수치해석
(2019학년도 1학기)

[10주/1차시 학습내용]: Polynomial Regression (다항 회귀)와 다변수 선형 회귀, 일반 선형 회귀를 이해한다.

국민대학교 소프트웨어학부

Course 스케줄

- Ch 7: Optimization (최적화)
- Ch 14: Linear Regression (선형 회귀)
- Ch 15: Polynomial Regression (Polynomial Regression (다항 회귀)와 다변수 선형 회귀, 일반 선형 회귀)
- Ch 14: Statistics Review (정규분포와 균등분포)
- Ch 16: Fourier Analysis (일반 선형 회귀)
- Ch 17: Polynomial Interpolation (다항식 보간법)
- Ch 19, 20, 21: Numerical Integration and Differentiation (수치 적분과 미분)

Polynomial Regression (다항 회귀)

다항 회귀를 이해한다

https://github.com/SCKIMOSU/Numerical-Analysis/blob/master/regression_polynomial.py

국민대학교 소프트웨어학부

Polynomial Regression (다항 회귀)

- 다항 회귀: x^2, x^3이 있는 곡선 방정식을 사용하여 곡선 접합 수행

15.1 POLYNOMIAL REGRESSION

In Chap.14, a procedure was developed to derive the equation of a straight line using the least-squares criterion. Some data, although exhibiting a marked pattern such as seen in Fig. 15.1, are poorly represented by a straight line. For these cases, a curve would be better suited to fit the data. As discussed in Chap. 14, one method to accomplish this objective is to use transformations. Another alternative is to fit polynomials to the data using *polynomial regression*.

The least-squares procedure can be readily extended to fit the data to a higher-order polynomial. For example, suppose that we fit a second-order polynomial or quadratic:

$$y = a_0 + a_1 x + a_2 x^2 + e \qquad (15.1)$$

Polynomial Regression (다항 회귀)

14.3 LINEAR LEAST-SQUARES REGRESSION

Where substantial error is associated with data, the best curve-fitting strategy is to derive an approximating function that fits the shape or general trend of the data without necessarily matching the individual points. One approach to do this is to visually inspect the plotted data and then sketch a "best" line through the points. Although such "eyeball" approaches have commonsense appeal and are valid for "back-of-the-envelope" calculations, they are deficient because they are arbitrary. That is, unless the points define a perfect straight line (in which case, interpolation would be appropriate), different analysts would draw different lines.

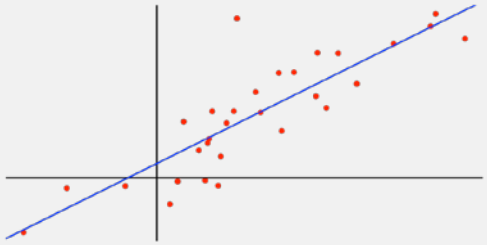
To remove this subjectivity, some criterion must be devised to establish a basis for the fit. One way to do this is to derive a curve that minimizes the discrepancy between the data points and the curve. To do this, we must first quantify the discrepancy. The simplest example is fitting a straight line to a set of paired observations: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. The mathematical expression for the straight line is

$$y = a_0 + a_1 x + e \tag{14.8}$$

where a_0 and a_1 are coefficients representing the intercept and the slope, respectively, and e is the error, or *residual*, between the model and the observations, which can be represented by rearranging Eq. (14.8) as

$$e = y - a_0 - a_1 x \tag{14.9}$$

Thus, the residual is the discrepancy between the true value of y and the approximate value, $a_0 + a_1 x$, predicted by the linear equation.



$$\text{Slope } (b) = \frac{n \times \Sigma xy - (\Sigma x) \times (\Sigma y)}{n \times \Sigma x^2 - (\Sigma x)^2}$$

$$\text{Intercept } (a) = \frac{\Sigma y - b(\Sigma x)}{n}$$

$$\text{Regression } (y) = a + bx$$

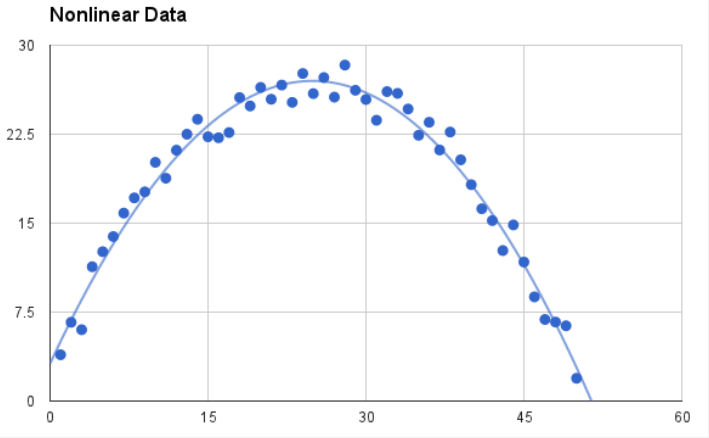
15.1 POLYNOMIAL REGRESSION

In Chap.14, a procedure was developed to derive the equation of a straight line using the least-squares criterion. Some data, although exhibiting a marked pattern such as seen in Fig. 15.1, are poorly represented by a straight line. For these cases, a curve would be better suited to fit the data. As discussed in Chap. 14, one method to accomplish this objective is to use transformations. Another alternative is to fit polynomials to the data using *polynomial regression*.

