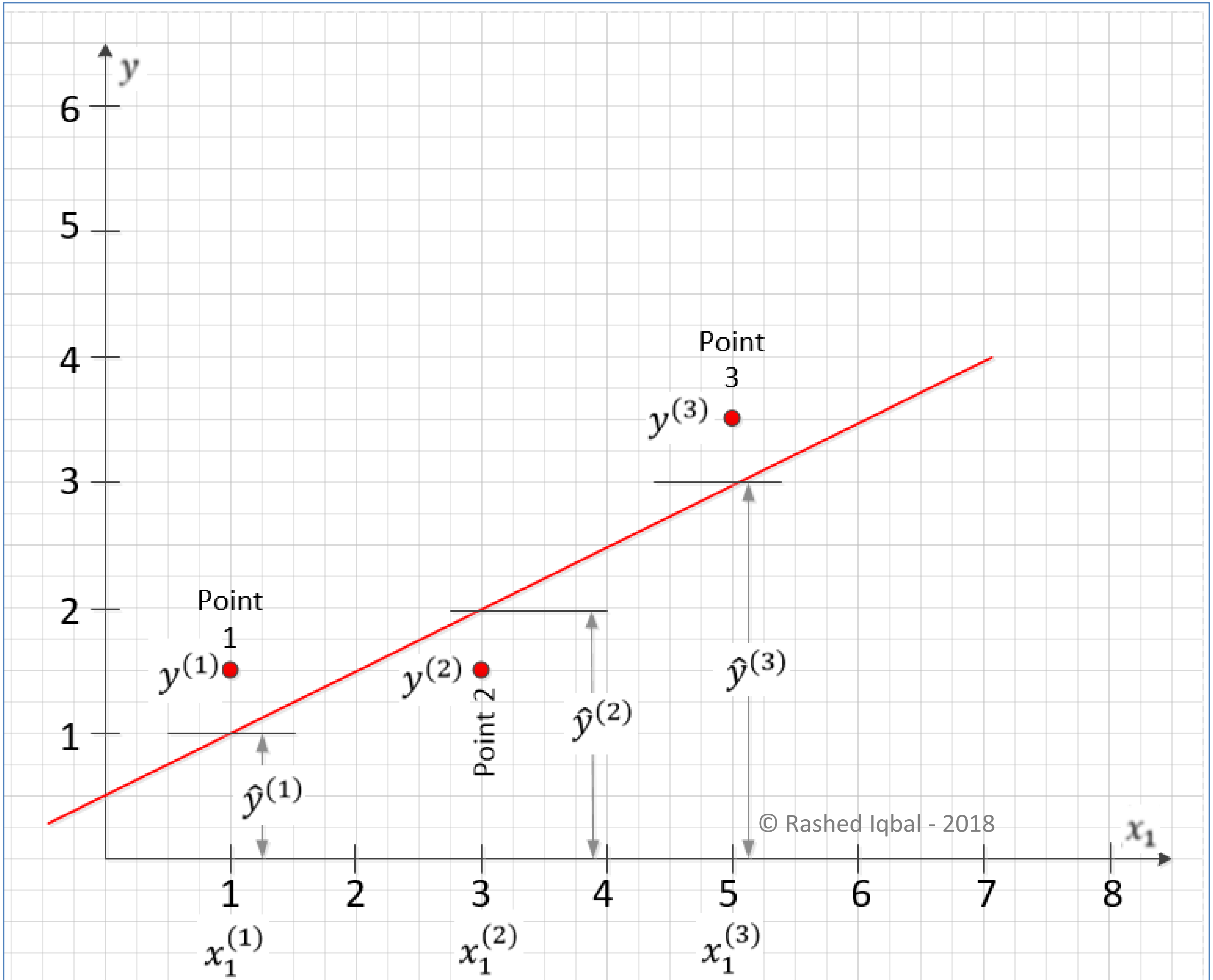
Let us assume only three training examples: $\left[y^{(i)}, x_1^{(i)}\right],\ i \in \{1,2,3\}$. Given some b and w_1, we can determine estimates for y's:

$$\hat{y}^{(1)} = w_1\, x_1^{(1)} + b$$

$$\hat{y}^{(2)} = w_1\, x_1^{(2)} + b$$

$$\hat{y}^{(3)} = w_1\, x_1^{(3)} + b$$

As all three training points do not lie on a straight line, $\hat{y}^{(1)}$, $\hat{y}^{(2)}$, and $\hat{y}^{(3)}$ are different than $y^{(1)}$, $y^{(2)}$, and $y^{(3)}$ as shown in the figure.



y
6
5
4
3
2
1
Point 3
$y^{(3)}$
Point 1
$y^{(1)}$
$y^{(2)}$
Point 2
$\hat{y}^{(1)}$
$\hat{y}^{(2)}$
$\hat{y}^{(3)}$
© Rashed Iqbal - 2018
x_1
1
2
3
4
5
6
7
8
$x_1^{(1)}$
$x_1^{(2)}$
$x_1^{(3)}$

The difference distance between the Points 1 and the line:

