Regression Analysis

Dr. Saerom Park
Statistical Learning and Computational Finance Lab.
Department of Industrial Engineering
psr6275@snu.ac.kr
http://slcf.snu.ac.kr

Table of Contents

Reference

- **Reading:** [Raschka. (2017), chapter 10], [GÉRON. (2017), chapter 4].

Introduction

선형 회귀(Linear Regression)란?

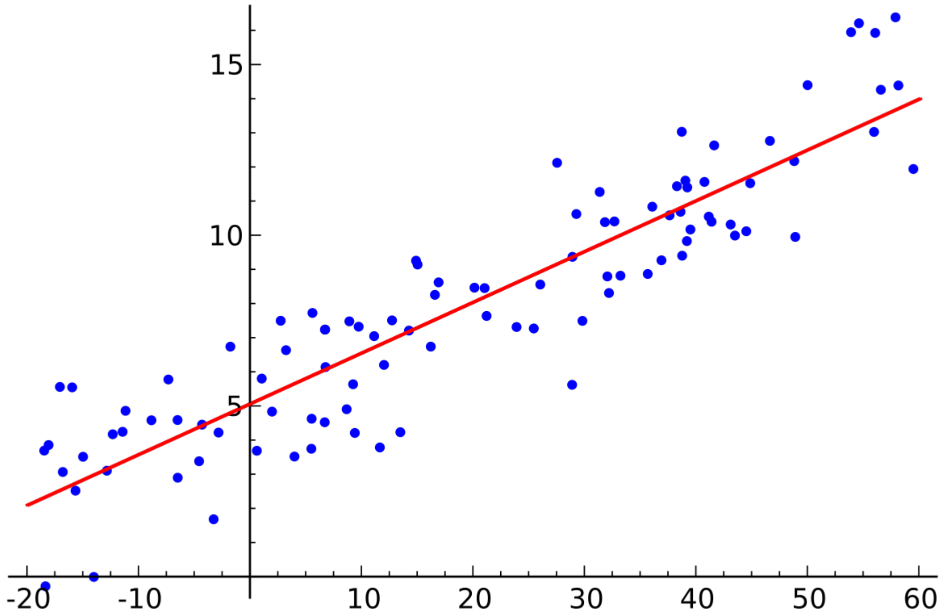
- 가장 간단한 모델 중 하나
- 연속된 변수를 예측하기 위한 모델
- 산업 및 과학 분야에 넓게 응용됨
- 두 변수간의 관계를 분석하거나, 변수를 예측하는 데에 주로 사용된다.
- Simple Linear Regression / Multiple Linear Regression / Polynomial Regression
- 이 외에도 다양한 회귀(Regression)가 존재

LINEAR REGRESSION MODEL

Linear Regression

Linear Regression의 모형과 특성

- 여러 특성이 특정 값과 얼마나 선형적인 관계를 갖는지 파악하는데 사용된다.
  - Correlation coefficient 를 통해 선형 적인 관계를 볼 수 있다.
- Supervised machine learning 중의 하나이다.
- 분류(Classification)과는 다르게 target으로 연속적인 변수를 다룬다.



https://en.wikipedia.org/wiki/Regression_analysis

Linear Regression

Simple Linear Regression

- 하나의 설명변수(Explanatory Variable) x로 예측변수(Target Variable) y를 예측하는 것이 목표이다.

$$y = w_0 + w_1 x$$

w_0 : y축 절편(intercept)
w_1 : 직선의 기울기(slope)

Multiple Linear Regression

- 여러 개의 설명변수 $x_0,\ x_1, \cdots, x_n$로 예측변수 y를 예측하는 것이 목표이다.

$$\hat{y} = w_0 x_0 + w_1 x_1 + \cdots + w_n x_n = \sum_{i=0}^{n} w_i x_i = w^T x$$

w_0 : y축 절편(intercept) with $x_0 = 1$

LEARNING MODEL

Learning Model

Normal Equation

- 데이터 행렬 $\boldsymbol{X} = \begin{bmatrix} - \boldsymbol{x}^{(1)} - \\ \cdots \\ - \boldsymbol{x}^{(n)} - \end{bmatrix} \in \mathbb{R}^{n \times p}$, target $\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ \cdots \\ y^{(n)} \end{bmatrix}$ 및 선형 회귀 모델 $\hat{\boldsymbol{y}} = \boldsymbol{Xw}$ 이 주어졌을 때 다음의 비용함수를 최소화 하는 모델을 구한다.