The least-squares procedure can be readily extended to fit the data to a higher-order polynomial. For example, suppose that we fit a second-order polynomial or quadratic:

$$y = a_0 + a_1 x + a_2 x^2 + e \tag{15.1}$$

361



선형회귀과 다항회귀의 비교

- Linear Regression: x가 있는 직선 방정식을 사용하여 곡선 접합 (Curve Fitting) 수행
- Polynomial Regression: x^2, x^3 이 있는 곡선 방정식을 사용하여 곡선 접합 (Curve Fitting) 수행

14.3 LINEAR LEAST-SQUARES REGRESSION

Where substantial error is associated with data, the best curve-fitting strategy is to derive an approximating function that fits the shape or general trend of the data without necessarily matching the individual points. One approach to do this is to visually inspect the plotted data and then sketch a "best" line through the points. Although such "eyeball" approaches have commonsense appeal and are valid for "back-of-the-envelope" calculations, they are deficient because they are arbitrary. That is, unless the points define a perfect straight line (in which case, interpolation would be appropriate), different analysts would draw different lines.

To remove this subjectivity, some criterion must be devised to establish a basis for the fit. One way to do this is to derive a curve that minimizes the discrepancy between the data points and the curve. To do this, we must first quantify the discrepancy. The simplest example is fitting a straight line to a set of paired observations: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. The mathematical expression for the straight line is

$$y = a_0 + a_1 x + e \tag{14.8}$$

where a_0 and a_1 are coefficients representing the intercept and the slope, respectively, and e is the error, or *residual*, between the model and the observations, which can be represented by rearranging Eq. (14.8) as

$$e = y - a_0 - a_1 x \tag{14.9}$$

Thus, the residual is the discrepancy between the true value of y and the approximate value, $a_0 + a_1 x$, predicted by the linear equation.

$$min \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$$

$$Slope\ (b) = \frac{n \times \Sigma xy - (\Sigma x) \times (\Sigma y)}{n \times \Sigma x^2 - (\Sigma x)^2}$$

$$Intercept\ (a) = \frac{\Sigma y - b(\Sigma x)}{n}$$

$$Regression\ (y) = a + bx$$

15.1 POLYNOMIAL REGRESSION

In Chap.14, a procedure was developed to derive the equation of a straight line using the least-squares criterion. Some data, although exhibiting a marked pattern such as seen in Fig. 15.1, are poorly represented by a straight line. For these cases, a curve would be better suited to fit the data. As discussed in Chap. 14, one method to accomplish this objective is to use transformations. Another alternative is to fit polynomials to the data using *polynomial regression*.

The least-squares procedure can be readily extended to fit the data to a higher-order polynomial. For example, suppose that we fit a second-order polynomial or quadratic:

$$y = a_0 + a_1 x + a_2 x^2 + e \tag{15.1}$$

361

$$min \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 {x_i}^2)^2$$

Polynomial Regression (다항 회귀) 이해하기

- 여러 개의 a_0와 a_1, a_2 값을 구하는 것이 아니라, 한 개의 a_0와 a_1, a_2 값을 구한다. 즉, 단 하나의 이차원 곡선을 구한다.

For this case the sum of the squares of the residuals is

$$S_r = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2)^2 \qquad (15.2)$$

To generate the least-squares fit, we take the derivative of Eq. (15.2) with respect to each of the unknown coefficients of the polynomial, as in

$$\frac{\partial S_r}{\partial a_0} = -2 \sum (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum x_i (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum x_i^2 (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

a_1에 대해 편미분하면, $-x_i$가 상수로 나온다
a_2 에 대해 편미분하면, $-x_i^2$가 상수로 나온다

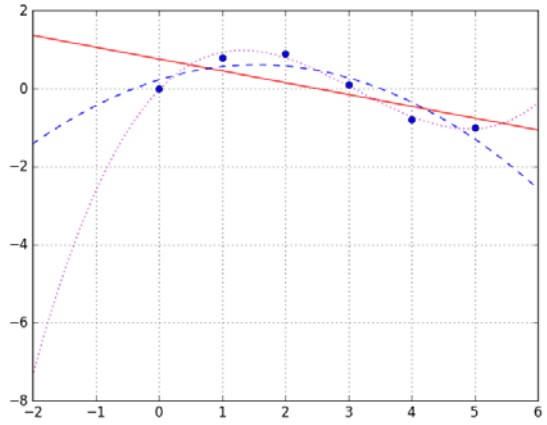
These equations can be set equal to zero and rearranged to develop the following set of normal equations:

$$(n)a_0 + (\sum x_i)a_1 + (\sum x_i^2)a_2 = \sum y_i$$

$$(\sum x_i)a_0 + (\sum x_i^2)a_1 + (\sum x_i^3)a_2 = \sum x_i y_i$$

$$(\sum x_i^2)a_0 + (\sum x_i^3)a_1 + (\sum x_i^4)a_2 = \sum x_i^2 y_i$$

각 방정식을 0으로 두게 된다. a_0와 a_1, a_2를 변수로 두고, 각 변수 앞을 정리한다



Linear Regression과 Polynomial Regression의 비교

- Polynomial Regression (다항 회귀)를 유도하기

Least Squares Regression

$$S = min \sum_{i=1}^{n} e_i^2 = min \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$$ Least Square Form

Minimize the sum of the squares of the errors→ optimization
Find min, max → differentiation