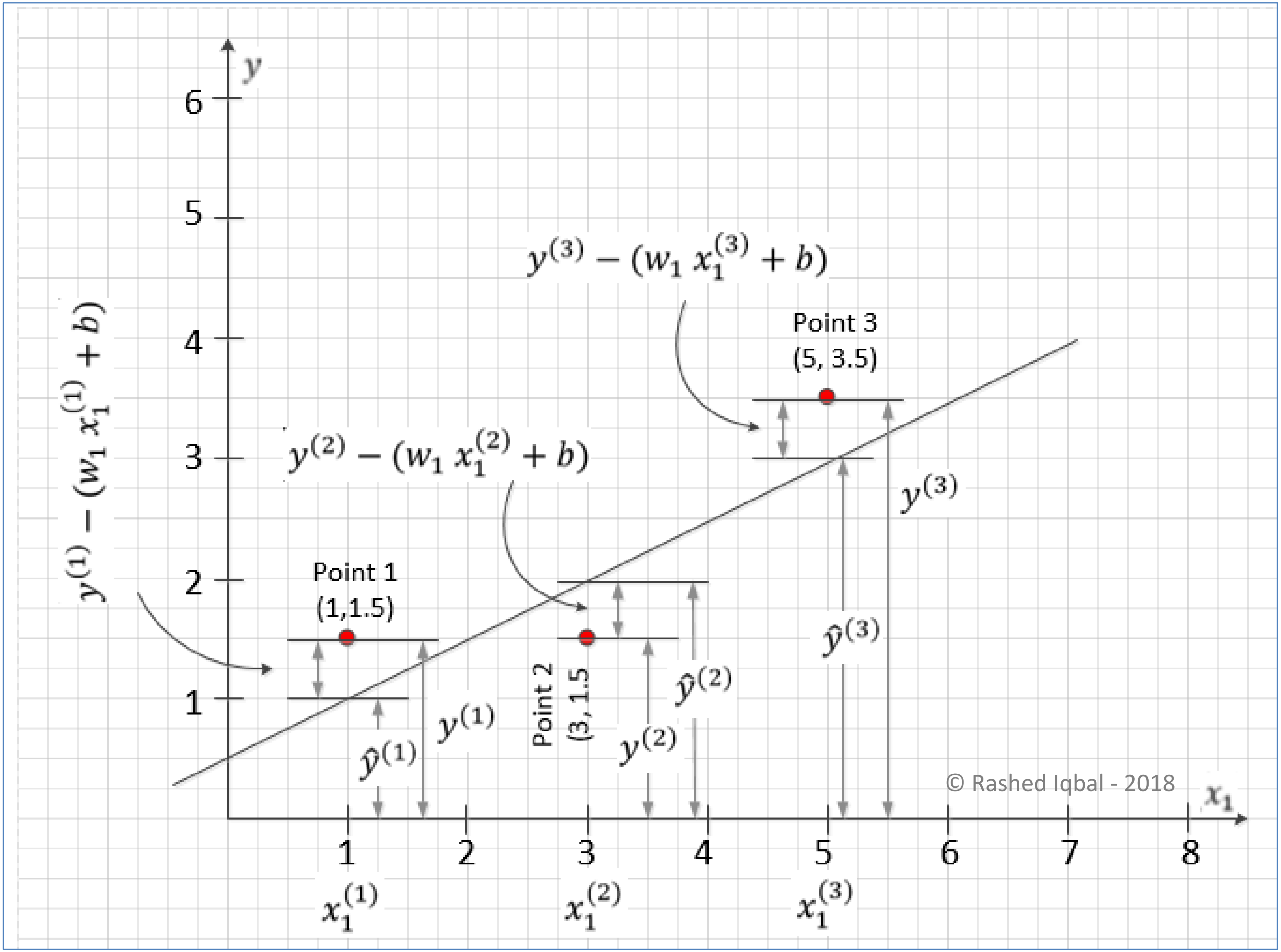
$$y^{(1)} - \hat{y}^{(1)} = y^{(1)} - (w_1\, x_1^{(1)} + b)$$

The difference distance between the Points 2 and the line:

$$y^{(2)} - \hat{y}^{(2)} = y^{(2)} - (w_1\, x_1^{(2)} + b)$$

The difference distance between the Points 3 and the line:

$$y^{(3)} - \hat{y}^{(3)} = y^{(3)} - (w_1\, x_1^{(3)} + b)$$



y
6
5
4
3
2
1
1
2
3
4
5
6
7
8
x_1
$x_1^{(1)}$
$x_1^{(2)}$
$x_1^{(3)}$
$y^{(3)} - (w_1\, x_1^{(3)} + b)$
$y^{(2)} - (w_1\, x_1^{(2)} + b)$
$y^{(1)} - (w_1\, x_1^{(1)} + b)$
Point 3
(5, 3.5)
Point 1
(1,1.5)
Point 2
(3, 1.5)
$y^{(1)}$
$\hat{y}^{(1)}$
$y^{(2)}$
$\hat{y}^{(2)}$
$y^{(3)}$
$\hat{y}^{(3)}$
© Rashed Iqbal - 2018

As these distances will be added later, we will take the square to remove the sign to produce what we call the Loss Function. The Loss Function for the training point 1 is written as:

$$L_1(b, w_1) = \left(y^{(1)} - (w_1\, x_1^{(1)} + b)\right)^2$$

Depending upon values of b and w_1 (or in other words upon inclination and height of the line), this Loss Function will go up or down. We are interested to find b and w_1 which will minimize this loss. The loss functions can be written for the two other training points as:

$$L_2(b, w_1) = \left(y^{(2)} - (w_1\, x_1^{(2)} + b)\right)^2$$

$$L_3(b, w_1) = \left(y^{(3)} - (w_1\, x_1^{(3)} + b)\right)^2$$

As our training examples do not all lie on a line, we will need to find b and w_1 that will minimize sum of these loss functions.

