Machine Learning

Linear Regression

Dr. Pratyay Kuila

Dept. of Computer Science & Engineering
NIT Sikkim, Ravangla-737139

Regression

- Regression analysis is a statistical technique that is mainly used to estimate the relationship among the variables.
- Let D denotes a dataset containing m observations, $D = \{(x_i, y_i) | i = 1, 2, \ldots, m\}$. Each x_i corresponds to the set of attributes of the i^{th} observation. These are called *explanatory variables* and can be discrete or continuous. y_i corresponds to the *target variable*.
- **Regression** is the task of learning a target function f that maps each attribute set x into a continuous-valued output y.
- When the target variable that we are trying to predict is continuous, we call the learning problem a *regression problem*. When y can take on only a small number of discrete values, we call it a *classification problem*.

Regression: Loss Functions

Sum of Squared Error (SSE): A loss function measures how accurate the outputs of the model are in comparison to the real outputs (target variables).

- For regression tasks the *Sum of Squared Error* (*SSE*) is a common choice for the loss function.

- Let us assume a model with *m* outputs. The *SSE* is calculated as Follows:

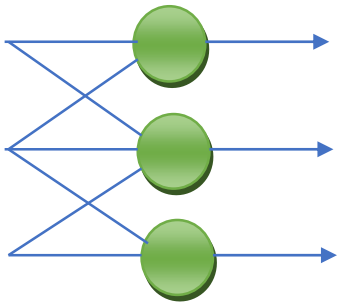
$$SSE = \sum_{i=1}^{m} (y_i - t_i)^2$$

- Where, y_i be the actual output and t_i be the target/predicted output.

Regression: Loss Functions

Sum of Squared Error (SSE):

Example: A model network with *m*= 3 outputs.



Actual Outputs	Target/Predicted Values
0.7	0.8
0.9	0.8
1.2	1.1

In this case, the *SSE* is calculated as follows:

$$SSE = \{(y_1 - t_1)^2 + (y_2 - t_2)^2 + (y_3 - t_3)^2\}$$

$$= \{(0.7 - 0.8)^2 + (0.9 - 0.8)^2 + (1.2 - 1.1)^2\} = 0.03$$

Regression: Loss Functions

Mean Squared Error (MSE):

- For regression tasks the *Mean Squared Error* (*MSE*) is a common choice for the loss function.

- Let us assume a model with *m* outputs. The MSE is calculated as Follows:

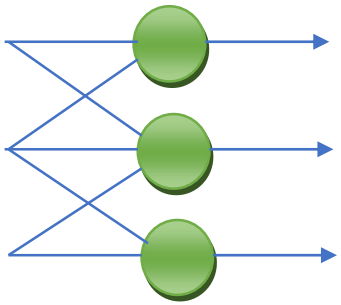
$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - t_i)^2$$

➢ Where, y_i be the actual output and t_i be the target/predicted output.

Regression: Loss Functions

Mean Squared Error (MSE):

Example: A model network with $m = 3$ outputs.



Actual Outputs	Target/Predicted Values
0.7	0.8
0.9	0.8
1.2	1.1

In this case, the MSE is calculated as follows:

$$MSE = \frac{1}{3}\{(y_1 - t_1)^2 + (y_2 - t_2)^2 + (y_3 - t_3)^2\}$$
$$= \frac{1}{3}\{(0.7 - 0.8)^2 + (0.9 - 0.8)^2 + (1.2 - 1.1)^2\}$$
$$=0.01$$

Regression: Loss Functions

Root Mean Squared Error (RMSE):

- For *Root Mean Squared Error* (*RMSE*) is a common choice for regression tasks. It is calculated as the root of the MSE.

- Let us assume a model with *m* outputs. The RMSE is calculated as Follows:

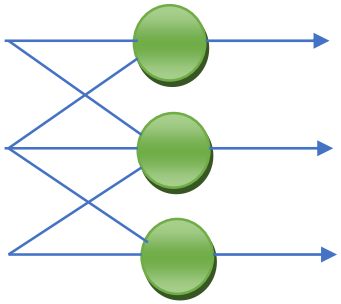
$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - t_i)^2}$$

- Where, y_i be the actual output and t_i be the target/predicted output.

Regression: Loss Functions

Root Mean Squared Error (RMSE):

Example: A model network with $m = 3$ outputs.



Actual Outputs	Target/Predicted Values
0.7	0.8
0.9	0.8
1.2	1.1

In this case, the MSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{3}\{(y_1 - t_1)^2 + (y_2 - t_2)^2 + (y_3 - t_3)^2\}}$$

$$= \sqrt{\frac{1}{3}\{(0.7 - 0.8)^2 + (0.9 - 0.8)^2 + (1.2 - 1.1)^2\}}$$

$$=0.1$$

Regression: Loss Functions

Mean Absolute Error (MAE):

- For regression tasks the *Mean Absolute Error* (*MAE*) is another common choice for the loss function.

