Lesson 7
Regression Models

Mathematics and Statistics for Data Science

Tapanan Yeophantong
Vincent Mary School of Science and Technology
Assumption University

Content

- Regression techniques & their applications
- Metrics for evaluating regression models

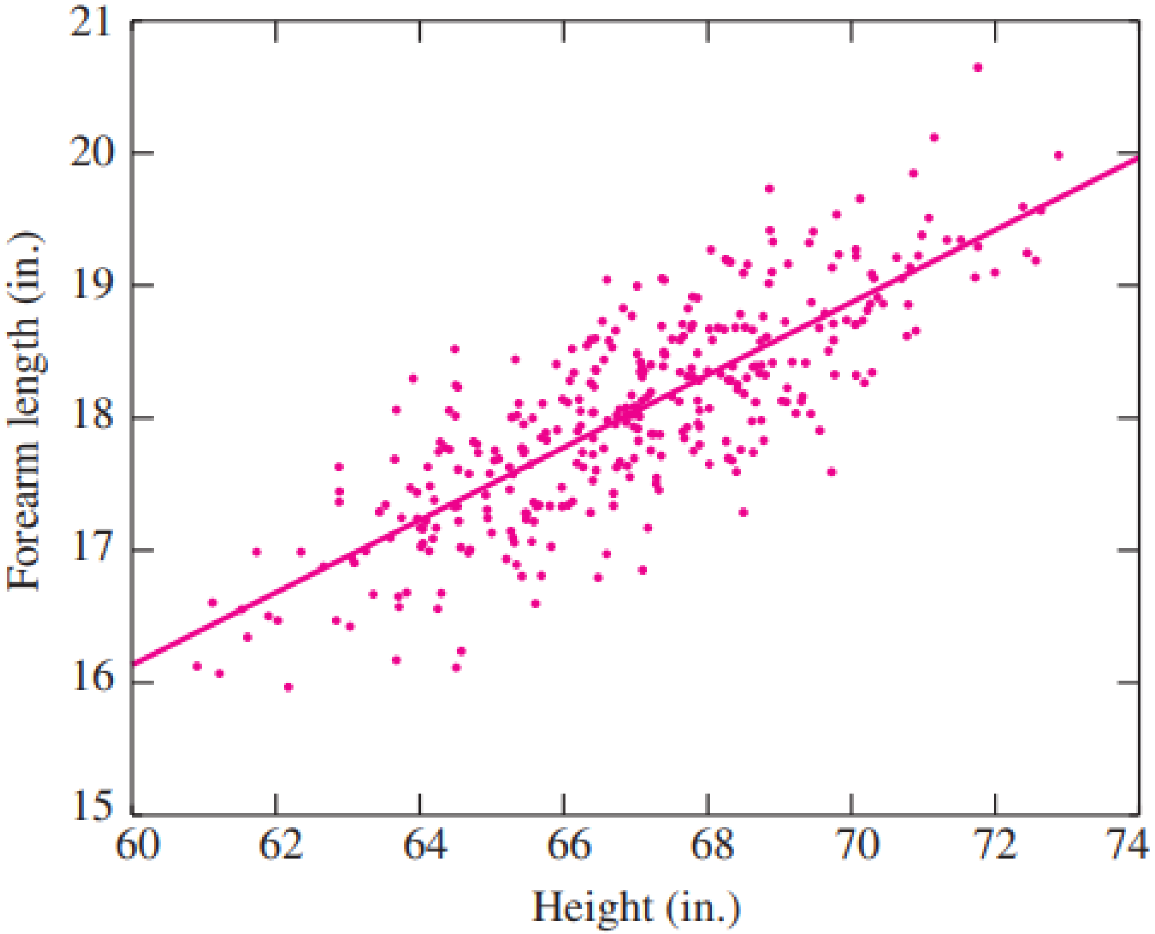
Regression Analysis (1)

- Regression analysis is used to analyse *multivariate* data, construct a model to “fit” the data, and use the model to make inferences.
- The most common form is linear regression.
- Regression analysis is generally used for:
  - Inferring causal relationships (confirmatory)
  - Prediction and forecasting (predictive)

Regression Analysis (2)

- A regression model consists of:
  - A set of independent variables ($X_1, X_2, X_3, \ldots, X_n$)
  - A dependent variable (Y_i)
  - Error terms (e_i)
  - Some unknown scalar/vector parameters (e.g. β)
- The goal of regression analysis is to estimate a function $f(X_i, \beta)$ that most closely fits the data.

Goodness of a "Fit"



Correlation

- The points tend to slope upward and to the right, indicating that taller men tend to have longer forearms.
- We say that there is a positive association between height and forearm length.
- The slope is approximately constant throughout the plot, indicating that the points are clustered around a straight line.
- The line superimposed on the plot is a special line known as the least-squares line - the line that fits the data best.

