MACHINE LEARNING CDS503

Topic 7: Linear Regression

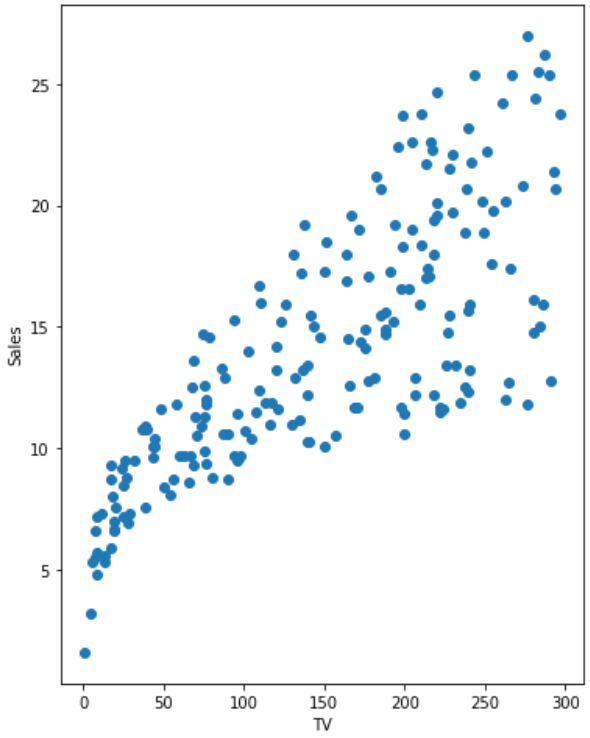
Mohd Halim Mohd Noor, PhD

Outline

- Introduction
- Linear Regression
- Parameter Estimation
  - Least Squares
  - Gradient Descent
- Performance Metrics
- Assumptions

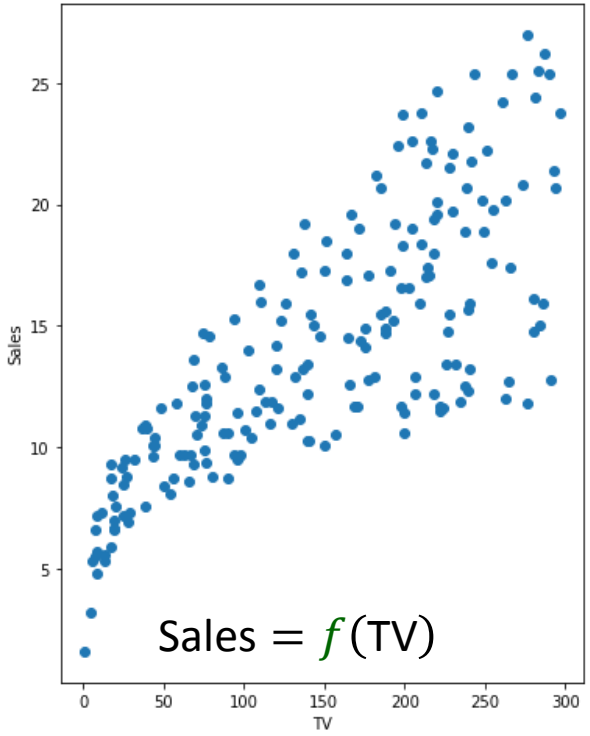
Introduction

- Suppose we have two-dimensional data, X and Y
- How the variables are related to each other? e.g. advertising (TV) and sales, people's weights and heights, study time and grades, etc.