Calculate slope (a_1) and intercept (a_0) in order to minimize the sum of error
Multi variables (a_0, a_1) → Partial Derivative

$$\frac{\partial S}{\partial a_0} = \frac{\partial}{\partial a_0} \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2 = 0$$

$$\frac{\partial S}{\partial a_1} = \frac{\partial}{\partial a_1} \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2 = 0$$

$$n\left(\sum x_i\right) a_0 + n\left(\sum x_i^2\right) a_1 = n \sum x_i y_i - ③$$

$$n\left(\sum x_i\right) a_0 + \left(\sum x_i\right)^2 a_1 = \sum x_i \sum y_i - ④$$

$$③ - ④ \quad \left(n\left(\sum x_i^2\right) - \left(\sum x_i\right)^2\right) a_1 = n \sum x_i y_i - \sum x_i \sum y_i$$

$$a_1 = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n(\sum x_i^2) - (\sum x_i)^2} \qquad a_0 = \frac{\sum y_i}{n} - \frac{\sum x_i}{n} a_1$$

선형 회귀(linear Regression)의 전개 과정

$$min \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2)^2$$

For this case the sum of the squares of the residuals is

$$S_r = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2)^2 \qquad (15.2)$$

To generate the least-squares fit, we take the derivative of Eq. (15.2) with respect to each of the unknown coefficients of the polynomial, as in

$$\frac{\partial S_r}{\partial a_0} = -2 \sum (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum x_i (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum x_i^2 (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

다항 회귀 (Polynomial Regression)의 전개 과정

These equations can be set equal to zero and rearranged to develop the following set of normal equations:

$$(n) a_0 + (\sum x_i) a_1 + (\sum x_i^2) a_2 = \sum y_i$$
$$(\sum x_i) a_0 + (\sum x_i^2) a_1 + (\sum x_i^3) a_2 = \sum x_i y_i$$
$$(\sum x_i^2) a_0 + (\sum x_i^3) a_1 + (\sum x_i^4) a_2 = \sum x_i^2 y_i$$

행렬 전개

- 다항 회귀(Polynomial Regression)의 전개 과정에서 필요한 행렬 전개

For this case the sum of the squares of the residuals is

$$S_r = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2)^2 \qquad (15.2)$$

To generate the least-squares fit, we take the derivative of Eq. (15.2) with respect to each of the unknown coefficients of the polynomial, as in

$$\frac{\partial S_r}{\partial a_0} = -2 \sum (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum x_i (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum x_i^2 (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

These equations can be set equal to zero and rearranged to develop the following set of normal equations:

$$(n)a_0 + (\sum x_i)\, a_1 + (\sum x_i^2)\, a_2 = \sum y_i$$
$$(\sum x_i)\, a_0 + (\sum x_i^2)\, a_1 + (\sum x_i^3)\, a_2 = \sum x_i y_i$$
$$(\sum x_i^2)\, a_0 + (\sum x_i^3)\, a_1 + (\sum x_i^4)\, a_2 = \sum x_i^2 y_i$$

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{bmatrix}$$

Partial Derivative (편미분)

- 편미분으로 Polynomial Regression (다항 회귀) 유도하기

$$\frac{\partial S}{\partial \boldsymbol{a_0}} = \frac{\partial}{\partial \boldsymbol{a_0}} \sum_{i=1}^{n} (y_i - \boldsymbol{a_0} - a_1 x_i - a_2 {x_i}^2)^2 = 0$$

$$y = \{f(x)\}^n$$

$$y' = n \cdot \{f(x)\}^{n-1} \cdot f'(x)$$

$$= 2 \cdot \sum_{i=1}^{n} (y_i - \boldsymbol{a_0} - a_1 x_i - a_2 {x_i}^2)^{2-1} \cdot \frac{\partial (y_i - \boldsymbol{a_0} - a_1 x_i - a_2 {x_i}^2)}{\partial \boldsymbol{a_0}}$$

$$= 2 \cdot \sum_{i=1}^{n} (y_i - \boldsymbol{a_0} - a_1 x_i - a_2 {x_i}^2) \cdot (-\boldsymbol{1})$$

$$= -\boldsymbol{2} \cdot \sum_{i=1}^{n} (y_i - \boldsymbol{a_0} - a_1 x_i - a_2 {x_i}^2) = \boldsymbol{0}$$

$$= \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} a_0 - \sum_{i=1}^{n} a_1 x_i - \sum_{i=1}^{n} a_2 {x_i}^2 = 0$$

$\boldsymbol{a_0}$ 에 대해 편미분해서 $\frac{\partial S}{\partial \boldsymbol{a_0}} = \boldsymbol{0}$이 되는 $\boldsymbol{a_0}$ 값을 구해야 한다

Polynomial Regression (다항 회귀)

- 편미분으로 Polynomial Regression (다항 회귀) 유도하기

$$\sum_{i=1}^{n} y_i - \sum_{i=1}^{n} a_0 - \sum_{i=1}^{n} a_1 x_i - \sum_{i=1}^{n} a_2 {x_i}^2 = 0$$

계속 전개 하면

$$\sum_{i=1}^{n} a_0 + \sum_{i=1}^{n} a_1 x_i + \sum_{i=1}^{n} a_2 {x_i}^2 = \sum_{i=1}^{n} y_i$$

$$n \cdot a_0 + \sum_{i=1}^{n} a_1 x_i + \sum_{i=1}^{n} a_2 {x_i}^2 = \sum_{i=1}^{n} y_i$$

$$\sum_{i=1}^{n} a_0 = a_0 \cdot \sum_{i=1}^{n} 1 = a_0 \cdot (1 + 1 + \cdots + 1) = a_0 \cdot n = n \cdot a_0$$

$$\boldsymbol{n a_0 + \left(\sum x_i\right) a_1 + \left(\sum {x_i}^2\right) a_2 = \sum y_i} \quad ①$$

a_0 에 대해 ① 번 식이 유도 된다

Polynomial Regression (다항 회귀)