Let us sum these loss functions for each of the training points. The result is the Cost Function $J(b, w_1)$:

$$J(b, w_1) = L_1(b, w_1) + L_2(b, w_1) + L_3(b, w_1)$$

$$= \left(y^{(1)} - (w_1\, x_1^{(1)} + b)\right)^2 + \left(y^{(2)} - (w_1\, x_1^{(2)} + b)\right)^2$$

$$+ \left(y^{(3)} - (w_1\, x_1^{(3)} + b)\right)^2$$

Or

$$J(b, w_1) = \sum_{i=1}^{3} \left(y^{(i)} - (b + w_1\, x_1^{(i)})\right)^2$$

The Linear Regression problem in this simple case is to find out values of w_1 and b in such a way that the resulting line minimizes the cost function $J(b, w_1)$. Recall that the cost function is sum of squared vertical distances between the line and the points.

Let us put value of the three points (1, 1.5), (3, 1.5), and (5, 3.5) in our cost function:

$$J(b, w_1) = (1.5 - b - w_1)^2 + (1.5 - b - 3\, w_1)^2 + (3.5 - b - 5\, w_1)^2$$

Let us take partial derivative of J first with respect to b and then with respect to w_1:

$$\frac{\partial J(b, w_1)}{\partial b} = \frac{\partial}{\partial b}(1.5 - b - \ w_1)^2 + \frac{\partial}{\partial b}(1.5 - b - 3\, w_1)^2 + \frac{\partial}{\partial b}(3.5 - b \ - 5\, w_1)^2$$

$$= 2(1.5 - b - \ w_1)\frac{\partial}{\partial b}(1.5 - b - \ w_1) + 2(1.5 - b - \ 3\, w_1)\ \frac{\partial}{\partial b}(1.5 - b \ - 3\, w_1)$$

$$+ 2(3.5 - b - 5\, w_1)\frac{\partial}{\partial b}(3.5 - b - 5\, w_1)$$

$$= 2(1.5 - b - \ w_1)(-1) + 2(1.5 - b - \ 3\, w_1)\ (-1) + 2(3.5 - b - 5\, w_1)(-1)$$

$$= -2(1.5 - b - \ w_1 + 1.5 - b - \ 3\, w_1 + 3.5 - b - 5\, w_1)$$

Or

$$\frac{\partial J(b, w_1,)}{\partial b} = -2(6.5 - 3b - 9\, w_1) \qquad [\text{Equation } \mathbf{dJdb}]$$

This partial derivative should be zero at the minimum. Therefore:

$$6.5 - 3b - 9\, w_1 = 0$$

$$3b + 9\, w_1 = 6.5 \quad \text{[Equation 1]}$$

Now taking partial derivative with respect to w_1:

$$\frac{\partial J(w_1,b)}{\partial w_1} = \frac{\partial}{\partial w_1}(1.5 - b - \ w_1)^2 + \frac{\partial}{\partial w_1}(1.5 - b - 3\,w_1)^2 + \frac{\partial}{\partial w_1}(3.5 - b - 5\,w_1)^2$$

$$= 2(1.5 - b - \ w_1)\frac{\partial}{\partial w_1}(1.5 - b - \ w_1) + 2(1.5 - b - \ 3\,w_1)\ \frac{\partial}{\partial w_1}(1.5 - b - 3\,w_1)$$

$$+ 2(3.5 - b - 5\,w_1)\frac{\partial}{\partial w_1}(3.5 - b - 5\,w_1)$$

$$= 2(1.5 - b - \ w_1)(-1) + 2(1.5 - b - \ 3\,w_1)\ (-3) + 2(3.5 - b - 5\,w_1)(-5)$$

$$= -2(1.5 - b - \ w_1 + 4.5 - 3b - \ 9\,w_1 + 17.5 - 5\,b + 25\,w_1)$$

Or

$$\frac{\partial J(b,w_1)}{\partial w_1} = -2(23.5 - 9\,b - 35\,w_1) \qquad [\text{Equation } \mathbf{dJdw1}]$$