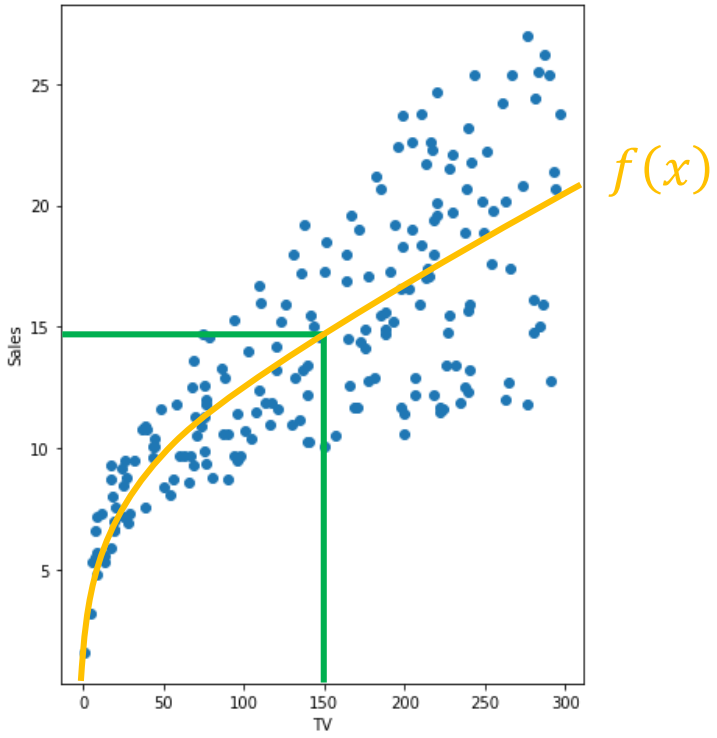
Introduction

- Model the relationship between x and Y



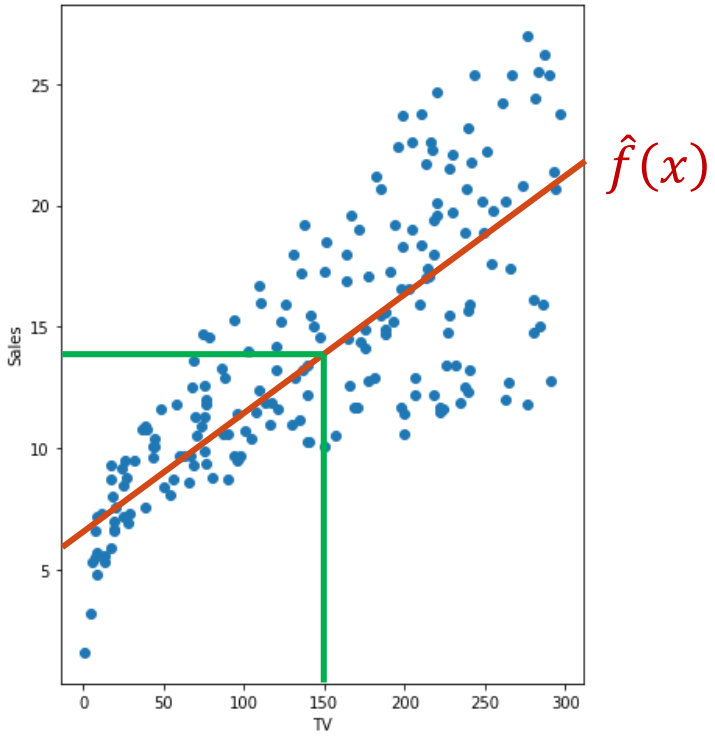
Make prediction of Y at new point x

Introduction



What is the expected value (average) of Y at $x = 150$?

Introduction



Construct a predictor $\hat{f}(x)$ that is a good estimate of the regression function $f(x)$

Linear Regression

Linear Regression

- Linear models do not work for everything in our world, but they do work well in many cases
- If the data tends to have linear associations, you may be well-served by a linear model

Number of rooms and house price

GDP and economic growth

Linear Regression

- Linear regression is a supervised learning approach that models the dependence of Y on covariate $x_1, x_2, \dots, x_p$ as being linear
- Suppose a dataset
  - p independent variables (attributes), $\mathbb{x}_i = (x_{i1}, x_{i2}, \dots x_{ip})$ where $i = 1, \dots, N$
  - 1 dependent variable (target), Y

Linear Regression

- Linear regression is a supervised learning approach that models the dependence of Y on covariate $x_1, x_2, \dots, x_p$ as being linear
- Suppose a dataset
  - p independent variables (attributes), $\mathbb{x}_i = (x_{i1}, x_{i2}, \dots x_{ip})$ where $i = 1, \dots, N$
  - 1 dependent variable (target), Y

$$\hat{f}(\mathbb{x}_i) = w_0 x_{i0} + \sum_{j=1}^{p} w_j x_{ij} = \mathbb{w}^T \mathbb{x}_i$$

