Introduction to Data Mining (DATS 6103 - 10)

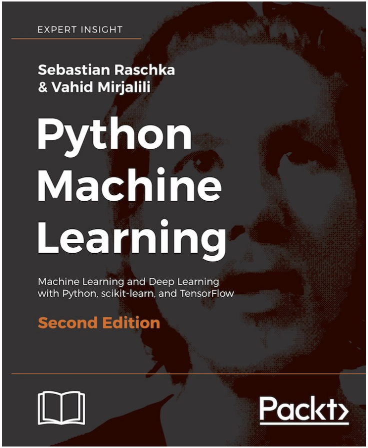
Linear Regression

Yuxiao Huang

Data Science, Columbian College of Arts & Sciences
George Washington University
yuxiaohuang@gwu.edu

June 12, 2018

Reference



Picture courtesy of the website of the book code repository and info resource

Reference

- This set of slices is an excerpt of the book by Raschka and Mirjalili, with some trivial changes by the creator of the slides
- Please find the reference to and website of the book below:
    - *Raschka S. and Mirjalili V. (2017). Python Machine Learning. 2nd Edition.*
    - `https://sebastianraschka.com/books.html`
- Please find the website of the book code repository and info resource below:
    - `https://github.com/rasbt/python-machine-learning-book-2nd-edition`

Overview

1. The linear regression model
2. Exploring and visualizing datasets
3. Implementing linear regression models
4. Evaluating regression models and diagnosing common problems

Simple (univariate) linear regression

- Simple linear regression expresses the relationship between two continuous-valued variables, x and y:

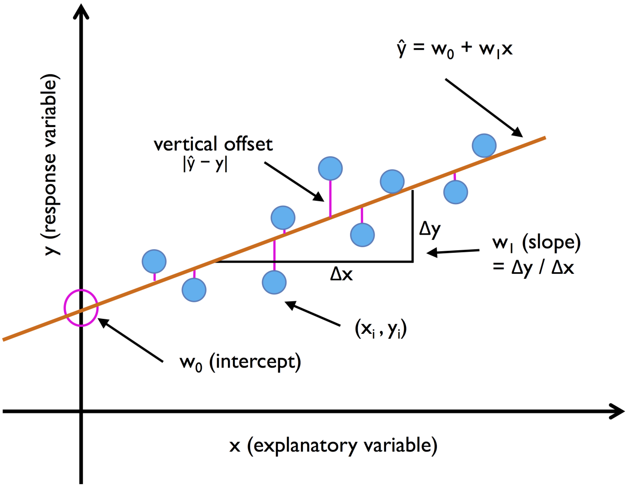
$$y = w_0 + w_1 x$$

- Here:
    - x: explanatory variable (or independent variable, regressor)
    - y: response variable (or dependent variable, regressand)
    - w_0: intercept
    - w_1: slope
- The goal is to:
    1. learn w_0 and w_1
    2. predict the value of y based on the value of x

The best fitting line (regression line)

- Linear regression can be understood as finding the best-fitting straight line through the sample points, as shown in Figure 1 (see next page)
- The best-fitting line is also called the regression line
- The vertical lines from the regression line to the sample points are the errors of our prediction (offsets, or residuals)

Figure 1



ŷ = w0 + w1x
vertical offset
|ŷ − y|
y (response variable)
Δy
w1 (slope)
= Δy / Δx
Δx
(xi , yi)
w0 (intercept)
x (explanatory variable)

Multiple linear regression

- We can generalize simple linear regression with only one explanatory variable to a model with multiple explanatory variables:

$$y = w_0 x_0 + w_1 x_1 + \cdots + w_m x_m = \sum_{i=0}^{m} w_i x_i = \mathbf{w}^T \mathbf{x}$$

- Such model is called multiple linear regression

Visualizing the important characteristics of a dataset

- Exploratory Data Analysis (EDA) is an important and recommended first step prior to the training of a machine learning model
- The graphical EDA toolbox may help
  - visually detect the presence of outliers
  - the distribution of the data
  - the relationship between features
- See details in `ch10.ipynb`

Two kinds of useful graphical summaries

- Scatterplot matrix: allows us to visualize the pair-wise correlations between different features
- Correlation matrix:
  - a square matrix that contains the Pearson product-moment correlation coefficients (or Pearson's r), which measures the linear dependence between pairs of features
  - can be calculated as the covariance between two features x and y, divided by the product of their standard deviations

$$r = \frac{\sum_{i=1}^{n}\left[\left(x^{(i)} - \mu_x\right)\left(y^{(i)} - \mu_y\right)\right]}{\sqrt{\sum_{i=1}^{n}\left(x^{(i)} - \mu_x\right)^2}\sqrt{\sum_{i=1}^{n}\left(y^{(i)} - \mu_y\right)^2}} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