This partial derivative should be zero at the minimum. Therefore:

$$23.5 - 9b \ - 35\,w_1 = 0$$

$$9b + 35\,w_1 = 23.5 \quad [\text{Equation 2}]$$

The values of b and w_1 that minimizes $J(b, w_1)$ can be obtained by solving the following simultaneous equations:

$$3b + 9\, w_1 = 6.5 \qquad \text{[Equation 1]}$$

$$9b + 35\, w_1 = 23.5 \qquad \text{[Equation 2]}$$

From the Equation 1:

$$9b + 27\, w_1 = 19.5 \text{ Or}$$

$$9b = 19.5 - 27\, w_1$$

Put this value in the Equation 2:

$$19.5 - 27\, w_1 + 35\, w_1 = 23.5$$

$$(35 - 27)\ w_1 = 23.5 - 19.5$$

$$8\, w_1 = 4$$

$$w_1 = \frac{1}{2} = 0.5$$

The Equation 1 is:

$$3b + 9\, w_1 = 6.5$$

Putting the value of w_1 yields:

$$3b + 9\,(0.5) = 6.5$$

$$3b + 4.5 = 6.5$$

$$3b = 2 \text{ or}$$

$$b = \frac{2}{3} = 0.66667$$

Thus the point (b = 0.66667, w_1= 0.5) will minimize:

$$J(b, w_1) = \sum_{i=1}^{3} \left(y^{(i)} - (b + \ w_1\, x_1^{(i)}) \right)^2$$

In other words, 0.66667 and 0.5 are values of b and w_1 where $J(b, w_1)$ is minimum.

However, we cannot be sure if *J*(0.66667, 0.5) indeed is a minimum as it could be a maximum or a saddle point (or point of inflection). So we call it an optimum until we are sure it indeed is a minimum that require additional testing described below.

We assume that $J(b, w_1)$ is twice differentiable real function for which second partial derivatives exist. In order to test we define Hessian matrix H of the loss function $J(b, w_1)$ which consists of double partial derivatives:

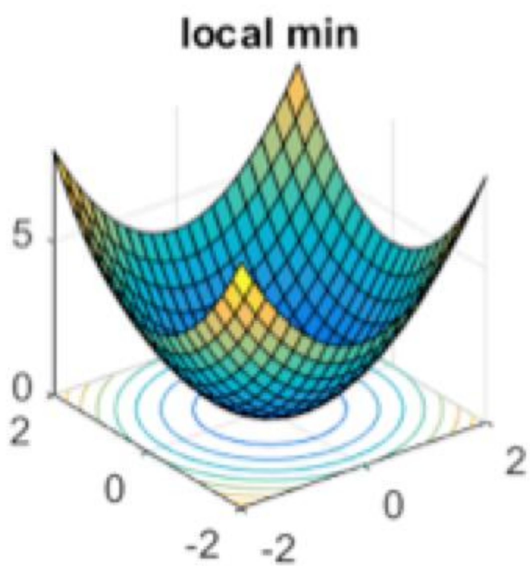
$$H(b, w_1) = \begin{bmatrix} \dfrac{\partial^2 J(b, w_1)}{\partial b^2} & \dfrac{\partial^2 J(b, w_1)}{\partial b \, \partial w_1} \\ \dfrac{\partial^2 J(b, w_1)}{\partial w_1 \, \partial b} & \dfrac{\partial^2 J(b, w_1)}{\partial w_1^2} \end{bmatrix}_{at\ optimum}$$

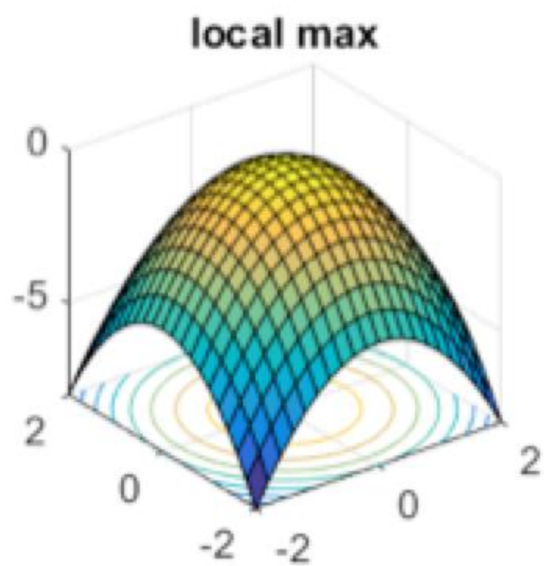
The determinant of the Hessian matrix is (Schwarz's Theorem: partial derivatives are commutative):

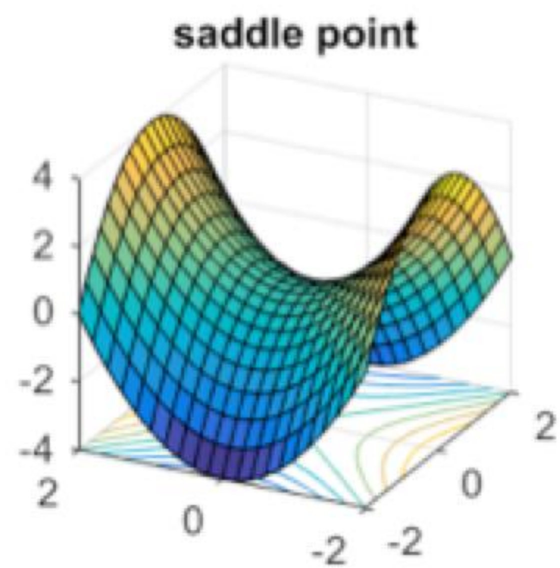
$$D(b, w_1) = \det(H(b, w_1)) = \frac{\partial^2 J(b, w_1)}{\partial b^2} \frac{\partial^2 J(b, w_1)}{\partial w_1^2} - \left(\frac{\partial^2 J(b, w_1)}{\partial b \, \partial w_1}\right)^2$$