- 편미분으로 Polynomial Regression (다항 회귀) 유도하기

$$\frac{\partial S}{\partial \boldsymbol{a_1}} = \frac{\partial}{\partial a_1}\sum_{i=1}^{n}(y_i - a_0 - \boldsymbol{a_1}x_i - a_2{x_i}^2)^2 = 0$$

$$y = \{f(x)\}^n$$

$$y' = n \cdot \{f(x)\}^{n-1} \cdot f'(x)$$

$$= 2 \cdot \sum_{i=1}^{n}(y_i - a_0 - \boldsymbol{a_1}x_i - a_2{x_i}^2)^{2-1} \cdot \frac{\partial(y_i - a_0 - \boldsymbol{a_1}x_i - a_2{x_i}^2)}{\partial \boldsymbol{a_1}}$$

$$= -2\boldsymbol{x_i} \cdot \sum_{i=1}^{n}(y_i - a_0 - \boldsymbol{a_1}x_i - a_2{x_i}^2) = 0$$

$$= \boldsymbol{x_i} \cdot \sum_{i=1}^{n} y_i - \boldsymbol{x_i} \cdot \sum_{i=1}^{n} a_0 - \boldsymbol{x_i} \cdot \sum_{i=1}^{n} a_1 x_i - \boldsymbol{x_i} \cdot \sum_{i=1}^{n} a_2 {x_i}^2 = 0$$

$$= \sum_{i=1}^{n} \boldsymbol{x_i} \cdot y_i - \sum_{i=1}^{n} \boldsymbol{x_i} \cdot a_0 - \sum_{i=1}^{n} a_1 \boldsymbol{x_i} \cdot x_i - \sum_{i=1}^{n} a_2 \boldsymbol{x_i} \cdot {x_i}^2 = 0$$

$\boldsymbol{a_1}$ 에 대해 편미분해서 $\frac{\partial S}{\partial \boldsymbol{a_1}} = \boldsymbol{0}$이 되는 $\boldsymbol{a_1}$ 값을 구해야 한다

$\boldsymbol{x_i}$ 를 $\sum_{i=1}^{n}$ 안에 넣어서 계산해서 전개한다

Polynomial Regression (다항 회귀)

$$\sum_{i=1}^{n} \boldsymbol{x_i} \cdot y_i - \sum_{i=1}^{n} \boldsymbol{x_i} \cdot a_0 - \sum_{i=1}^{n} a_1 \boldsymbol{x_i} \cdot x_i - \sum_{i=1}^{n} a_2 \boldsymbol{x_i} \cdot {x_i}^2 = 0$$

전개한다

$$\sum_{i=1}^{n} \boldsymbol{x_i} \cdot y_i - \sum_{i=1}^{n} a_0 \boldsymbol{x_i} - \sum_{i=1}^{n} a_1 {x_i}^2 - \sum_{i=1}^{n} a_2 {x_i}^3 = 0$$

$$\sum_{i=1}^{n} \boldsymbol{a_0} x_i + \sum_{i=1}^{n} a_1 {x_i}^2 + \sum_{i=1}^{n} a_2 {x_i}^3 = \sum_{i=1}^{n} \boldsymbol{x_i} \cdot y_i$$

$$\left(\sum \boldsymbol{x_i}\right) \boldsymbol{a_0} + \left(\sum \boldsymbol{x_i^2}\right) \boldsymbol{a_1} + \left(\sum \boldsymbol{x_i^3}\right) \boldsymbol{a_2} = \sum \boldsymbol{x_i} \cdot y_i \quad ②$$

$\boldsymbol{a_1}$ 에 대해 ② 번 식이 유도된다

Polynomial Regression (다항 회귀)

- 지금까지 나온 ① 식과 ② 식을 같이 적어 보면 다음과 같다

$$n a_0 + \left(\sum x_i\right) a_1 + \left(\sum {x_i}^2\right) a_2 = \sum y_i \quad ①$$

$$\left(\sum x_i\right) a_0 + \left(\sum {x_i}^2\right) a_1 + \left(\sum {x_i}^3\right) a_2 = \sum x_i \cdot y_i \quad ②$$

Polynomial Regression (다항 회귀)

- 편미분으로 Polynomial Regression (다항 회귀) 유도하기

$$\frac{\partial S}{\partial \boldsymbol{a_2}} = \frac{\partial}{\partial a_2}\sum_{i=1}^{n}(y_i - a_0 - a_1x_i - \boldsymbol{a_2}{x_i}^2)^2 = 0$$

$$y = \{f(x)\}^n$$

$$y' = n \cdot \{f(x)\}^{n-1} \cdot f'(x)$$

$$= 2 \cdot \sum_{i=1}^{n}(y_i - a_0 - a_1x_i - \boldsymbol{a_2}{x_i}^2)^{2-1} \cdot \frac{\partial(y_i - a_0 - a_1x_i - \boldsymbol{a_2}{x_i}^2)}{\partial \boldsymbol{a_2}}$$

$$= -2\boldsymbol{x_i}^2 \cdot \sum_{i=1}^{n}(y_i - a_0 - a_1x_i - \boldsymbol{a_2}{x_i}^2) = 0$$

$$= \boldsymbol{x_i}^2 \cdot \sum_{i=1}^{n} y_i - \boldsymbol{x_i}^2 \cdot \sum_{i=1}^{n} a_0 - \boldsymbol{x_i}^2 \cdot \sum_{i=1}^{n} a_1x_i - \boldsymbol{x_i}^2 \cdot \sum_{i=1}^{n} a_2{x_i}^2 = 0$$

$$= \sum_{i=1}^{n} \boldsymbol{x_i}^2 \cdot y_i - \sum_{i=1}^{n} \boldsymbol{x_i}^2 \cdot a_0 - \sum_{i=1}^{n} a_1\boldsymbol{x_i}^2 \cdot x_i - \sum_{i=1}^{n} a_2\boldsymbol{x_i}^2 \cdot {x_i}^2 = 0$$