- $\mathbb{w}$ is the weight (coefficient) vector
- $x_{i0} = 1$

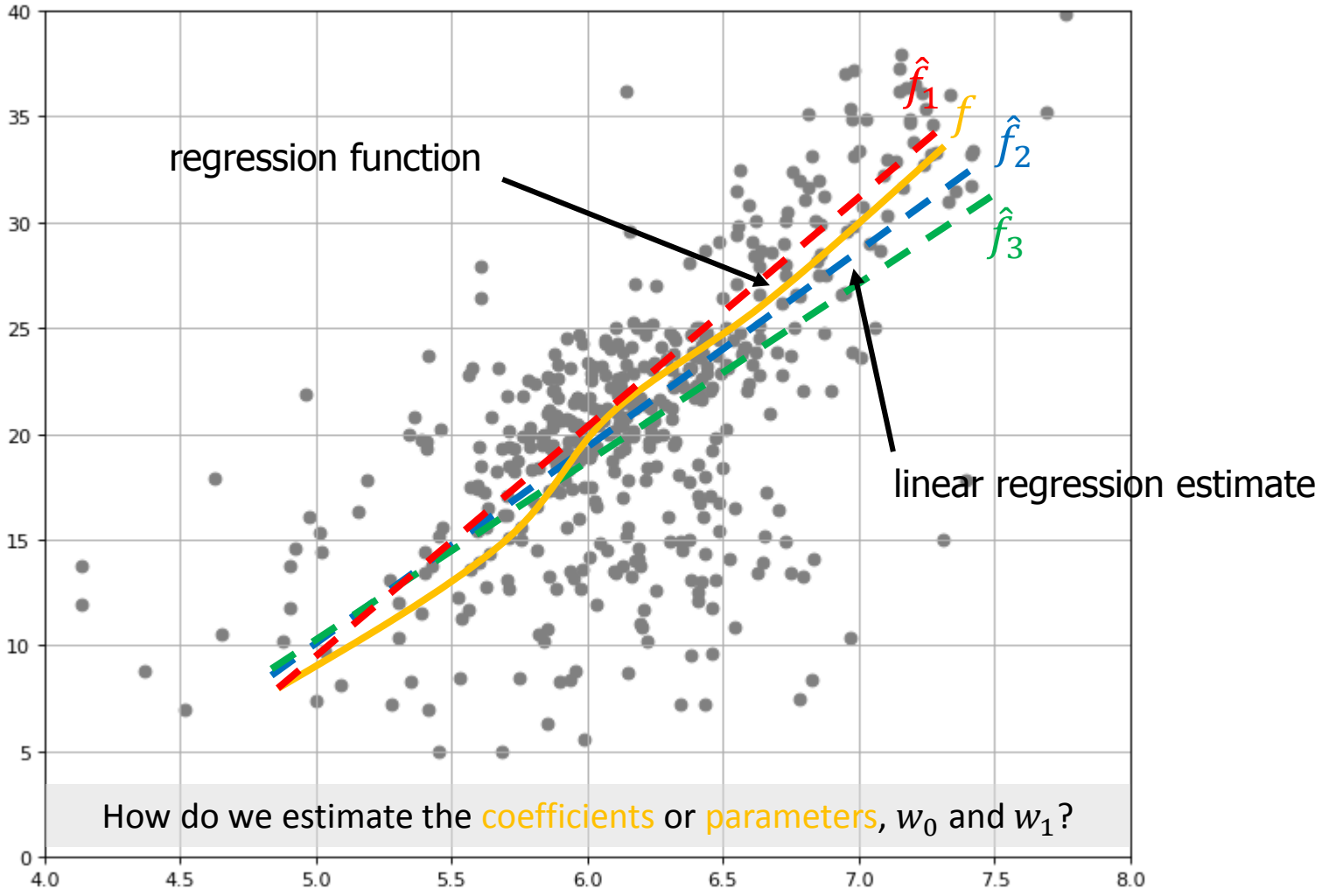
Linear Regression

- Suppose the data is one dimensional, then the linear regression model is

$$\hat{f}(\mathbb{x}_i) = \mathbb{w}^T \mathbb{x}_i = w_0 + w_1 x_i$$

- w_0 is the intercept
- w_1 is the slope

Linear Regression



Parameter Estimation

Parameter Estimation

- Suppose that we have p dimensional data (x_i, Y_i), $i = 1, \dots, N$

$$\mathbb{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_N \end{bmatrix} \quad \mathbb{X} = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{Np} \end{bmatrix}$$

- The coefficients, w_i can be estimated by minimizing the sum of squared errors (cost function), J

$$J(\mathbb{w}) = \sum_{i=1}^{N} \left(Y_i - \hat{f}(\mathbb{x}_i) \right)^2$$

$$= \sum_{i=1}^{N} \left(Y_i - \left[\hat{w}_0 x_{01} + \hat{w}_1 x_{i1} + \cdots + \hat{w}_p x_{ip} \right] \right)^2$$

$$= \sum_{i=1}^{N} \left(Y_i - \hat{w}_0 x_0 + \sum_{j=1}^{p} \hat{w}_j x_{ij} \right)^2 = \left[\mathbb{Y} - \mathbb{w}\mathbb{x} \right]^2$$

Parameter Estimation (SLR)

- One dimensional data (x_i, Y_i), $i = 1, \ldots, N$

$$\mathbb{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{bmatrix} \quad \mathbb{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