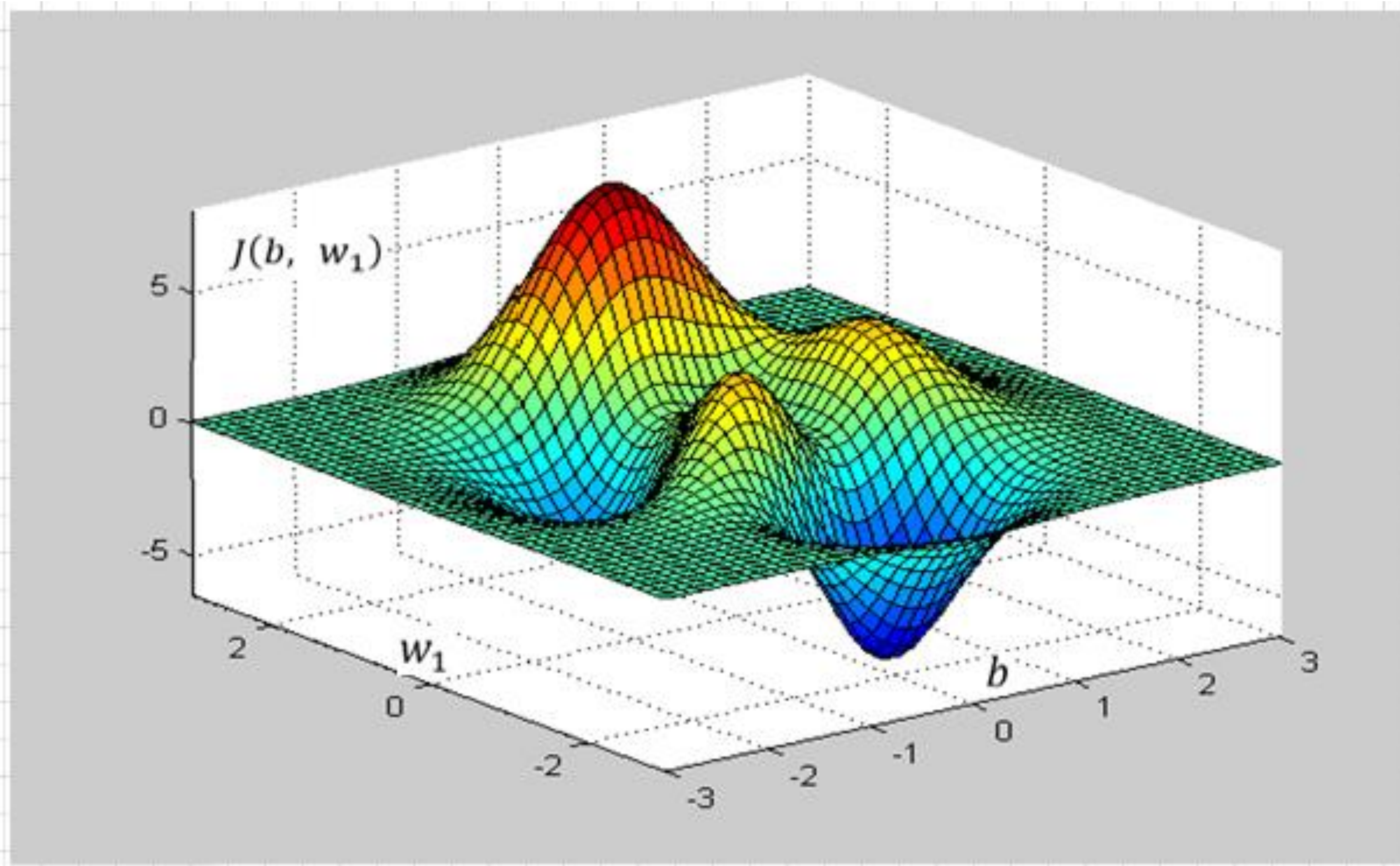
The determinant of the Hessian matrix $D(b, w_1)$ and the second derivative of $J(b, w_1)$ with respect of b can be used to determine if the optimum developed from the first derivative indeed is the minimum using the following table:

$D(b,w_1)$ **at the optimum point**	$\frac{\partial^2 J(b,w_1)}{\partial b^2}$ **at the optimum point**	**Conclusion**
> 0	> 0	Local minimum
> 0	< 0	Local maximum
< 0		Saddle point
$= 0$		The second derivative test is not conclusive, that is, the point could be a minimum, maximum, or a saddle point.