Correlation Coefficient

- Correlation coefficient is computed by:

$$r = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\,\bar{y}}{\sqrt{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2}\,\sqrt{\sum_{i=1}^{n} y_i^2 - n\bar{y}^2}}$$

- The coefficient is always between -1 and 1.

Fitting Techniques

- (Linear) regression/(ordinary) least squares
  - Choose parameters for the regression function, typically linear, by the principle of least squares: minimizing the sum of squared errors.
  - That is, solves the function in one go.
- Non-linear regression/least squares
  - Estimate the parameters of the model by a function and to refine the parameters by successive iterations.
  - That is, approximates the function parameters over time.

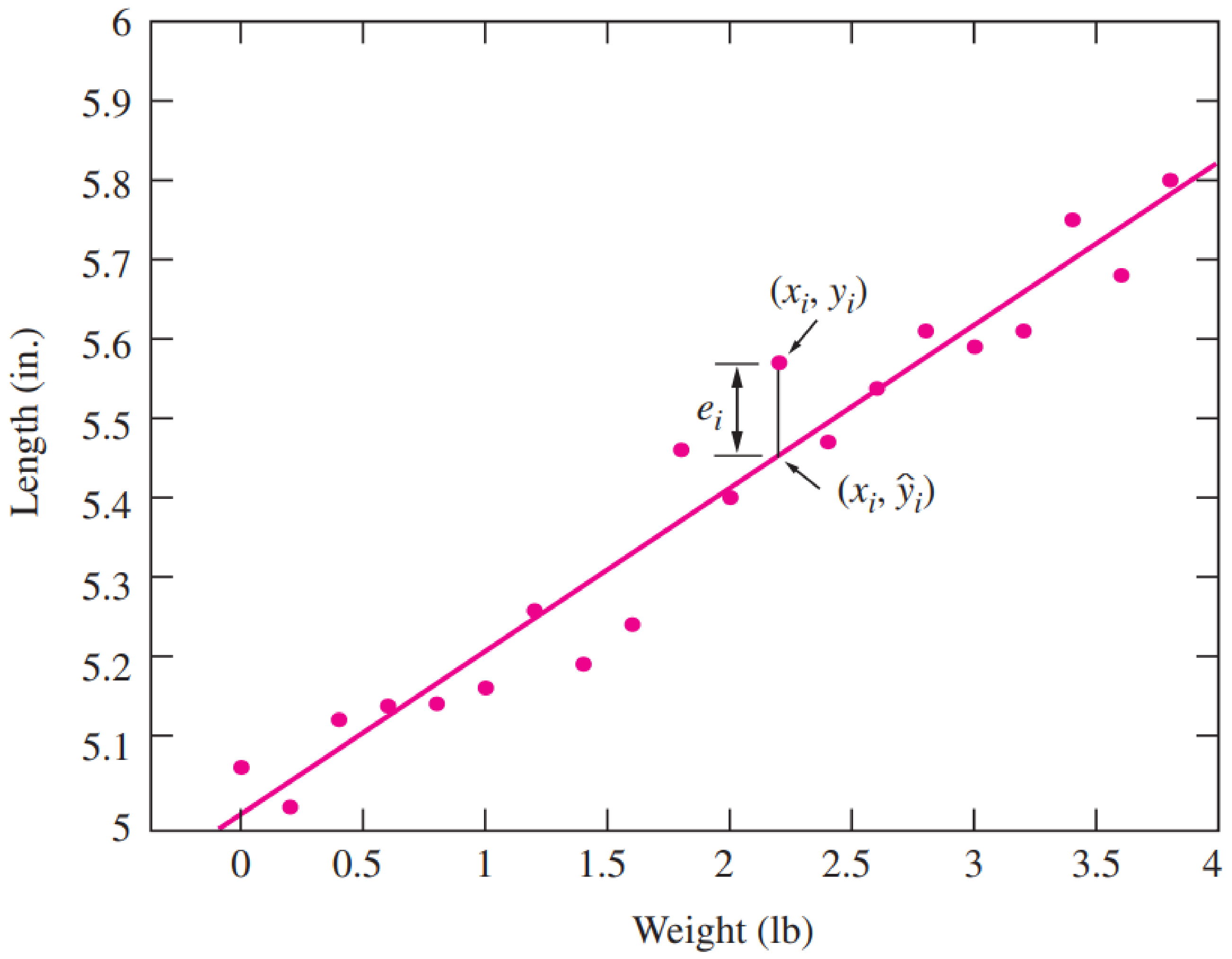
Regression Models

- Linear regression models
  - Simple linear regression
  - Multiple linear regression
  - Generalised linear model
- Non-linear regression models
  - Logistic regression
  - Polynomial regression
  - Non-parametric regression

Least-Squares

- A standard approach in regression analysis.
- In any least-squares methods, the "best fit" is to try to minimize the sum of squared residuals between the observed and predicted values by the model.
- Two categories:
  - Linear or ordinary least squares
  - Nonlinear least squares

Concept of “Least Squares”



Linear Models

- y_i is called the dependent variable.
- x_i is called the independent variable.
- β_0 and β_1 are the regression (least-squares) coefficients.
- ε_i is called the error.