$\boldsymbol{a_2}$ 에 대해 편미분해서 $\frac{\partial S}{\partial a_2} = \boldsymbol{0}$이 되는 $\boldsymbol{a_2}$ 값을 구해야 한다

$\boldsymbol{x_i}^2$ 을 $\sum_{i=1}^{n}$ 안에 넣어서 계산해서 전개한다

Polynomial Regression (다항 회귀)

$$\sum_{i=1}^{n} \boldsymbol{x_i^2} \cdot y_i - \sum_{i=1}^{n} \boldsymbol{x_i^2} \cdot a_0 - \sum_{i=1}^{n} a_1 \boldsymbol{x_i^2} \cdot x_i - \sum_{i=1}^{n} a_2 \boldsymbol{x_i^2} \cdot x_i^2 = 0$$

$$\sum_{i=1}^{n} \boldsymbol{x_i^2} \cdot y_i - \sum_{i=1}^{n} \boldsymbol{a_0 x_i^2} - \sum_{i=1}^{n} a_1 \boldsymbol{x_i^3} - \sum_{i=1}^{n} a_2 x_i^4 = 0$$

전개한다

$$\sum_{i=1}^{n} \boldsymbol{a_0 x_i^2} + \sum_{i=1}^{n} a_1 x_i^3 + \sum_{i=1}^{n} a_2 x_i^4 = \sum_{i=1}^{n} x_i^2 \cdot y_i$$

$$\left(\sum \boldsymbol{x_i^2}\right) \boldsymbol{a_0} + \left(\sum \boldsymbol{x_i^3}\right) \boldsymbol{a_1} + \left(\sum \boldsymbol{x_i^4}\right) \boldsymbol{a_2} = \sum_{i=1}^{n} x_i^2 \cdot y_i \quad ③$$

$\boldsymbol{a_2}$ 에 대해 ③번 식이 유도된다

Polynomial Regression (다항 회귀)

- 미지의 수 a_0, a_1, a_2에 대하여, 세 개의 방정식 ①, ②, ③ 을 $\frac{\partial S}{\partial a_0} = 0, \frac{\partial S}{\partial a_1} = 0, \frac{\partial S}{\partial a_2} = 0$ 으로 구했다.
- 여기서 $S = \sum_{i=1}^{n}(y_i - a_0 - a_1 x_i - a_2 {x_i}^2)^2$이다. 모든 실제 값과 모델 값의 에러의 합이다.

$$n a_0 + \left(\sum x_i\right) a_1 + \left(\sum {x_i}^2\right) a_2 = \sum y_i \quad ①$$

$$\left(\sum x_i\right) a_0 + \left(\sum {x_i}^2\right) a_1 + \left(\sum {x_i}^3\right) a_2 = \sum x_i \cdot y_i \quad ②$$

$$\left(\sum {x_i}^2\right) a_0 + \left(\sum {x_i}^3\right) a_1 + \left(\sum {x_i}^4\right) a_2 = \sum {x_i}^2 \cdot y_i \quad ③$$

For this case the sum of the squares of the residuals is

$$S_r = \sum_{i=1}^{n} \left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right)^2 \qquad (15.2)$$

To generate the least-squares fit, we take the derivative of Eq. (15.2) with respect to each of the unknown coefficients of the polynomial, as in

$$\frac{\partial S_r}{\partial a_0} = -2 \sum \left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum x_i \left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right)$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum x_i^2 \left(y_i - a_0 - a_1 x_i - a_2 x_i^2\right)$$

These equations can be set equal to zero and rearranged to develop the following set of normal equations:

$$(n) a_0 + \left(\sum x_i\right) a_1 + \left(\sum x_i^2\right) a_2 = \sum y_i$$

$$\left(\sum x_i\right) a_0 + \left(\sum x_i^2\right) a_1 + \left(\sum x_i^3\right) a_2 = \sum x_i y_i$$

$$\left(\sum x_i^2\right) a_0 + \left(\sum x_i^3\right) a_1 + \left(\sum x_i^4\right) a_2 = \sum x_i^2 y_i$$

행렬 전개를 통한 해 구하기

- 계수(n, $(\sum x_i)$, $(\sum {x_i}^2)$) 를 각 ① , ② , ③ 식에 곱해서 통합 정리해서 구하는 기존의 방법은 해 (a_0, a_1, a_2) 를 구하기가 굉장히 어려움 → **행렬 전개를 통한 해를 구함**

$$n a_0 + \left(\sum x_i\right) a_1 + \left(\sum {x_i}^2\right) a_2 = \sum y_i \quad ①$$

$$\left(\sum x_i\right) a_0 + \left(\sum {x_i}^2\right) a_1 + \left(\sum {x_i}^3\right) a_2 = \sum_{i=1}^{n} x_i \cdot y_i \quad ②$$