- The coefficients, w_i can be estimated by minimizing the sum of squared errors (cost function), J

$$J(\mathbb{w}) = \sum_{i=1}^{N} \left(Y_i - \hat{f}(x_i) \right)^2$$

$$= \sum_{i=1}^{N} (Y_i - [\hat{w}_0 x_0 + \hat{w}_1 x_i])^2 = [\mathbb{Y} - \mathbb{w}\mathbb{X}]^2$$

- $e = Y_i - \hat{f}(x_i)$ is also called a residual

Least Squares

- The cost function can be defined as

$$J(\mathbb{w}) = [\mathbb{Y} - \mathbb{w}\mathbb{x}]^2 = (\mathbb{Y} - \mathbb{w}\mathbb{x})^T(\mathbb{Y} - \mathbb{w}\mathbb{x})$$

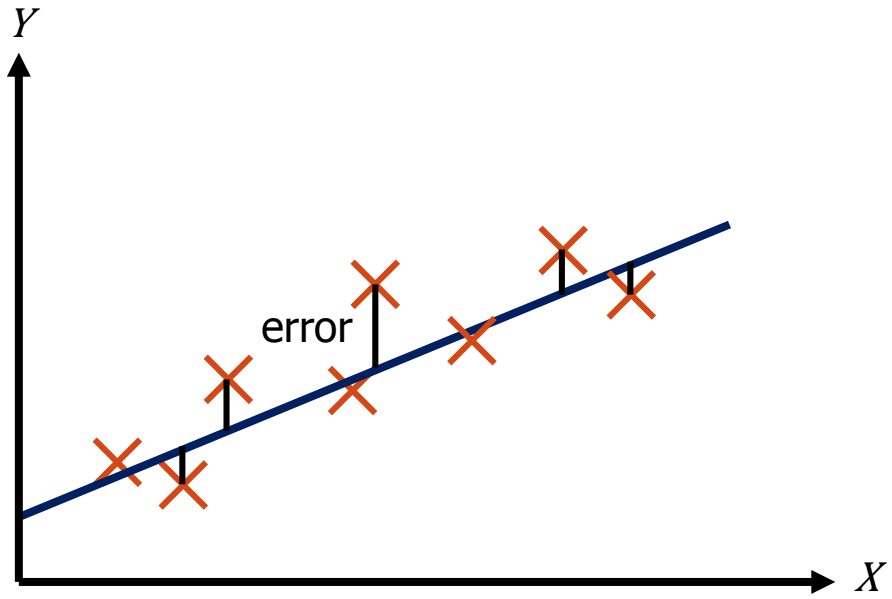
- Differentiating J with respect to $\mathbb{w}$

$$\frac{\partial J(\mathbb{w})}{\partial \mathbb{w}} = -2\mathbb{x}^T(\mathbb{Y} - \mathbb{w}\mathbb{x})$$

- Set the derivative to zero to solve for $\mathbb{w}$

$$\mathbb{w} = (\mathbb{x}^T\mathbb{x})^{-1}\mathbb{x}^T\mathbb{Y}$$

Least Squares in 1-dimension



- The blue line shows least squares fit for the regression of X onto Y
- Lines from the observed points to the regression line illustrate the residuals (errors)
- For any other choice of coefficients, the sum of squared errors would be larger than the blue line

Numerical Example

- Suppose that we have data (x_i, Y_i), $i = 1, \ldots, 5$

$$\mathbb{Y} = \begin{bmatrix} 2.0 \\ 2.6 \\ 2.2 \\ 3.2 \\ 3.0 \end{bmatrix} \quad \mathbb{X} = \begin{bmatrix} 1 & 3.1 \\ 1 & 2.9 \\ 1 & 3.5 \\ 1 & 4.3 \\ 1 & 3.6 \end{bmatrix}$$

- Calculate $(\mathbb{X}^T\mathbb{X})^{-1}$ and $\mathbb{X}^T\mathbb{Y}$

$$(\mathbb{X}^T\mathbb{X})^{-1} = \begin{bmatrix} 10.5685 & -2.9795 \\ -2.9795 & 0.8561 \end{bmatrix}$$

$$\mathbb{X}^T\mathbb{Y} = \begin{bmatrix} 13 \\ 46 \end{bmatrix}$$

$$\mathbb{W} = \begin{bmatrix} 0.3356 \\ 0.6507 \end{bmatrix}$$

Least Squares

- It might happen that $\mathbb{X}^T\mathbb{X}$ is singular (is not invertible)
- $\mathbb{X}$ are nearly collinear or perfectly correlated e.g. $x_2 = 3x_1$
- $\mathbb{W}$ cannot be solved

Friedman, J., Hastie, T. and Tibshirani, R., 2001. *The elements of statistical learning* (Vol. 1, No. 10). New York, NY, USA:: Springer series in statistics.