$$\boldsymbol{min_w}\ (\boldsymbol{Xw} - \boldsymbol{y})^{\boldsymbol{T}} (\boldsymbol{Xw} - \boldsymbol{y})$$

- 위의 목적함수를 최소화 하는 w는 다음과 같다.
  - $\boldsymbol{w} = \left(\boldsymbol{X^T X}\right)^{-1} \boldsymbol{X^T y}$
  - 위의 식은 feature 수 p에대해 $O(p^3)$의 계산 복잡도를 가진다.
- 특성 수가 큰 경우에는 다른 방법으로 파라미터를 구하는 것이 필요
  - Gradient Descent Method

Solving Regression

Regression with Gradient descent(GD)

- 목적함수의 Cost를 줄이기 위해 Gradient Descent(GD), Stochastic Gradient Descent(SGD)를 사용할 수 있음
  - Gradient 방향의 반대 방향으로 조금씩 이동하여 목적함수의 값을 감소시켜 나감

- Regression 문제는 Sum of Squared Error(SSE)를 Cost로 하여 이를 최소화 하는 파라미터를 찾고자 함

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

$$\hat{y} = w^T x$$

Practice

Code for Linear Regression

- Exploring the Housing dataset into a data frame
- Visualizing the important characteristics of a dataset
- Code 01 참고

Solving Regression

Scikit-learn의 LinearRegression class

- sklearn.linear_model.LinearRegression(fit_intercept=True, normalize=False, copy_X=True, n_jobs=1)

Parameters	
fit_intercept	절편(intercept)의 여부. True면 절편 포함하여 회귀
normalize	Standard scaler를 X에다가 적용할 것인가에 대한 여부
copy_X	X를 복사하여 쓸 것인가에 대한 여부. 아니면 X에 덮어쓰기 됨
n_jobs	여러 CPU를 병행하여 사용할 것인가에 대한 여부 (-1이면 모두 사용)

Attributes	
coef_	예측된 계수들을 array로 표현
intecerpt_	절편(intercept)를 array로 표현

- http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

Solving Regression

Regression with scikit-learn

- Scikit-learn에서는 LinearRegression 모델을 제공하고 있으며, 이를 통해 쉽게 regression을 사용할 수 있다.
- 이 때 Normal Equation이 사용되는데, Normal Equation이란 수학적으로 계산된 최적의 가중치를 나타내며, 아래와 같이 계산된다.

$$w = (x^T \cdot x)^{-1} x^T \cdot y$$

RANSAC 알고리즘

- Regression은 이상치(outliers)에 의해 많은 영향을 받는 성질이 있어, 이상치의 부정적인 영향을 제거하기 위한 RANdom Sample Consensus (RANSAC) 알고리즘을 제공하고 있다. RANSAC은 아래와 같은 과정을 거쳐 진행된다.

1. 데이터 중 임의의 개수를 선택하여 inlier로 가정하고 회귀 모델을 구한다.
2. 나머지 데이터를 회귀 모델과 비교하여 사용자 지정 오차 내의 데이터를 inlier로 포함시킨다.
3. Inlier를 토대로 다시 회귀 모델을 구한다.
4. 오차가 사용자 지정 오차 내에 있거나, 반복횟수에 도달하며 종료. 만약 그렇지 않다면 1번으로 돌아가 같은 위의 과정을 반복한다.

EVALUATION

Solving Regression

Evaluating the performance of linear regression

- Residuals(잔차)를 활용한다.
  - 잔차란 실제 값과 예측 값을 차이를 말하며, Residual plot을 통해 확인한다.
  - 잔차가 0을 기준으로 정규 분포와 같이 보이는 것이 선호됨

- Mean Squared Error(MSE)을 활용한다.