$J(b,\ w_1)$
5
0
-5
2
w_1
0
-2
-3
-2
-1
0
b
1
2
3

Let us now determine elements of the Hessian matrix for our cost function $J(b, w_1)$. Recall first partial derivate of $J(b, w_1)$ with respect to b from Equation **dJdb**:

$$\frac{\partial J(b, w_1)}{\partial b} = -2(6.5 - 3\,b - 9\,w_1)$$

Taking the second partial derivative with respect to b yields H_{11}:

$$H_{11} = \left|\frac{\partial^2 J(b, w_1)}{\partial^2 b^2}\right|_{b=\frac{2}{3}, w_1=\frac{1}{2}} = \left|-2\,\frac{\partial}{\partial b}(6.5 - 3\,b - 9\,w_1)\right|_{b=\frac{2}{3}, w_1=\frac{1}{2}} = 6$$

Taking the second partial derivative with respect to w_1 gives us H_{12}:

$$H_{12} = \left|\frac{\partial^2 J(b, w_1)}{\partial b\ \partial w_1}\right|_{b=\frac{2}{3}, w_1=\frac{1}{2}} = \left|-2\,\frac{\partial}{\partial w_1}(6.5 - 3\,b - 9\,w_1)\right|_{b=\frac{2}{3}, w_1=\frac{1}{2}} = 18$$

Next recall first partial derivative of $L(b, w_1)$ with respect to w_1 from Equation **dJdw1**:

$$\frac{\partial J(b, w_1)}{\partial w_1} = -2(23.5 - 9\,b - 35\,w_1)$$

Taking the second partial derivative with respect to w_1 yields H_{22}:

$$H_{22} = \left|\frac{\partial^2 J(b, w_1)}{\partial w_1^2}\right|_{b=\frac{2}{3}, w_1=\frac{1}{2}} = \left|-2\,\frac{\partial}{\partial w_1}(23.5 - 9\,b - 35\,w_1)\right|_{b=\frac{2}{3}, w_1=\frac{1}{2}} = 70$$

Although from Schwarz's Theorem we know that partial derivatives are commutative,

taking the second partial derivative with respect to b gives H_{21} which is same as H_{12}.

$$H_{21} = \left|\frac{\partial^2 J(b, w_1)}{\partial w_1\,\partial b}\right|_{b=\frac{2}{3}, w_1=\frac{1}{2}} = \left|-2\,\frac{\partial}{\partial b}(23.5 - 9\,b - 35\,w_1)\right|_{b=\frac{2}{3}, w_1=\frac{1}{2}} = 18$$

See Schwarz’s Theorm: https://en.wikipedia.org/wiki/Symmetry_of_second_derivatives

So the Hessian matrix can be written as:

$$H(b, w_1) = \begin{bmatrix} 6 & 18 \\ 18 & 70 \end{bmatrix}$$

Thus the determinant $D(b, w_1)$ of the Hessian matrix will be:

$$D(b, w_1) = 6 \text{ x } 70 - 18 \text{ x } 18 = 96 > 0$$

We also note that:

$$\text{dJdb2} = \frac{\partial^2 J(b, w_1)}{\partial^2 b^2} = 6 > 0$$

As both dJdb2 and $D(b, w_1)$ are greater than zero, $(\frac{2}{3}, \frac{1}{2})$ is the minimum.

Let us rewrite the cost function:

$$J(b, w_1) = \sum_{i=1}^{3} \left(y^{(i)} - (w_1\, x_1^{(i)} + b)\right)^2$$

This cost function is for one feature ($\mathbb{R}^2$ space) for three training points. Let us rewrite for *m* training examples:

$$J(b, w_1) = \sum_{i=1}^{m} \left(y^{(i)} - (w_1\, x_1^{(i)} + b)\right)^2$$

Extending to n features (or $\mathbb{R}^{n+1}$ space) yields:

$$J(w_1, \dots, w_n, b) = J(\mathrm{W}, b) = \sum_{i=1}^{m} \left(y^{(i)} - \left(\sum_{j=1}^{n} \left(w_j\, x_j^{(i)} \right) + b \right) \right)^2$$

Here W represents parameters $w_1, \ldots, w_n$. The Linear Regression problem for *n* features and *m* training examples now can be written as to finding $w_1, \ldots, w_n$ and b that will minimize the cost function:

$$\mathrm{W}, b \text{ at minimum} = \underset{\mathrm{W},b}{\operatorname{argmin}} J(\mathrm{W}, \mathrm{b}) = \underset{\mathrm{W},\, b}{\operatorname{argmin}} \sum_{i=1}^{m} \left(y^{(i)} - \left(\sum_{j=1}^{n} \left(w_j\, x_j^{(i)} \right) + b \right) \right)^2$$

In programming, a summation is typically implemented in loops. For example, the Cost Function below has two loops one that determines weighted sum of input features for each training example and the second that sums the square of distance:

$$J(w_1, \dots, w_n, b) = \sum_{i=1}^{m} \left(y^{(i)} - \left(\sum_{j=1}^{n} \left(w_j \, x_j^{(i)} \right) + b \right) \right)^2$$