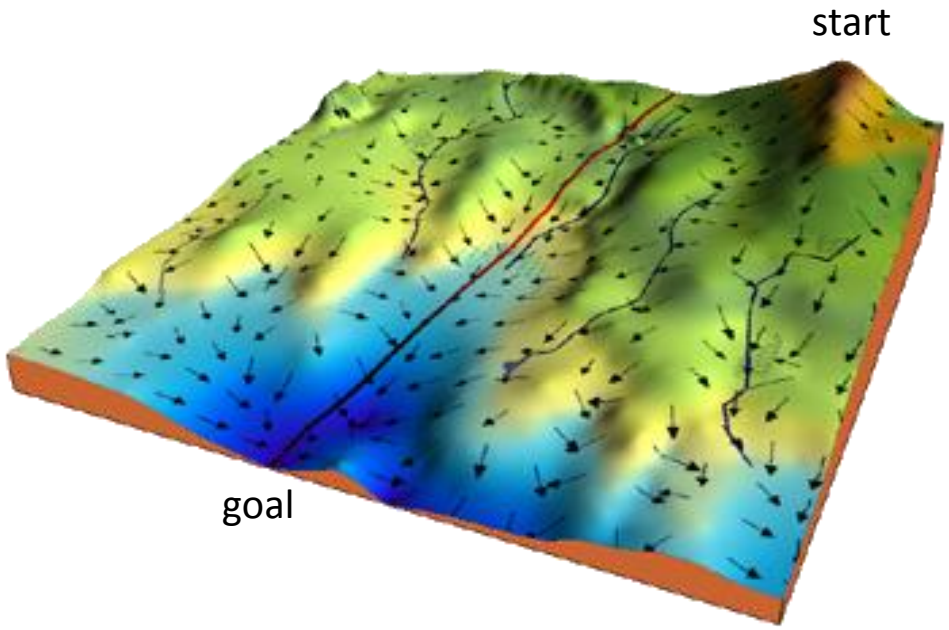
Gradient Descent (GD)

- Optimization algorithm finds the coefficients that minimize the cost function (error in prediction)
- Suppose $\mathbb{w}$ denotes the weight vector and $\mathrm{E}(\mathbb{w}|\mathbb{x})$ is the error with parameters $\mathbb{w}$ on the given feature vector $\mathbb{x}$

$$\mathbb{w} = \arg \min_{\mathbb{w}} \mathrm{E}(\mathbb{w}|\mathbb{x})$$

How GD works?

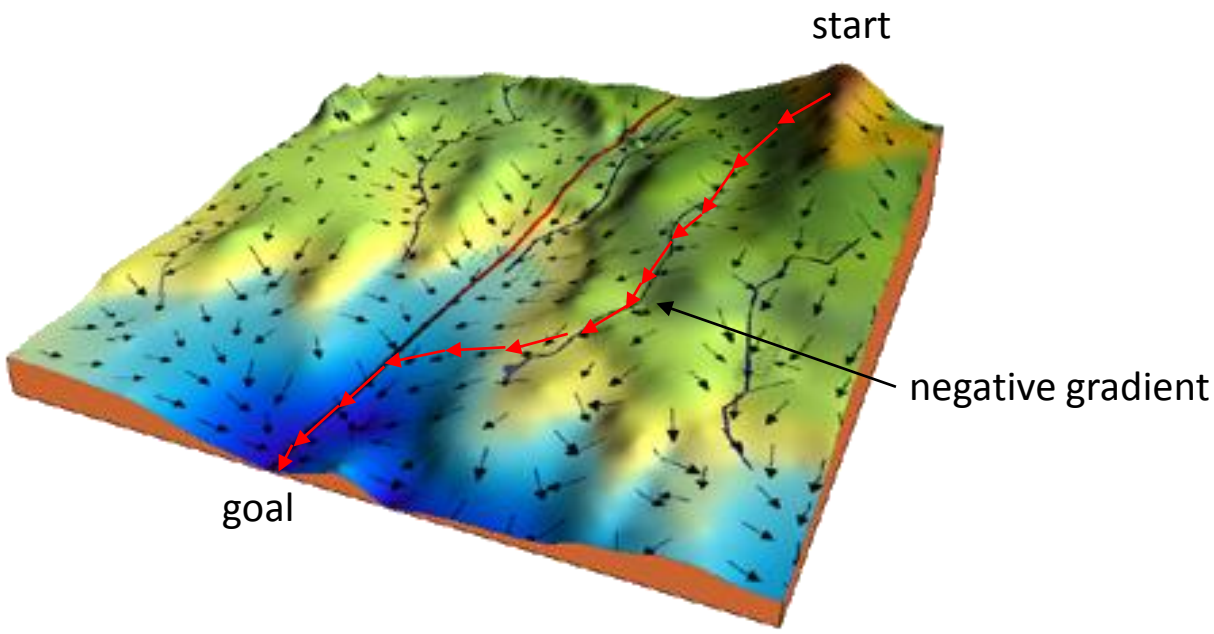
- Think of the cost function (error) as a mountain that you are hiking down
- Your goal is to reach the bottom (to get to minimum error)
- To accomplish this is by proceeding through the path that makes you descend (opposite of the steepest mountain direction upwards)