- Let us assume a model with *m* outputs. The *MAE* is calculated as Follows:

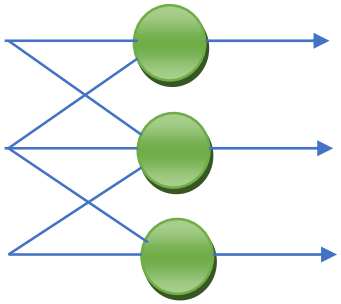
$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - t_i|$$

- Where, y_i be the actual output and t_i be the target/predicted output.

Regression: Loss Functions

Mean Absolute Error (MAE):

Example: A model network with $m = 3$ outputs.



Actual Outputs	Target/Predicted Values
0.7	0.8
0.9	0.8
1.2	1.1

In this case, the MAE is calculated as follows:

$$MAE = \frac{1}{3}\{|y_1 - t_1| + |y_2 - t_2| + |y_3 - t_3|\}$$

$$= \tfrac{1}{3}\{|0.7 - 0.8| + |0.9 - 0.8| + |1.2 - 1.1|\} = 0.1$$

Regression: Loss Functions

Mean Squared Logarithmic Error (MSLE):

- For regression task the *Mean Squared Logarithmic Error* (*MSLE*) is another common choice for the loss function.

- The *MSLE* is the particularly suitable when the difference between output and target value can be high. In such a case, the MSE would be very high, which could lead to problems when training the network.
- Let us assume a model with *m* outputs. The *MSLE* is calculated as Follows:

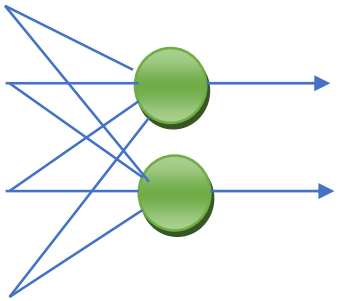
$$MAE = \frac{1}{m}\sum_{i=1}^{m}\{\log(y_i + 1) - \log(t_i + 1)\}^2$$

- Where, y_i be the actual output and t_i be the target/predicted output.

Regression: Loss Functions

Mean Squared Logarithmic Error (MSLE):

Example: A model network with $m = 2$ outputs.



Actual Outputs	Target/Predicted Values
0.7	0.8
0.9	0.8

In this case, the MSLE is calculated as follows:

$$MSLE = \frac{1}{2}\{\log(y_1+1)-\log(t_1+1)\}^2 + \{\log(y_2+1)-\log(t_2+1)\}^2$$

$$= \frac{1}{2}\{\log(0.7+1)-\log(0.8+1)\}^2 + \{\log(0.9+1)-\log(0.8+1)\}^2$$

$$= \frac{1}{2}\{\log(1.7)-\log(1.8)\}^2 + \{\log(1.9)-\log(1.8)\}^2$$

$$= 0.000058$$

Regression: Loss Functions

***Cross-Entropy Loss or Log Loss*:**

- Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1.
- Cross-entropy loss increases as the predicted probability diverges from the actual label. It can be measured as follows:

$$\text{Cross-entropy Loss} = -\sum_{i=1}^{m} y_i \log t_i$$

- Where, y_i be the actual output and t_i be the target/predicted output.
- For binary classification model:

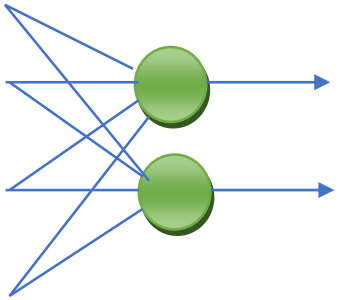
$$\text{Cross-entropy Loss} = -y_i \log t_i - (1 - y_i) \log (1 - t_i)$$

- Use cross-entropy loss if probabilities are involved.

Regression: Loss Functions

Cross-Entropy Loss or Log Loss:

Example: For a binary classification model (*m*= 2 outputs).



Actual Outputs	Target/Predicted Values
0.7	0.8
0.3	0.2

In this case, the Cross-entropy is calculated as follows:

$$\begin{aligned}\text{Cross-entropy Loss} &= -0.7\log 0.8 - (1-0.7)\log (1-0.8)\\ &= -0.7\log 0.8 - 0.3\log 0.2\\ &= 0.639\end{aligned}$$

- The loss (or error) is measured as a number between 0 and 1, with 0 being a perfect model.