This equation may be written in matrix form avoiding computationally expensive loops:

$$J(\mathrm{W}, b) = \ [(WX + B) - Y] \ \times [(WX + B) - Y]^T$$

where:

$$Y = \ [y^{(1)} \quad y^{(2)} \quad \dots \quad y^{(m)}]_{1 \times m} \qquad\qquad \mathrm{B} = [b \quad b \quad \dots \quad b]_{1 \times m}$$

$$W = \ [w_1 \quad w_2 \quad \dots \quad w_n]_{1 \times n}$$

$$X = [X^{(1)} \quad X^{(2)} \quad \dots \quad X^{(m)}]_{n \times m} = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(m)} \\ x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(m)} \\ \vdots & \vdots & \vdots & \vdots \\ x_n^{(1)} & x_n^{(2)} & \dots & x_n^{(m)} \end{bmatrix}_{n \times m}$$

Let us develop derivatives of the Cost Function with respect to $w_1, \dots, w_n$ and b. First write the Loss Function for the *i*th example:

$$L^{(i)}(w_1, \dots, w_n, b) = \left(y^{(i)} - \left(\sum_{j=1}^{n} \left(w_j \, x_j^{(i)} \right) + b \right) \right)^2$$

Take derivate with respect to b:

$$\frac{\partial L^{(i)}(w_1, \dots, w_n, b)}{\partial b} = 2 \left(y^{(i)} - \left(\sum_{j=1}^{n} \left(w_j \, x_j^{(i)} \right) + b \right) \right) \frac{\partial}{\partial b} \left(y^{(i)} - \left(\sum_{j=1}^{n} \left(w_j \, x_j^{(i)} \right) + b \right) \right)$$

$$\frac{\partial L^{(i)}(w_1, \dots, w_n, b)}{\partial b} = -2 \left(y^{(i)} - \left(\sum_{j=1}^{n} \left(w_j \, x_j^{(i)} \right) + b \right) \right)$$

Now taking derivative with respect to w_k:

$$\frac{\partial L^{(i)}(w_1, \dots, w_n, b)}{w_k} = 2\left(y^{(i)} - \left(\sum_{j=1}^{n}\left(w_j\, x_j^{(i)}\right) + b\right)\right)\frac{\partial}{\partial w_k}\left(y^{(i)} - \left(\sum_{j=1}^{n}\left(w_j\, x_j^{(i)}\right) + b\right)\right)$$

$$\frac{\partial L^{(i)}(w_1, \dots, w_n, b)}{\partial w_k} = 2\left(y^{(i)} - \left(\sum_{j=1}^{n}\left(w_j\, x_j^{(i)}\right) + b\right)\right)\left(-x_k^{(i)}\right)$$

$$= -2x_k^{(i)}\left(y^{(i)} - \left(\sum_{j=1}^{n}\left(w_j\, x_j^{(i)}\right) + b\right)\right)$$

As the Cost Function is sum of Loss Functions for the *m* training examples, we can determine derivative of the Cost Function by summing up derivatives of the Loss Functions.

$$\frac{\partial J(w_1, \dots, w_n, b)}{\partial b} = -2 \sum_{i=1}^{m} \left(y^{(i)} - \left(\sum_{j=1}^{n} \left(w_j \, x_j^{(i)} \right) + b \right) \right)$$

$$= -2 \, * \left(\mathrm{Y} - (\mathrm{WX} + \mathrm{B}) \right) \mathbf{1}$$

Here **1** is a vector of size *m* x 1 with all ones.

$$\frac{\partial J(w_1, \dots, w_n, b)}{w_k} = -2 \sum_{i=1}^{m} \left(x_k^{(i)} \left(y^{(i)} - \left(\sum_{j=1}^{n} \left(w_j \, x_j^{(i)} \right) + b \right) \right) \right)$$

$$= -2 \, * \left(\mathrm{Y} - (\mathrm{WX} + \mathrm{B}) \right) \mathrm{X}_k^T$$

Here X_k is the *k*th horizontal slice of X matrix.

Let us rewrite important formulas of Linear Regression in matrix form:

$$J(\mathrm{W}, b) = [(WX + B) - Y] \times [(WX + B) - Y]^T$$

where:

$$Y = [y^{(1)} \quad y^{(2)} \quad \dots \quad y^{(m)}]_{1 \times m} \qquad \mathrm{B} = [b \quad b \quad \dots \quad b]_{1 \times m}$$

$$W = [w_1 \quad w_2 \quad \dots \quad w_n]_{1 \times n}$$

$$X = [X^{(1)} \quad X^{(2)} \quad \dots \quad X^{(m)}]_{n \times m} = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(m)} \\ x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(m)} \\ \vdots & \vdots & \vdots & \vdots \\ x_n^{(1)} & x_n^{(2)} & \dots & x_n^{(m)} \end{bmatrix}_{n \times m}$$

Note that here in matrix X, the training examples are arranged horizontally. In other words, each column represents one training example. The derivatives are:

$$\frac{\partial J(w_1, \dots, w_n, b)}{\partial b} = -2 * \left(\mathrm{Y} - (\mathrm{WX} + \mathrm{B})\right) \mathbf{1}$$

$$\frac{\partial J(w_1, \dots, w_n, b)}{w_k} = -2 * \left(\mathrm{Y} - (\mathrm{WX} + \mathrm{B})\right) \mathrm{X}_k^T$$

Here **1** is a vector of size *m* x 1 with all ones and X_k as the *k*th horizontal slice of X matrix.