https://ml-cheatsheet.readthedocs.io/en/latest/gradient_descent.html

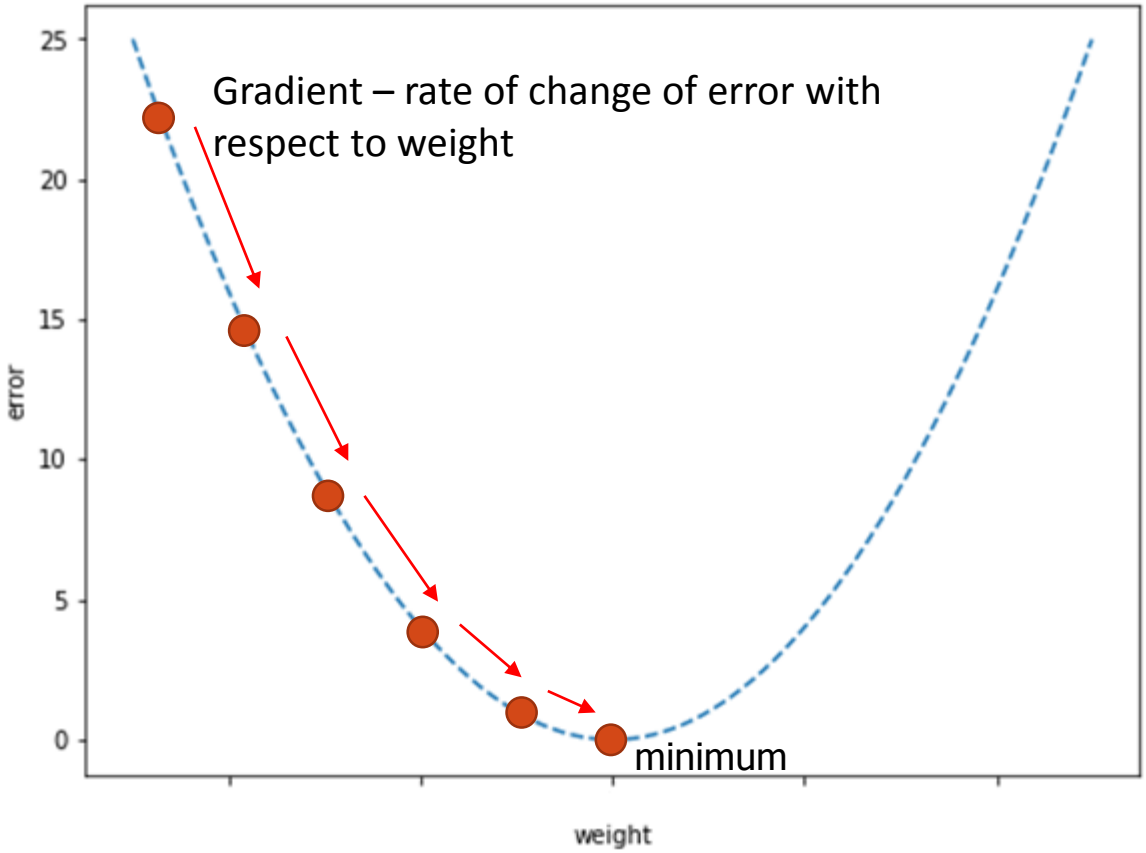
How GD works?

- Think of the cost function (error) as a mountain that you are hiking down
- Your goal is to reach the bottom (to get to minimum error)
- To accomplish this is by proceeding through the path that makes you descend (opposite of the steepest mountain direction upwards)



https://ml-cheatsheet.readthedocs.io/en/latest/gradient_descent.html

Gradient



"A gradient measures how much the output of a function changes if you change the inputs a little bit." — Lex Fridman (MIT)

Gradient Descent

- Let $J(w)$ be the cost function (error) of parameter vector $\mathbb{w}$

$$J(\mathbb{w}) = \frac{1}{2}\sum_{i=1}^{N}\left(Y_i - w_0 + \sum_{j=1}^{p} w_j x_{ij}\right)^2 = \frac{1}{2}\sum_{i=1}^{N}\left(Y_i - \hat{Y}_i\right)^2$$