Linear Regression: Least Square Approach

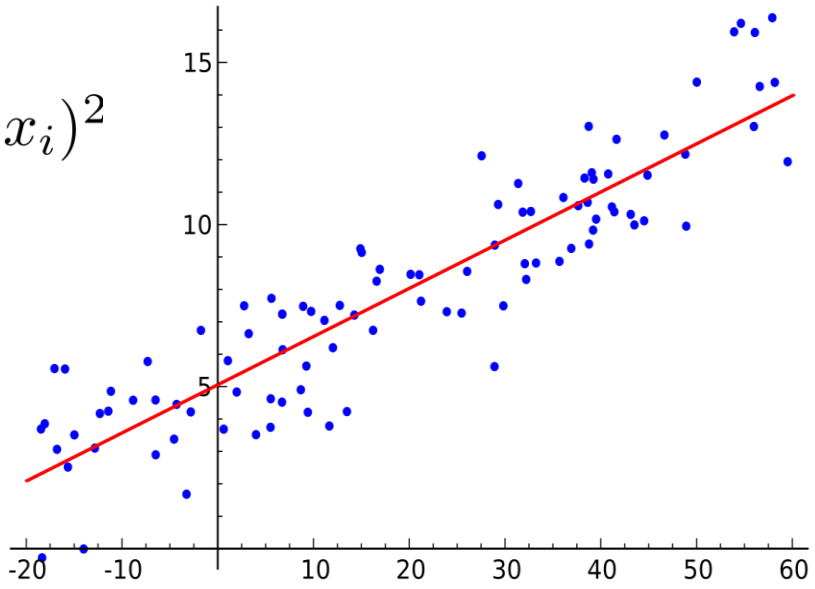
- The goal of linear regression is to find a target function that can minimize the error.
- The most common approach to estimating a regression equation is the least squares approach.
- This approach leads to a fitted line that minimizes the *sum of the squared errors*, i.e.,

$$\text{Minimize } SSE = \sum (y_i - f(x_i))^2$$

Or

$$\text{Minimize } SSE = \sum (y_i - b_1 - b_2 x_i)^2$$

- Given a set of points (x_i, y_i) on a scatter-plot find the best-fit line $f(x_i)=b_1+b_2x_i$, such that $SSE= \sum \{y_i - f(x_i)\}^2$ is minimized.
- We have to estimate the value of b_1 and b_2 so that it leads to minimum error. It can be estimated as follow:



Linear Regression: Least Square Approach

- The problem is the estimation of b_1 and b_2.
- It can be estimated as follow. First we apply partial derivatives on SSE by b_1.

$$\frac{\partial SSE}{\partial b_1} = -2\sum_{i=1}^{m}(y_i - b_1 - b_2 x_i) = 0$$

$$\text{or, } \sum_{i=1}^{m}(y_i - b_1 - b_2 x_i) = 0$$

$$\text{or, } mb_1 + b_2\sum_{i=1}^{m} x_i = \sum_{i=1}^{m} y_i \qquad \ldots. \ (a)$$

$$\text{or, } b_1 = \frac{1}{m}\sum_{i=1}^{m} y_i - b_2\frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\text{or, } b_1 = \bar{y} - b_2\bar{x}$$

$$b_1 = \bar{y} - b_2\bar{x}$$

Linear Regression: Least Square Approach

Now we apply derivative by b_2:

$$\frac{\partial SSE}{\partial b_2} = -2\sum_{i=1}^{m}(y_i - b_1 - b_2 x_i)x_i = 0$$

or, $\sum_{i=1}^{m} x_i(y_i - b_1 - b_2 x_i) = 0$

or, $b_1 \sum_{i=1}^{m} x_i + b_2 \sum_{i=1}^{m} x_i^2 = \sum_{i=1}^{m} x_i y_i$ .... *(b)*

Putting $b_1 = \frac{1}{m}\sum_{i=1}^{m} y_i - b_2 \frac{1}{m}\sum_{i=1}^{m} x_i$ // Refer to slide 16

$$\left(\frac{1}{m}\sum_{i=1}^{m} y_i - b_2 \frac{1}{m}\sum_{i=1}^{m} x_i\right)\sum_{i=1}^{m} x_i + b_2 \sum_{i=1}^{m} x_i^2 = \sum_{i=1}^{m} x_i y_i$$

Linear Regression: Least Square Approach

We have:

$$\left(\frac{1}{m}\sum_{i=1}^{m} y_i - b_2\frac{1}{m}\sum_{i=1}^{m} x_i\right)\sum_{i=1}^{m} x_i + b_2\sum_{i=1}^{m} x_i^2 = \sum_{i=1}^{m} x_i y_i$$

Or, $$-b_2\frac{1}{m}\left(\sum_{i=1}^{m} x_i\right)^2 + b_2\sum_{i=1}^{m} x_i^2 = \sum_{i=1}^{m} x_i y_i - \frac{1}{m}\sum_{i=1}^{m} y_i \sum_{i=1}^{m} x_i$$

Or, $$b_2 = \frac{\sum_{i=1}^{m} x_i y_i - \frac{1}{m}\sum_{i=1}^{m} y_i \sum_{i=1}^{m} x_i}{\left(\sum_{i=1}^{m} x_i^2 - \frac{1}{m}\left(\sum_{i=1}^{m} x_i\right)^2\right)}$$