  - identical to a covariance matrix computed from standardized data
- Combine the two summaries for choosing explanatory variables
- See details in `ch10.ipynb`

Estimating the parameters with gradient descent

- The parameters of the regression can be approximated by minimizing the cost function
- The cost function here is the Ordinary Least Squares (OLS) function

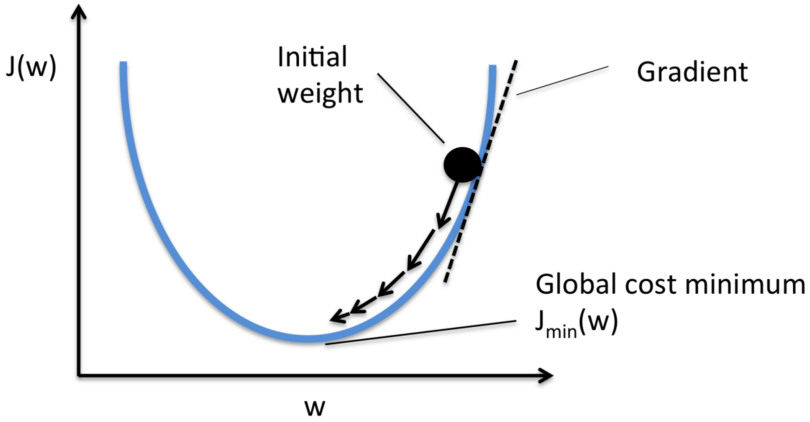
$$J(w) = \frac{1}{2} \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

- We can minimize the cost function to learn the weights via optimization algorithms, such as Gradient Descent (GD)
  - the parameters are updated as follows

$$w = w + \Delta w \quad \text{where} \quad \Delta w = -\eta \nabla J(w)$$

  - The idea of GD is shown in Figure 2 (see next page)
  - See details in `ch10.ipynb`

Figure 2



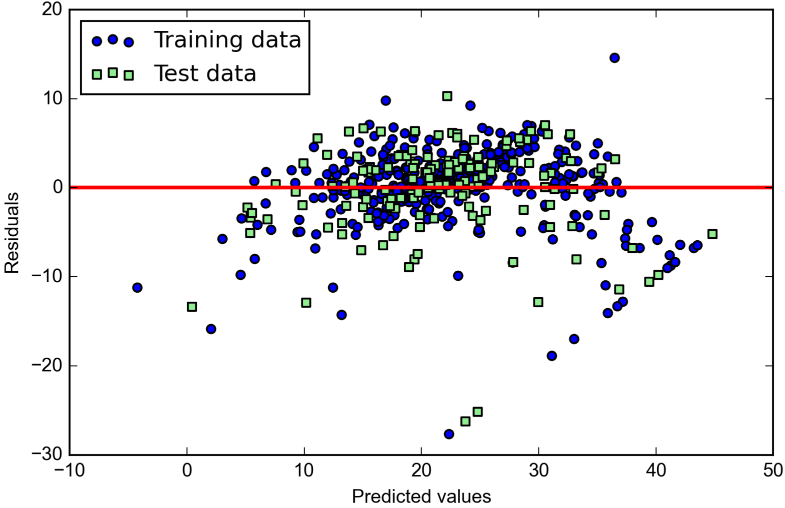
Estimating the parameters via scikit-learn

- scikit-learn's LinearRegression object is an efficient implementation of linear regression model
- See details in `ch10.ipynb`

Evaluating the performance using residual plot

- When our model uses multiple explanatory variables, we cannot visualize the model in a two-dimensional plot
- Instead we can plot the residuals (the difference or vertical distances between the actual and predicted values) versus the predicted values to diagnose our regression model
- One residual plot is shown in Figure 3 (see next page)

Figure 3


Training data
Test data
Residuals
Predicted values
20
10
0
−10
−20
−30
−10
0
10
20
30
40
50

Evaluating the performance using Mean Squared Error

- Another useful quantitative measure of a model's performance is the so-called Mean Squared Error (MSE)

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)} \right)^2$$

- The MSE is useful for comparing different regression models or for tuning their parameters via a grid search and cross-validation
- See details in `ch10.ipynb`