$$\left(\sum {x_i}^2\right) a_0 + \left(\sum {x_i}^3\right) a_1 + \left(\sum {x_i}^4\right) a_2 = \sum_{i=1}^{n} {x_i}^2 \cdot y_i \quad ③$$

$$\begin{bmatrix} n & \sum x_i & \sum {x_i}^2 \\ \sum x_i & \sum {x_i}^2 & \sum {x_i}^3 \\ \sum {x_i}^2 & \sum {x_i}^3 & \sum {x_i}^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum {x_i}^2 y_i \end{bmatrix}$$

$$n a_0 + \left(\sum x_i\right) a_1 = \sum y_i - ①$$

$$\left(\sum x_i\right) a_0 + \left(\sum {x_i}^2\right) a_1 = \sum x_i y_i - ②$$

$$② \times n - ① \times \left(\sum x_i\right)$$

$$③ - n\left(\sum x_i\right) a_0 + n\left(\sum {x_i}^2\right) a_1 = n \sum x_i y_i$$

$$④ - n\left(\sum x_i\right) a_0 + \left(\sum x_i\right)\left(\sum x_i\right) a_1 = \sum x_i \sum y_i$$

행렬전개를 이용함

기존의 방법은 적용하기 어려움

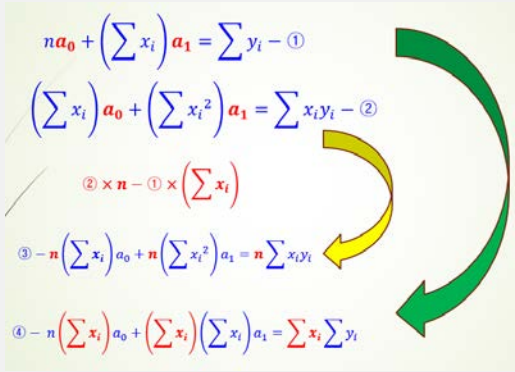
기존의 3차연립방정식의 해 방법

- 기존의 3차연립방정식의 해 풀이 방법은 풀기가 어렵다

$$n a_0 + \left(\sum x_i\right) a_1 + \left(\sum x_i^2\right) a_2 = \sum y_i \quad ①$$

$$\left(\sum x_i\right) a_0 + \left(\sum x_i^2\right) a_1 + \left(\sum x_i^3\right) a_2 = \sum x_i \cdot y_i \quad ②$$

$$\left(\sum x_i^2\right) a_0 + \left(\sum x_i^3\right) a_1 + \left(\sum x_i^4\right) a_2 = \sum x_i^2 \cdot y_i \quad ③$$



$$a_0 = \frac{1}{Z}\left(-\left(\Sigma x^3 \Sigma x - (\Sigma x^2)^2\right)\Sigma x^2 y + \left(\Sigma x^4 \Sigma x - \Sigma x^3 \Sigma x^2\right)\Sigma x y - \left(\Sigma x^4 \Sigma x^2 - (\Sigma x^3)^2\right)\Sigma y\right)$$

$$a_1 = \frac{1}{Z}\left(\left(n\Sigma x^3 - \Sigma x^2 \Sigma x\right)\Sigma x^2 y - \left(n\Sigma x^4 - (\Sigma x^2)^2\right)\Sigma x y + \left(\Sigma x^4 \Sigma x - \Sigma x^3 \Sigma x^2\right)\Sigma y\right)$$

$$a_2 = \frac{1}{Z}\left(-\left(n\Sigma x^2 - (\Sigma x)^2\right)\Sigma x^2 y + \left(n\Sigma x^3 - \Sigma x^2 \Sigma x\right)\Sigma x y - \left(\Sigma x^3 \Sigma x - (\Sigma x^2)^2\right)\Sigma y\right)$$

where:

$$Z = n(\Sigma x^3)^2 - 2\Sigma x^3 \Sigma x^2 \Sigma x + (\Sigma x^2)^3 - \left(n\Sigma x^2 - (\Sigma x)^2\right)\Sigma x^4$$

다항회귀: 행렬 전개를 통한 해 구하기

다항회귀: 행렬 전개를 통한 해 구하기

국민대학교 소프트웨어학부

행렬 전개 : np.linalg.solve() 메소드

EXAMPLE 15.1 Polynomial Regression

Problem Statement. Fit a second-order polynomial to the data in the first two columns of Table 15.1.

TABLE 15.1 Computations for an error analysis of the quadratic least-squares fit.

x_i	y_i	$(y_i - \bar{y})^2$	$(y_i - a_0 - a_1 x_i - a_2 x_i^2)^2$
0	2.1	544.44	0.14332
1	7.7	314.47	1.00286
2	13.6	140.03	1.08160
3	27.2	3.12	0.80487
4	40.9	239.22	0.61959
5	61.1	1272.11	0.09434
$\sum$	152.6	2513.39	3.74657

Solution. The following can be computed from the data:

$$m = 2 \qquad \sum x_i = 15 \qquad \sum x_i^4 = 979$$

$$n = 6 \qquad \sum y_i = 152.6 \qquad \sum x_i y_i = 585.6$$

$$\bar{x} = 2.5 \qquad \sum x_i^2 = 55 \qquad \sum x_i^2 y_i = 2488.8$$

$$\bar{y} = 25.433 \qquad \sum x_i^3 = 225$$

행렬 전개 : np.linalg.solve() 메소드

- Polynomial Regression로 Curve Fitting 해보기

Therefore, the simultaneous linear equations are

$$\begin{bmatrix} 6 & 15 & 55 \\ 15 & 55 & 225 \\ 55 & 225 & 979 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} 152.6 \\ 585.6 \\ 2488.8 \end{Bmatrix}$$