Or, $$b_2 = \frac{\sum_{i=1}^{m} x_i y_i - m\bar{x}\bar{y}}{\sum_{i=1}^{m} x_i^2 - m(\bar{x})^2}$$

Or, $$b_2 = \frac{\overline{x_i y_i} - \bar{x}\bar{y}}{\overline{x_i^2} - (\bar{x})^2}$$

Linear Regression: Least Square Approach

Let us take, $S_{XY} = \sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})$

Therefore, $S_{XX} = \sum_{i=1}^{m}(x_i - \bar{x})^2$ and $S_{YY} = \sum_{i=1}^{m}(y_i - \bar{y})^2$

We have,

$$b_2 = \frac{\sum_{i=1}^{m} x_i y_i - \frac{1}{m}\sum_{i=1}^{m} y_i \sum_{i=1}^{m} x_i}{\left(\sum_{i=1}^{m} x_i^2 - \frac{1}{m}\left(\sum_{i=1}^{m} x_i\right)^2\right)} = \frac{\sum_{i=1}^{m} x_i y_i - m\bar{x}\bar{y}}{\sum_{i=1}^{m} x_i^2 - m(\bar{x})^2}$$

or, $b_2 = \frac{S_{XY}}{S_{XX}}$

$$b_1 = \bar{y} - b_2\bar{x}$$

Linear Regression: Least Square Approach

***Example*:** Consider the set of 5 points:

x	1	2	3	4	5
y	1.2	1.8	2.6	3.2	3.8

After analysis:

x_i	y_i	x_i^2	$x_i y_i$
1	1.2	1	1.2
2	1.8	4	3.6
3	2.6	9	7.8
4	3.2	16	12.8
5	3.8	25	19
$\sum x_i = 15$	$\sum y_i = 12.6$	$\sum x_i^2 = 55$	$\sum (x_i y_i) = 44.4$
$\bar{x_i} = 3$	$\bar{y_i} = 2.52$	$\bar{x_i^2} = 11$	$\overline{x_i y_i} = 8.88$

A general solution to the normal equation can be expressed as

$$b_2 = \frac{\sum_{i=1}^{m} x_i y_i - m\bar{x}\bar{y}}{\sum_{i=1}^{m} x_i^2 - m(\bar{x})^2} \text{ and } b_1 = \bar{y} - b_2\bar{x}$$

Linear Regression: Least Square Approach

***Example*:** Consider the set of 5 points:

x	1	2	3	4	5
y	1.2	1.8	2.6	3.2	3.8

After analysis:

x_i	y_i	x_i^2	$x_i y_i$
1	1.2	1	1.2
2	1.8	4	3.6
3	2.6	9	7.8
4	3.2	16	12.8
5	3.8	25	19
$\sum x_i = 15$	$\sum y_i = 12.6$	$\sum x_i^2 = 55$	$\sum (x_i y_i) = 44.4$
$\bar{x_i} = 3$	$\bar{y_i} = 2.52$	$\bar{x_i^2} = 11$	$\overline{x_i y_i} = 8.88$

A general solution to the normal equation can be expressed as

$$b_2 = \frac{44.4 - (5).(3).(2.52)}{55 - 5.(3)^2} \text{ and } b_1 = 2.52 - b_2.3$$

Linear Regression: Least Square Approach

***Example*:** Consider the set of 5 points:

x	1	2	3	4	5
y	1.2	1.8	2.6	3.2	3.8

After analysis:

x_i	y_i	x_i^2	$x_i y_i$
1	1.2	1	1.2
2	1.8	4	3.6
3	2.6	9	7.8
4	3.2	16	12.8
5	3.8	25	19
$\sum x_i = 15$	$\sum y_i = 12.6$	$\sum x_i^2 = 55$	$\sum (x_i y_i) = 44.4$
$\bar{x_i} = 3$	$\bar{y_i} = 2.52$	$\bar{x_i^2} = 11$	$\overline{x_i y_i} = 8.88$

A general solution to the normal equation can be expressed as

$$b_2 = 0.66 \text{ and } b_1 = 0.54$$

Therefore, $f(x_i) = 0.54 + 0.66 x_i$

Linear Regression: Least Square Approach

***Example*:** Consider the set of 10 points:

x	1	2	3	4	4	5	5	6	6	7
y	7	8	9	8	9	11	10	13	14	13

A general solution to the normal equation can be expressed as

$$b_1 = \bar{y} - b_2\bar{x} \text{ and } b_2 = \frac{S_{XY}}{S_{XX}} = \frac{\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{m}(x_i - \bar{x})^2}$$

A linear model that results in the minimum squared error is than

$$f(x_i) = b_1 + b_2 x_i$$

Here, $\bar{x} = \frac{1}{10}\sum x_i = 4.3$ and $\bar{y} = \frac{1}{10}\sum y_i = 10.2$