- Define the gradient of $J(\mathbb{w})$ to be

$$\nabla_{\mathbb{w}}(\mathbb{w}) = \begin{bmatrix} \frac{\partial J(w_0)}{\partial w_0} \\ \vdots \\ \frac{\partial J(w_p)}{\partial w_p} \end{bmatrix}$$

$$\frac{\partial J(w_j)}{\partial w_j} = \sum_{i=1}^{N} -x_j\left(Y_i - \hat{Y}_i\right)$$

Gradient Descent

- To update the weights

$$w_j := w_j - \alpha \frac{\partial J(w_j)}{\partial w_j}$$

- α is the learning rate (step size)
- if α is too small, it will take much time to converge
- if α is too big, it might overshoot and diverge (infinite loop)

Algorithm

- Define initial parameter values, $\mathbb{w} = \{w_0, w_1, \ldots, w_p\}$
- Iteratively change $\mathbb{w}$ to minimize $J(\mathbb{w})$ until it is converged (reach local minimum)
  - Calculate the gradient

$$\frac{\partial J(w_j)}{\partial w_j} = \sum_{i=1}^{N} -x_j(Y_i - \hat{Y}_i)$$

  - Update the weights

$$w_j := w_j - \alpha \frac{\partial J(w_j)}{\partial w_j}$$

Numerical Example

- Suppose that we have data (x_i, Y_i), $i = 1, \ldots, 5$

$$\mathbb{Y} = \begin{bmatrix} 2.0 \\ 2.6 \\ 2.2 \\ 3.2 \\ 3.0 \end{bmatrix} \quad \mathbb{x} = \begin{bmatrix} 3.1 \\ 2.9 \\ 3.5 \\ 4.3 \\ 3.6 \end{bmatrix}$$

- Step 1: Define initial values of weights

$$\mathbb{w} = \begin{bmatrix} 0.45 \\ 0.75 \end{bmatrix}$$

- Step 2: Calculate the predictions

$$\hat{\mathbb{Y}} = \begin{bmatrix} 2.8 \\ 2.6 \\ 3.1 \\ 3.7 \\ 3.2 \end{bmatrix}$$

$$SSE = 1.74$$

- Step 3: Calculate the gradient

$$\frac{\partial J(w_j)}{\partial w_j} = \sum_{i=1}^{N} -x_j\left(Y_i - \hat{Y}_i\right)$$

$$\frac{\partial J(w_0)}{\partial w_0} = 0.8 + 0.0 + 0.9 + 0.5 + 0.2 = 2.3$$

$$\frac{\partial J(w_1)}{\partial w_1} = 2.4 + 0.1 + 3.1 + 2.0 + 0.5 = 8.12$$

$$\nabla_{\mathbb{W}}(\mathbb{W}) = \begin{bmatrix} 2.30 \\ 8.12 \end{bmatrix}$$

- Step 4: Update the weights and suppose $\alpha = 0.01$

$$w_j := w_j - \alpha \frac{\partial J(w_j)}{\partial w_j}$$

$$w_0 = 0.45 - 0.01 \times 2.30 = 0.427$$

$$w_1 = 0.75 - 0.01 \times 8.12 = 0.669$$

- Step 5: Calculate the (new) predictions and SSE

$$SSE = 0.665$$

- The SSE has decreased from 1.615 to 0.665
- Repeat step 2-5 until the weights does not significantly reduces the error or maximum iterations has reached.

Performance Metrics

Performance Metrics

- Mean absolute error
- Root mean squared error
- R-squared

Mean Absolute Error

- Measures the average magnitude of the errors of the predictions

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|Y_i - \hat{Y}_i\right|$$

- If absolute value is not taken, the average error becomes the Mean Bias Error
  - Measure average model bias

Root Mean Squared Error

- Measures the square root of the average of the error (squared difference)
- Euclidean distance between the actual outputs and prediction outputs

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(Y_i - \hat{Y}_i\right)^2}{N}}$$

R-squared

- Measures the goodness of fit of the linear regression model

$$R^2 = 1 - \frac{SSE}{SST}$$

- SSE is the sum of squared errors of the regression model

$$SSE = \sum_{i=1}^{N} \left(Y_i - \hat{Y}\right)^2$$

- SST is the sum of errors of the baseline model
  - The baseline model always predicts mean of Y

$$SST = \sum_{i=1}^{N} (Y_i - \bar{Y})^2$$

  - where $\bar{Y} = \frac{1}{N}\sum_{i=1}^{N} Y_i$

R-squared

- $R^2 = 0$ means worst possible regression model
- $R^2 = 1$ means perfect model
- If $R^2 = 0.6$, it implies that 60% variations in dependent variable Y can be explained by the independent variables in the regression model