These equations can be solved to evaluate the coefficients. For example, using MATLAB:

```
>> N = [6 15 55;15 55 225;55 225 979];
>> r = [152.6 585.6 2488.8];
>> a = N\r

a =
    2.4786
    2.3593
    1.8607
```

Therefore, the least-squares quadratic equation for this case is

$$y = 2.4786 + 2.3593x + 1.8607x^2$$

행렬 전개 : np.linalg.solve() 메소드

- Polynomial Regression 의 행렬 전개로 Curve Fitting 해보기

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \end{bmatrix}$$

```
# 편미분을 이용한 다항 회귀
a=np.array([[np.size(x), np.sum(x), np.sum(x**2)],
[np.sum(x), np.sum(x**2), np.sum(x**3) ],
[np.sum(x**2), np.sum(x**3), np.sum(x**4) ] ])
#array([[  6,  15,  55],
#       [ 15,  55, 225],
#       [ 55, 225, 979]])
b=np.array([np.sum(y), np.sum(x*y),
np.sum(x**2*y)])
# array([ 152.6,  585.6, 2488.8])
t=np.linalg.solve(a, b)
# array([2.47857143, 2.35928571, 1.86071429])

x1=np.linspace(0,5,10)
y1=t[2]*x1**2+t[1]*x1+t[0]
# derived from ploynomial regression
plt.figure(2)
plt.plot(x,y,'ro', x1, y1, 'b*:')
plt.legend(['Real Data','Polynomial Regression by
Partial Derivative'])
plt.grid()
```

np.linalg.solve() 메소드

- **np.linalg.solve() 메소드는 행렬의 해를 구해주는 메소드이다.**
- ***Solve a linear matrix equation, or system of linear scalar equations.***

$$y_1 = 1.8607{x_1}^2 + 2.3593x_1 + 2.4786$$

```
# 편미분을 이용한 다항 회귀
a=np.array([[np.size(x), np.sum(x), np.sum(x**2)], [np.sum(x),
np.sum(x**2), np.sum(x**3) ], [np.sum(x**2), np.sum(x**3), np.sum(x**4)
] ])
#array([[  6,   15,   55],
#       [ 15,   55, 225],
#       [ 55, 225, 979]])
b=np.array([np.sum(y), np.sum(x*y), np.sum(x**2*y)])
# array([ 152.6,  585.6, 2488.8])
t=np.linalg.solve(a, b)
# array([2.47857143, 2.35928571, 1.86071429])
```