Hence, $S_{XY} = 37.4$ and $S_{XX} = 32.1$ // Check the calculations…

Therefore, $f(x_i) = b_1 + b_2 x_i = \bar{y} + \frac{S_{XY}}{S_{XX}}(x_i - \bar{x})$

$$= 10.2 + \frac{37.4}{32.1}(x_i - 4.3)$$

$$= 5.19 + 1.17x_i$$

Linear Regression: Least Square Approach

***Example*:** Consider the set of 10 points:

x	1	2	3	4	4	5	5	6	6	7
y	7	8	9	8	9	11	10	13	14	13

A general solution to the normal equation can be expressed as

$$b_1 = \bar{y} - b_2\bar{x} \text{ and } b_2 = \frac{\overline{x_i y_i} - \bar{x}\bar{y}}{\overline{x_i^2} - (\bar{x})^2}$$

A linear model that results in the minimum squared error is than

$$f(x_i) = b_1 + b_2 x_i$$

Here, $\bar{x} = \frac{1}{10}\sum x_i = 4.3$, $\bar{y} = \frac{1}{10}\sum y_i = 10.2$ and $\bar{x}\bar{y} = 43.86$

Here, $(\bar{x})^2 = 18.49$, $\overline{x_i^2} = 21.7$ and $\overline{x_i y_i} = 47.6$

Therefore, $b_2 = \frac{47.6 - 43.86}{21.7 - 18.49} = 1.1651 \approx 1.17$ // Check the calculations…

Therefore, $b_1 = 10.2 - (1.17)(4.3) = 5.19$

Linear Regression: Least Square Approach

Matrix representation: We have,

$$mb_1 + b_2 \sum_{i=1}^{m} x_i = \sum_{i=1}^{m} y_i \quad \text{.... } (a) \text{ //Slide no.16}$$

$$b_1 \sum_{i=1}^{m} x_i + b_2 \sum_{i=1}^{m} x_i^2 = \sum_{i=1}^{m} x_i y_i \quad \text{.... } (b) \text{ //Slide no.17}$$

Therefore,

$$\begin{bmatrix} m & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \end{bmatrix}$$

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} m & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum y_i \\ \sum x_i y_i \end{bmatrix}$$

Linear Regression: Least Square Approach

***Example*:** Consider the set of 10 points:

x	1	2	3	4	4	5	5	6	6	7
y	7	8	9	8	9	11	10	13	14	13

Therefore, $\sum_{i=1}^{m} x_i = 43$ and $\sum_{i=1}^{m} x_i^2 = 217$

$\sum_{i=1}^{m} y_i = 102$, $\sum_{i=1}^{m} y_i^2 = 1049$ and $\sum_{i=1}^{m} x_i y_i = 476$

We have,

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} m & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum y_i \\ \sum x_i y_i \end{bmatrix} = \begin{bmatrix} 10 & 43 \\ 43 & 217 \end{bmatrix}^{-1} \begin{bmatrix} 102 \\ 476 \end{bmatrix}$$

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 5.19 \\ 1.17 \end{bmatrix}$$

Hence, $f(x_i)=5.19+1.17x_i$

Multivariate Linear Regression

- What if we have multiple *x* components. It can be represented as follow:

$$
\begin{aligned}
y_1 &= b_0 + b_1 x_{11} + b_2 x_{12} + \ldots + b_n x_{1n} \\
y_2 &= b_0 + b_1 x_{21} + b_2 x_{22} + \ldots + b_n x_{2n} \\
&\vdots \qquad\qquad\qquad\qquad \vdots \\
y_m &= b_0 + b_1 x_{m1} + b_2 x_{m2} + \ldots + b_n x_{mn}
\end{aligned}
$$

- It can be represented in matrix as: $Y = XB$

Where,

$$
Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad X = \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1n} \\ 1 & x_{21} & x_{22} & \ldots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{m1} & x_{m2} & \ldots & x_{mn} \end{bmatrix} \quad B = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}
$$

Multivariate Linear Regression

- Let us introduce an error term as: $\in = \begin{bmatrix} \epsilon_0 & \epsilon_1 & \dots & \epsilon_n \end{bmatrix}^T$
- Now, it can be represented in matrix as: $Y = XB + \in$
- Error term can be expressed as:

$$SSE = \sum_{i=0}^{n} \epsilon_i^2 = \in^T \in = [Y - XB]^T [Y - XB]$$

- Applying partial derivatives,

$$\frac{\partial SSE}{\partial \hat{b}} = -2X^T Y + 2X^T X\hat{b} = 0$$

$$\text{or, } 2X^T X\hat{b} = 2X^T Y$$

$$\text{or, } \hat{b} = (X^T X)^{-1} X^T Y$$

Multivariate Linear Regression

Example:

$$\mathbf{X} = \begin{bmatrix} 1 & 2 & 50 \\ 1 & 8 & 110 \\ 1 & 11 & 120 \\ 1 & 10 & 550 \\ 1 & 8 & 295 \\ 1 & 4 & 200 \\ 1 & 2 & 375 \\ 1 & 2 & 52 \\ 1 & 9 & 100 \\ 1 & 8 & 300 \\ 1 & 4 & 412 \\ 1 & 11 & 400 \\ 1 & 12 & 500 \\ 1 & 2 & 360 \\ 1 & 4 & 205 \\ 1 & 4 & 400 \\ 1 & 20 & 600 \\ 1 & 1 & 585 \\ 1 & 10 & 540 \\ 1 & 15 & 250 \\ 1 & 15 & 290 \\ 1 & 16 & 510 \\ 1 & 17 & 590 \\ 1 & 6 & 100 \\ 1 & 5 & 400 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 9.95 \\ 24.45 \\ 31.75 \\ 35.00 \\ 25.02 \\ 16.86 \\ 14.38 \\ 9.60 \\ 24.35 \\ 27.50 \\ 17.08 \\ 37.00 \\ 41.95 \\ 11.66 \\ 21.65 \\ 17.89 \\ 69.00 \\ 10.30 \\ 34.93 \\ 46.59 \\ 44.88 \\ 54.12 \\ 56.63 \\ 22.13 \\ 21.15 \end{bmatrix}$$

The $\mathbf{X}'\mathbf{X}$ matrix is

$$\mathbf{X}'\mathbf{X} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 2 & 8 & \cdots & 5 \\ 50 & 110 & \cdots & 400 \end{bmatrix} \begin{bmatrix} 1 & 2 & 50 \\ 1 & 8 & 110 \\ \vdots & \vdots & \vdots \\ 1 & 5 & 400 \end{bmatrix}$$

$$= \begin{bmatrix} 25 & 206 & 8{,}294 \\ 206 & 2{,}396 & 77{,}177 \\ 8{,}294 & 77{,}177 & 3{,}531{,}848 \end{bmatrix}$$

and the $\mathbf{X}'\mathbf{y}$ vector is

$$\mathbf{X}'\mathbf{y} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 2 & 8 & \cdots & 5 \\ 50 & 110 & \cdots & 400 \end{bmatrix} \begin{bmatrix} 9.95 \\ 24.45 \\ \vdots \\ 21.15 \end{bmatrix} = \begin{bmatrix} 725.82 \\ 8{,}008.47 \\ 274{,}816.71 \end{bmatrix}$$

The least squares estimates are

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

Multivariate Linear Regression

Example

or

$$\begin{bmatrix}\hat{\beta}_0\\ \hat{\beta}_1\\ \hat{\beta}_2\end{bmatrix} = \begin{bmatrix}25 & 206 & 8,294\\ 206 & 2,396 & 77,177\\ 8,294 & 77,177 & 3,531,848\end{bmatrix}^{-1}\begin{bmatrix}725.82\\ 8,008.37\\ 274,811.31\end{bmatrix}$$

$$= \begin{bmatrix}0.214653 & -0.007491 & -0.000340\\ -0.007491 & 0.001671 & -0.000019\\ -0.000340 & -0.000019 & +0.0000015\end{bmatrix}\begin{bmatrix}725.82\\ 8,008.47\\ 274,811.31\end{bmatrix}$$

$$= \begin{bmatrix}2.26379143\\ 2.74426964\\ 0.01252781\end{bmatrix}$$

Therefore, the fitted regression model with the regression coefficients rounded to five decimal places is

$$\hat{y} = 2.26379 + 2.74427x_1 + 0.01253x_2$$

Linear Regression: LMS Algorithm

- Here, we use the error function as, $E = \frac{1}{2}\sum_{i=1}^{m}\{f(\vec{x}_i) - y_i\}^2$
- We want to choose b_j so as to minimize E. To do so, let us use a search algorithm that starts with some "initial guess" for b_j, and that repeatedly changes b_j to make E smaller, until hopefully we converge to a value of b_j that minimizes E. Specifically, let us consider the ***gradient descent*** algorithm, which starts with some initial b_j, and repeatedly performs the following update:

$$b_j = b_j - \eta \frac{\partial E}{\partial b_j}$$

- This update is simultaneously performed for all values of $j = 1, \ldots, n$. Here, η is called the *learning rate*. This is a very natural algorithm that repeatedly takes a step in the direction of steepest decrease of E.