With derivatives the following update formulas could be written:

$$b = b - \alpha \frac{\partial J(W, b)}{\partial b}$$

$$w_k = w_k - \alpha \frac{\partial J(W, b)}{\partial w_k} \text{ for } k = 1, 2, .., n$$

Where:

$$\frac{\partial J(w_1, \ldots, w_n, b)}{\partial b} = -2 \sum_{i=1}^{m} \left(y^{(i)} - \left(\sum_{j=1}^{n} \left(w_j \, x_j^{(i)} \right) + b \right) \right)$$

$$\frac{\partial J(w_1, \ldots, w_n, b)}{w_k} = -2 \sum_{i=1}^{m} \left(x_k^{(i)} \left(y^{(i)} - \left(\sum_{j=1}^{n} \left(w_j \, x_j^{(i)} \right) + b \right) \right) \right)$$

Recap

- Reviewed a simple Linear Regression problem and derived necessary equations.
- Defined the terms Loss Function and Cost Function.
- Wrote the Linear Regression problem and Gradient Descent update formulas a in general form
- Wrote the Linear Regression problem in matrix notation including the derivatives (for update formulas)

Converting Summation Representation of Derivatives to Matrix Representation:

Let us first take derivative of the Loss Function for the *i*th training example. Here $X^{(i)}$ the *i*th vertical slice of X. This is multiplied with W to which b is added before subtracting it from $y^{(i)}$. This is then multiplied with $x_k^{(i)}$.

$$\frac{\partial L^{(i)}(w_1, \dots, w_n, b)}{w_k} = -2x_k^{(i)} \left(y^{(i)} - \left(\sum_{j=1}^{n} \left(w_j \; x_j^{(i)} \right) + b \right) \right)$$

$$= -2x_k^{(i)} \left(y^{(i)} - \left(WX^{(i)} + b \right) \right) = \; -2 \, p^{(i)} \, q^{(i)}$$

Here

$$p^{(i)} = x_k^{(i)}$$

And

$$q^{(i)} = \; y^{(i)} - \left(WX^{(i)} + b \right)$$

We now can sum for *m* training examples:

$$\frac{\partial J(w_1, \dots, w_n, b)}{w_k} = -2 \sum_{i=1}^{m} \left(p^{(i)} \, q^{(i)} \right)$$

$$= -2 \, P^T \, Q$$

$$= -2 \; * \; \mathrm{X}_k^T \, (\mathrm{Y} - (\mathrm{WX} + \mathrm{B}))$$

Logistic Regression Solution:

Loop: iteration $j = 1$ to h:

$J = 0$

$\mathrm{dJdb} = 0$

Loop: $k = 1$ to n

$\mathrm{dJd}w_k = 0$

Loop: $i = 1$ to m

$\hat{y}^{(i)} = 0$

Loop: $k = 1$ to n

$\hat{y}^{(i)} = \hat{y}^{(i)} + w_k\, x_k^{(i)}$

$\hat{y}^{(i)} = \hat{y}^{(i)} + b$

$\hat{y}^{(i)} = \sigma(\hat{y}^{(i)})$

$J = J + \left(y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log\left(1 - \hat{y}^{(i)}\right)\right)$

$\mathrm{dY}^{(i)} = \hat{y}^{(i)} - y^{(i)}$

$\mathrm{dJdb} = \mathrm{dJdb} + \mathrm{dY}^{(i)}$

Loop: $k = 1$ to n

$\mathrm{dJd}w_k = \mathrm{dJd}w_k + \mathrm{dJ}\, x_k^{(i)}$

$\mathrm{dJdb} = \frac{\mathrm{dJdb}}{m}$

Loop: $k = 1$ to n

$\mathrm{dJd}w_k = \frac{\mathrm{dJd}w_k}{m}$

$\mathrm{J}^{(g)} = -\frac{J}{m}$

$b = b - \alpha\, \mathrm{dJdb}$

Loop: $k = 1$ to n

$w_k = w_k - \alpha\, \mathrm{dJd}w_k$

$$\frac{\partial J(w_1, \dots, w_n, b)}{\partial b}$$

$$= -2 \sum_{i=1}^{m} \left(y^{(i)} - \left(\sum_{j=1}^{n} \left(w_j\, x_j^{(i)} \right) + b \right) \right)$$

$$\frac{\partial J(w_1, \dots, w_n, b)}{w_k}$$

$$= -2 \sum_{i=1}^{m} \left(x_k^{(i)} \left(y^{(i)} - \left(\sum_{j=1}^{n} \left(w_j\, x_j^{(i)} \right) + b \right) \right) \right)$$

© Rashed Iqbal - 2018

$$J(w_1, \dots, w_n, b)$$

$$= \sum_{i=1}^{m} \left(y^{(i)} - \left(\sum_{j=1}^{n} \left(w_j\, x_j^{(i)} \right) + b \right) \right)^2$$

© Rashed Iqbal - 2018