  MSE란 오차의 제곱 평균을 말하며, SSE의 평균값을 뜻한다.

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

- Coefficient of determination(R^2)을 활용한다.

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

  - 최대 값은 1, 최소 값은 0을 가지며, 값이 클수록 모델이 학습 데이터를 잘 설명하는 것을 의미함

REGULARIZED LINEAR MODEL

Regularized methods for regression

Regularized methods for regression

- Overfitting을 막기 위해, scikit learn에서는 다음과 같은 regularized linear regression을 제공한다. $\boldsymbol{\lambda}$는 hyper-parameter로써 regularization의 강도를 나타낸다. 높으면 높을수록 가중치의 크기가 클수록 비용이 커진다.

- Ridge Regression

$$J(\boldsymbol{w})_{Ridge} = \sum_{i=1}^{n}\left(\boldsymbol{y}^{(i)} - \hat{\boldsymbol{y}}^{(i)}\right)^2 + \boldsymbol{\lambda}\sum_{j=1}^{m} \boldsymbol{w_j}^2$$

- Least Absolute Shrinkage and Selection Operator (LASSO).

$$J(\boldsymbol{w})_{LASSO} = \sum_{i=1}^{n}\left(\boldsymbol{y}^{(i)} - \hat{\boldsymbol{y}}^{(i)}\right)^2 + \boldsymbol{\lambda}\sum_{j=1}^{m} |\boldsymbol{w_j}|$$

- Elastic Net

$$J(\boldsymbol{w})_{Elastic\ Net} = \sum_{i=1}^{n}\left(\boldsymbol{y}^{(i)} - \hat{\boldsymbol{y}}^{(i)}\right)^2 + \boldsymbol{\lambda_1}\sum_{j=1}^{m} \boldsymbol{w_j}^2 + \boldsymbol{\lambda_2}\sum_{j=1}^{m} |\boldsymbol{w_j}|$$

Practice

Code for Solving Regression

- Solving regression for regression parameters with gradient descent
- Estimating the coefficient of a regression model via scikit-learn
- Fitting a robust regression model using RANSAC
- Evaluating the performance of linear regression models
- Using regularized methods for regression
- Code 02 참고

NONLINEAR REGRESSION MODEL

Non-Linear regression

Polynomial regression

- 앞서 사용한 Linear regression은 선형관계를 파악하는 것에 목표가 있다.
- 하지만, 이는 비선형관계를 파악하는 데에는 적합하지 않다.
- 따라서 다음과 같은 모델을 통해 polynomial regression을 실행한다.

$$\hat{y} = w_0 + w_1 x + w_2 x^2 + \cdots + w_n x^n$$

- 최고 차항의 차수에 따라 1차(Linear), 2차(Quadratic), 3차(Cubic)등의 회귀를 진행할 수 있다.

Regression with Log-transformation

- 다항함수 관계가 아닌 다른 관계가 예측된다면, 데이터를 정리하여 비선형 관계를 나타낼 수 있다.

$$y = 2^x$$
$$log_2(y) = x$$

Practice

Non-linear Regression

- Turning a linear regression model into a curve - polynomial regression
- Modeling nonlinear relationships in the Housing Dataset
- Code 03 참고

Reference

- [Raschka. (2017)] Raschka, Sebastian, and Vahid Mirjalili. *Python machine learning*. Packt Publishing Ltd, 2017.
- [Müller. (2016)] Müller, Andreas C., and Sarah Guido. *Introduction to machine learning with Python: a guide for data scientists*. 2016.
- [GÉRON. (2017)] GÉRON, Aurélien. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. Sebastopol, CA: O, 2017.

Correlation

Pearson Correlation coefficient

- 두 변수간 얼마나 선형관계가 있는지 나타내는 상수

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

σ_{xy} : x와 y의 공분산
σ_x : y의 표준편차
σ_y : x의 표준편차

- r은 -1에서 1까지의 값을 가지며, 1이면 강한 양의 상관관계, 0은 상관관계 없음, -1은 강한 음의 상관관계를 가진다.
- 혹은 다음과 같이 나타낼 수도 있다.

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