Linear Regression: LMS Algorithm

In order to implement this algorithm, we have to work out what is the partial derivative term on the right hand side. Let us first work it out. We have,

$$\frac{\partial E}{\partial b_j} = \frac{\partial}{\partial b_j}\frac{1}{2}\sum_{i=1}^{m}\{f(\vec{\mathbf{x}}_i) - y_i\}^2$$

Here, $f(\vec{\mathbf{x}}_i) = \sum_{k=0}^{n} b_k x_{ik}$

$f(\vec{\mathbf{x}}_i) = b_0 x_{i0} + b_1 x_{i1} + \ldots + b_n x_{in}$

$$= \frac{1}{2}\sum_{i=1}^{m}\frac{\partial}{\partial b_j}\{f(\vec{\mathbf{x}}_i) - y_i\}^2$$

$$= \frac{1}{2}\sum_{i=1}^{m} 2\{f(\vec{\mathbf{x}}_i) - y_i\}\frac{\partial}{\partial b_j}\{f(\vec{\mathbf{x}}_i) - y_i\}$$

$$= \frac{1}{2}\sum_{i=1}^{m} 2\{f(\vec{\mathbf{x}}_i) - y_i\}\frac{\partial}{\partial b_j}\{\sum_{k=0}^{n} b_k x_{ik} - y_i\}$$

$$= \sum_{i=1}^{m}\{f(\vec{\mathbf{x}}_i) - y_i\}x_{ij}$$

Linear Regression: LMS Algorithm

LMS update rule (LMS stands for "least mean squares"), is also known as the Widrow-Hoff learning rule can be stated as.

Repeat until convergence {

$$b_j = b_j - \eta \sum_{i=1}^{m} \{f(\vec{\mathbf{x}}_i) - y_i\} x_{ij}; \forall j \in \{0, 1, 2, ..., n\}$$

}

This method looks at every instance in the entire training set on every step, and is called *batch gradient descent*. Note that, while gradient descent can be vulnerable to local minima in general, the optimization problem we have posed here for linear regression has only one global, and no other local, optima. Thus gradient descent always converges to the global minimum (assuming the learning rate η is not too large). Indeed, E is a convex quadratic function.

Linear Regression: LMS Algorithm

There is an alternative to batch gradient descent that also works very well.

Repeat until convergence {
    for (i = 1 to m) {
        $b_j = b_j - \eta(f(\vec{\mathbf{x}}_i) - y_i)x_{ij}; \forall j \in \{0, 1, 2, ..., n\}$
    }
}

- In this algorithm, we repeatedly run through the training set, and each time we encounter a training example, we update the parameters according to the gradient of the error with respect to that single training example only.
- This algorithm is called *stochastic gradient descent* (also *incremental gradient descent*).

Linear Regression: LMS Algorithm

- Whereas *batch gradient descent* has to scan through the entire training set before taking a single step — a costly operation if *m* is large.
- *Stochastic gradient descent* can start making progress right away, and continues to make progress with each example it looks at.
- Often, stochastic gradient descent gets *b* "close" to the minimum much faster than batch gradient descent.
- Note however that it may never "converge" to the minimum, and the parameters *b* will keep oscillating around the minimum of *E*. But in practice, most of the values near the minimum will be reasonably good approximations to the true minimum.
- For these reasons, *particularly when the training set is large, stochastic gradient descent is often preferred over batch gradient descent*.

Evaluating Goodness of Fit

- *Residuals*: The differences $y_1 - t_1, y_2 - t_2, \ldots, y_n - t_n$ between the observed/ given (y) and fitted (t) values.
- The residual can be thought of as a measure of deviation.
- The residual is a positive number if the point lies above the line and a negative number if it lies below the line.
- The *sum of squares error* (equivalently, *residual sum of squares*), denoted by *SSE*.

$$SSE = \sum \{y_i - t_i\}^2 = \sum \{y_i - f(\vec{\mathbf{x}}_i)\}^2$$

- *Sum of squared by the model* (SSM): $SSM = \sum \{f(\vec{\mathbf{x}}_i) - \bar{y}\}^2$
- *Total sum of squared* (SST): $SST = \sum \{y_i - \bar{y}\}^2$

$$SST = SSE + SSM$$

Evaluating Goodness of Fit

- The parameter σ^2 determines the amount of spread about the true regression line.
- Many large deviations (residuals) suggest a large value of σ^2, whereas deviations all of which are small in magnitude suggest that σ^2 is small.
- The estimate of σ^2 is

$$\sigma^2 = \frac{SSE}{n-2} = \frac{\sum\{y_i - t_i\}^2}{n-2}$$

- The divisor $n - 2$ is the number of *degrees of freedom* (*df*) associated with SSE. This is because, the two parameters b_0 and b_1 must first be estimated, which results in a loss of 2 *df* for simple linear regression (one *x* component).
- For multiple/multivariate regression, the *df* will be as per the number of *x* component.

Evaluating Goodness of Fit

- The *coefficient of determination*, denoted by $\boldsymbol{R^2}$.
- In order to measure how well data points fit to our line, we use a method called ***R* squared ($\boldsymbol{R^2}$).**
- This value ranges from 0 to 1. It is close to 1 if most variability observed in the target variable can be explained by the regression model.

$$R^2 = 1 - \frac{SSE}{SST} = \frac{SSM}{SST} = \frac{\sum\{f(x_i) - \bar{y}\}^2}{\sum\{y_i - \bar{y}\}^2} = \frac{(S_{XY})^2}{S_{XX}S_{YY}}$$

The numerator is the *sum of squared by the model* (SSM) and the denominator is the *total sum of squared* (SST).