- The equation is called a linear model: $l_i = \beta_0 + \beta_1 x_i$

Computing the Equation

- Computing the error:

$$e_i = y_i - \widehat{y}_i = y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_i$$

- Thus, β_O andβ_1 are the quantities that minimise the sum:

$$S = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_i)^2$$

- Finally:

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n} (x_i - \overline{x})^2}$$

$$\widehat{\beta}_0 = \overline{y} - \widehat{\beta}_1 \overline{x}$$

Non-linear Least Squares

- A form of least squares analysis used to fit observations with a non-linear model.
- It is a form of non-linear regression.
- The idea is to fit a linear model and refine the parameters by successive iterations.
- Examples : SVD, Gradient Descent.

Non-parametric Algorithms

- A category of regression analysis where the predictor does not take a pre-determined form.
- That is, no assumption on the distribution of the data.
- Model is constructed from information derived from data, and hence requires larger sample sizes.
- Examples: nearest neighbours, neural networks.

Regression on Scikit-Learn

- Linear Models
  - LinearRegression (simple)
  - Ridge (L2)
  - Lasso (L1)
  - ElasticNet (L1 + L2)
- Based on the classifiers
  - DecisionTreeRegressor
  - KNeighborsRegressor

Gradient Descent

- Gradient descent search finds a weight vector that minimises E by
  - Starting with an arbitrary initial weight vector.
  - Repeatedly modifying it in small steps.
  - At each step, weight vector is modified in the direction that produces the steepest descent along the error surface.

Fundamentals of Gradient Descent

- Negated derivative gives direction of steepest descent.
- The gradient (derivative) of E with respect to each component of the vector w

$$\nabla E[\vec{w}] \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \cdots \frac{\partial E}{\partial w_n} \right]$$

- Notice $\nabla E[\vec{w}]$ is a vector of partial derivatives.
- Specifies the direction that produces steepest increase in E.
- Negative of this vector specifies direction of steepest decrease.

Gradient Descent Rule (1)

- The gradient descent training rule updates weights according to the following formula:

$$\vec{w} \leftarrow \vec{w} + \Delta \overrightarrow{w}$$

- where

$$\Delta \overrightarrow{w} = -\eta \nabla E[\overrightarrow{w}]$$

- η is the positive constant learning rate, determining the step size of gradient descent search.

Gradient Descent Rule (2)

- Written in component form:

$$w_i \leftarrow w_i + \Delta w_i$$

  - where

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

Weight Update Rule

- Pick an initial random weight vector.
- Apply the linear unit to all training examples, then compute for each weight according to

$$\Delta w_i = \eta \sum_d (t_d - o_d) x_{id}$$

- Update each wi by adding Δw_i then repeat process.

Gradient Descent Algorithm

- Initialize each w_i to some small random value
- Until the termination condition is met, do
  - Initialize each Δw_i to zero.
  - For each $\langle \vec{x}, \vec{t} \rangle$ in training_examples, do
    - Input the instance $\vec{x}$ to the unit and compute the output o
    - For each linear unit weight w_i, do

$$\Delta w_i \leftarrow \Delta w_i + \eta(t - o)x_i$$

  - For each linear unit weight w_i, do

$$w_i \leftarrow w_i + \Delta w_i$$

Stochastic Gradient Descent

- Stochastic gradient descent updates wi for each example.
- Update weight as each sample is processed, using:

$$\Delta w_i \leftarrow \Delta w_i + \eta(t - o)x_i$$

SGD Algorithm

- Initialize each w_i to some small random value
- Until the termination condition is met, do
  - Initialize each Δw_i to zero.
  - For each $\langle \vec{x}, \vec{t} \rangle$ in training examples, do
    - Input the instance $\vec{x}$ to the unit and compute the output o
    - For each linear unit weight w_i, do

$$w_i \leftarrow w_i + \eta(t - o)x_i$$

Coefficient of Determination

- R2 regression score computed by dividing sum of squares of residuals (SS_{res}) by the total sum of squares (SS_{tot}):

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

- R2 score closer to 1.0 means the regression is good.

Mean Squared Error

- Mean Squared Error (MSE) is simply the average (mean) of the squared error:

$$\text{MSE}(y, \hat{y}) = \frac{1}{n_\text{samples}} \sum_{i=0}^{n_\text{samples}-1} (y_i - \hat{y}_i)^2.$$