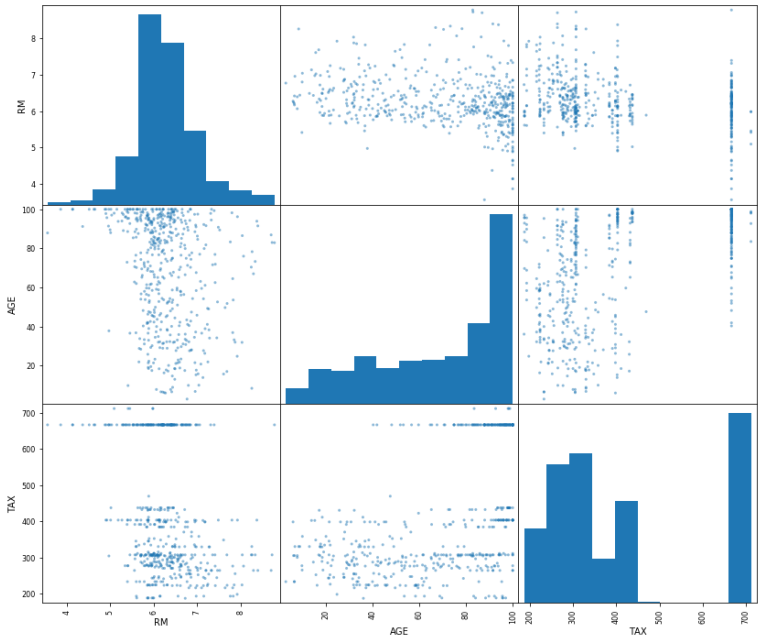
Assumptions

Assumption of Linear Regression

- Linear relationship between the attributes (features) and target
  - Linear model captures only linear relationship
  - e.g. house price increases as # of rooms increases

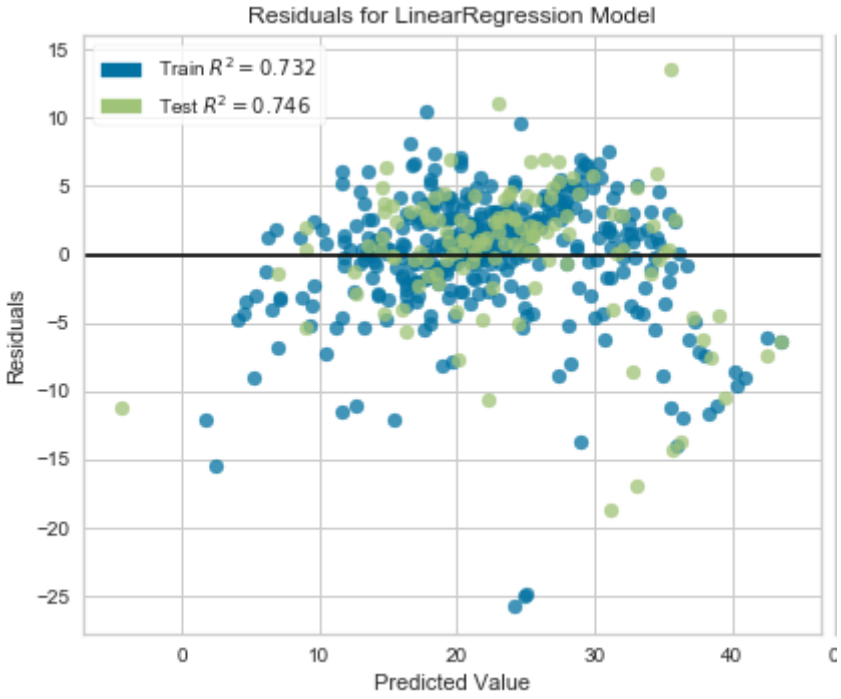
Assumption of Linear Regression

- Little or no multicollinearity between the attributes (features)
  - Little or no correlation among the attributes
  - The coefficients represent the mean change in the target for each unit change of the attributes
  - If features are correlated, changes in one attribute in turn change another attribute(s)



Assumption of Linear Regression

- Homoscedasticity
  - The variance of the error term (residuals) is roughly the same for all values of attributes
  - The residual is the difference between the observed value of the dependent variable (y) and the predicted value ($\hat{y}$)



Assumption of Linear Regression

- Normal distribution of error terms
  - The error follow a normal distribution
  - If the sample size is large, the assumption is not needed

UNIVERSITI SAINS MALAYSIA

End