Linear Regression: Least Square Approach

***Example*:** Consider the set of 10 points:

x	1	2	3	4	4	5	5	6	6	7
y	7	8	9	8	9	11	10	13	14	13

Its linear regression model is $f(x_i)=5.19+1.17x_i$. Therefore,

x	1	2	3	4	4	5	5	6	6	7
t	**6.36**	**7.53**	**8.7**	**9.87**	**9.87**	**11.04**	**11.04**	**12.21**	**12.21**	**13.38**

$$SSE = \sum\{y_i - t_i\}^2 = \sum\{y_i - f(\vec{\mathbf{x}}_i)\}^2 = 10.0301$$

$$SSM = \sum\{f(\vec{\mathbf{x}}_i) - \bar{y}\}^2 = 43.9461 \qquad \text{Here, } \bar{y} = 10.2$$

$$SST = \sum\{y_i - \bar{y}\}^2 = 53.6$$

$$R^2 = 1 - \frac{SSE}{SST} = \frac{SSM}{SST} = \frac{43.9461}{53.6} = 0.81989$$

Evaluating Goodness of Fit

- R^2 mirrors the relationship between observed and predicted values.
- The higher the value of R^2, the more successful is the simple linear regression model in explaining y variation.
- *Example*: Suppose, the coefficient of determination for a model is computed as $R^2 = 0.791$.

  That is, 79.1% of the observed variation is attributable to (can be explained by) the simple linear regression relationship.

Assignments

***Assignment* REG1:** A study was conducted at Virginia Tech to determine if certain static arm-strength measures have an influence on the "dynamic lift" characteristics of an individual. Sixteen individuals were subjected to strength tests and then were asked to perform a weightlifting test in which weight was dynamically lifted overhead. The data are given here.

Instance	*Arm Strength, x*	*Dynamic Lift, y*	*Instance*	*Arm Strength, x*	*Dynamic Lift, y*
1	17.3	71.7	9	27.3	75.7
2	19.3	48.3	10	29.3	88.3
3	19.5	88.3	11	29.5	88.3
4	19.7	75.0	12	29.7	95.0
5	22.9	91.7	13	32.9	91.7
6	23.1	100.0	14	33.1	100.0
7	26.4	73.3	15	36.4	103.3
8	26.8	65.0	16	36.8	105.0

(*a***)** Estimate the *linear regression* curve, $f(x_i)=b_1+b_2x_i$. (*b*) Find a point estimate of $f(30)$.

Assignments

***Assignment* REG2:** A study was made on the amount of converted sugar in a certain process at various temperatures. The data were coded and recorded as follows:

Temperature, *x*	**Converted Sugar, *y***
1.0	8.1
1.1	7.8
1.2	8.5
1.3	9.8
1.4	9.5
1.5	8.9
1.6	8.6
1.7	10.2
1.8	9.3
1.9	9.2
2.0	10.5

Estimate the *linear regression* curve and estimate the amount of converted sugar produced when the coded temperature is 1.75.

For Assignment 1 and 2 use **least squares approach**, **batch gradient descent** and **stochastic gradient descent**.

Assignments

***Assignment* REG3:** The following data represent the chemistry grades for a random sample of 12 freshmen at a certain college along with their scores and the number of class periods missed by the students. The complete data are shown.

Student	*Grade, y*	*Test Score, x_1*	*Missed Class, x_2*	*Student*	*Grade, y*	*Test Score, x_1*	*Missed Class, x_2*
1	85	65	1	7	94	65	2
2	74	50	7	8	98	70	5
3	76	55	5	9	81	55	4
4	90	65	2	10	91	70	3
5	85	55	6	11	76	50	1
6	87	70	3	12	74	55	4

(*a*) Fit the multiple linear regression equation. (*b*) Estimate the chemistry grade for a student who has an intelligence test score of 60 and missed 4 classes.

Assignments

***Assignment* REG4:** An experiment was conducted to determine if the weight of an animal can be predicted after a given period of time on the basis of the initial weight of the animal and the amount of feed that was eaten. The following data, measured in kilograms, were recorded:

Final Weight, y	Initial Weight, x_1	Feed Weight, x_2
95	42	272
77	33	226
80	33	259
100	45	292
97	39	311
70	36	183
50	32	173
80	41	236
92	40	230
84	38	235

(*a*) Fit the multiple linear regression equation. (*b*) Predict the final weight of an animal having an initial weight of 35 kilograms that is given 250 kilograms of feed.

Books:

1. Applied Statistics and Probability for Engineers by Douglas C. Montgomery and George C. Runger, John Wiley & Sons, 5ed.
2. Foundations of Machine Learning by Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar, MIT Press, Cambridge.
3. Machine Learning: A Probabilistic Perspective by Kevin P. Murphy, MIT Press, Cambridge.