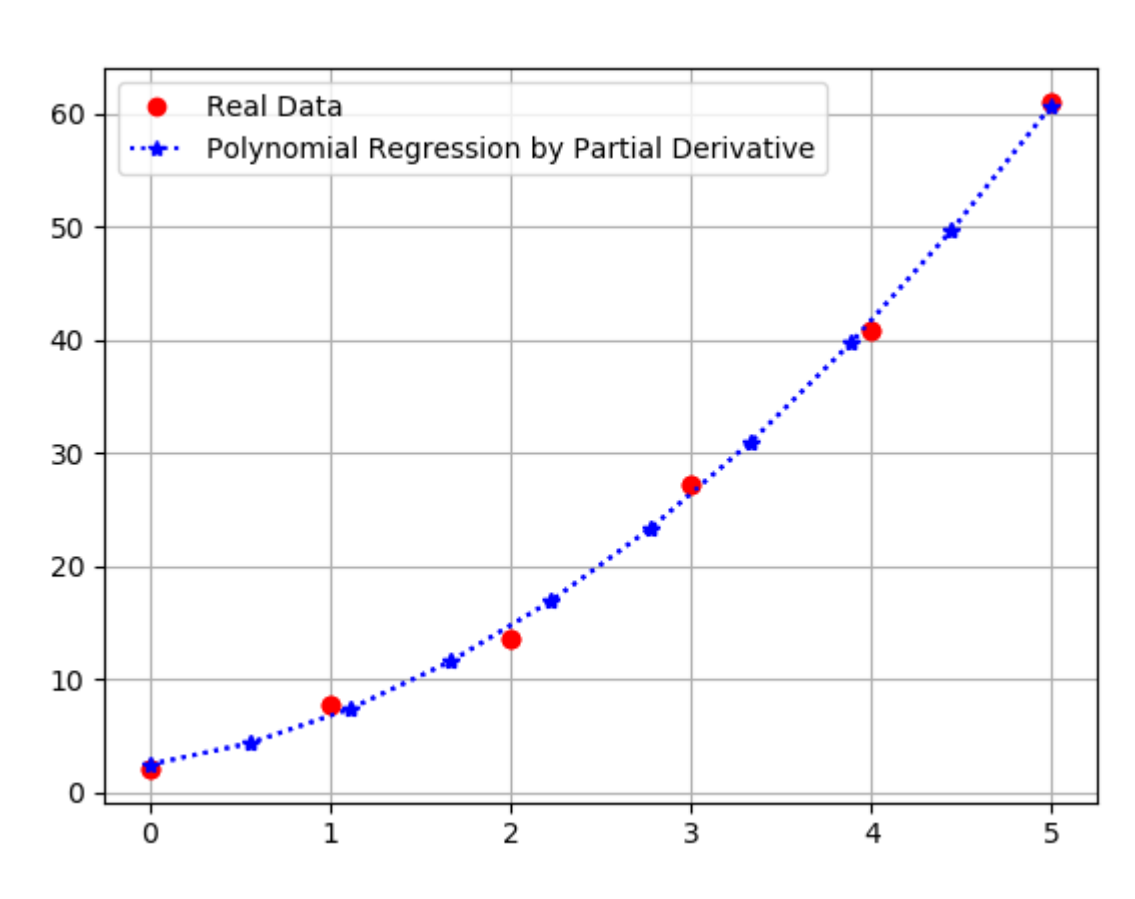
행렬 전개를 통한 해 구하기

- Python Code for Polynomial Regression
- Polynomial Regression로 Curve Fitting 해보기

```
x1=np.linspace(0,5,10)
y1=t[2]*x1**2+t[1]*x1+t[0]
# derived from ploynomial regression
plt.figure(2)
plt.plot(x,y,'ro', x1, y1, 'b*:')
plt.legend(['Real Data','Polynomial Regression by Partial Derivative'])
plt.grid()
```

다항회귀: 행렬 전개를 통한 해 구하기

- 편미분으로 행렬 전개를 통해 다항회귀 해 구하여 보기



https://github.com/SCKIMOSU/Numerical-Analysis/blob/master/regression_polynomial.py

다항회귀: np.polyfit(x,y,2)로 Curve Fitting 해보기

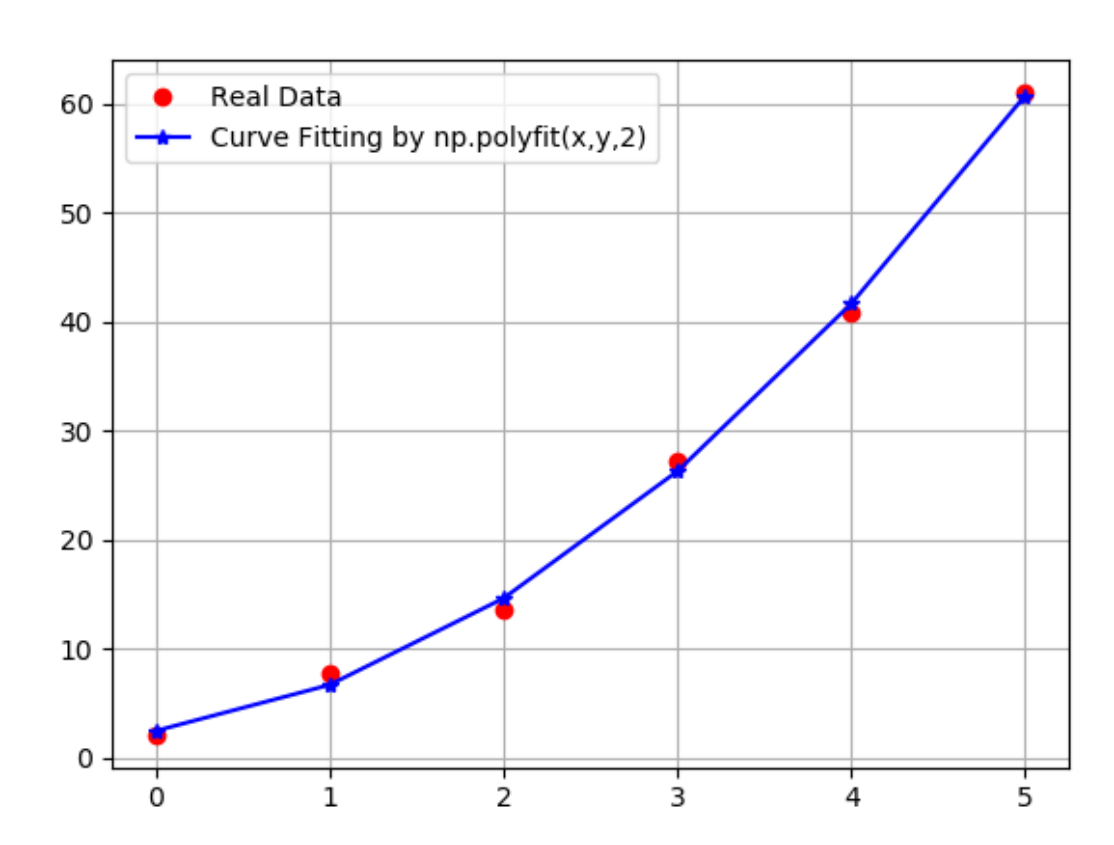
- 다항회귀: np.polyfit(x,y,2)로 Curve Fitting 해보기
- np.polyfit(x,y,2)로 Curve Fitting 해보기

```
import numpy as np
import matplotlib.pyplot as plt

# np.polyfit(x,y,2)를 이용한 다항 회귀
x=np.array([0,1,2,3,4,5])
y=np.array([2.1, 7.7, 13.6, 27.2, 40.9, 61.1])

p2=np.polyfit(x,y,2)
# array([1.86071429, 2.35928571, 2.47857143])
plt.figure(1)
plt.plot(x, y, 'ro', np.polyval(p2, x), 'b*-' )  # x,
p2[0]*x**2+p2[1]*x+p2[2]
plt.legend(['Real Data', 'Polynomial Regression by np.polyfit(x,y,2)'])
plt.grid()
```

https://github.com/SCKIMOSU/Numerical-Analysis/blob/master/regression_polynomial.py

다항회귀: np.polyfit(x,y,2)로 Curve Fitting 해보기

- 다항회귀: np.polyfit(x,y,2)로 Curve Fitting 해보기
- np.polyfit(x,y,2)로 Curve Fitting 해보기



다항회귀: 편미분으로 Curve Fitting 해보기

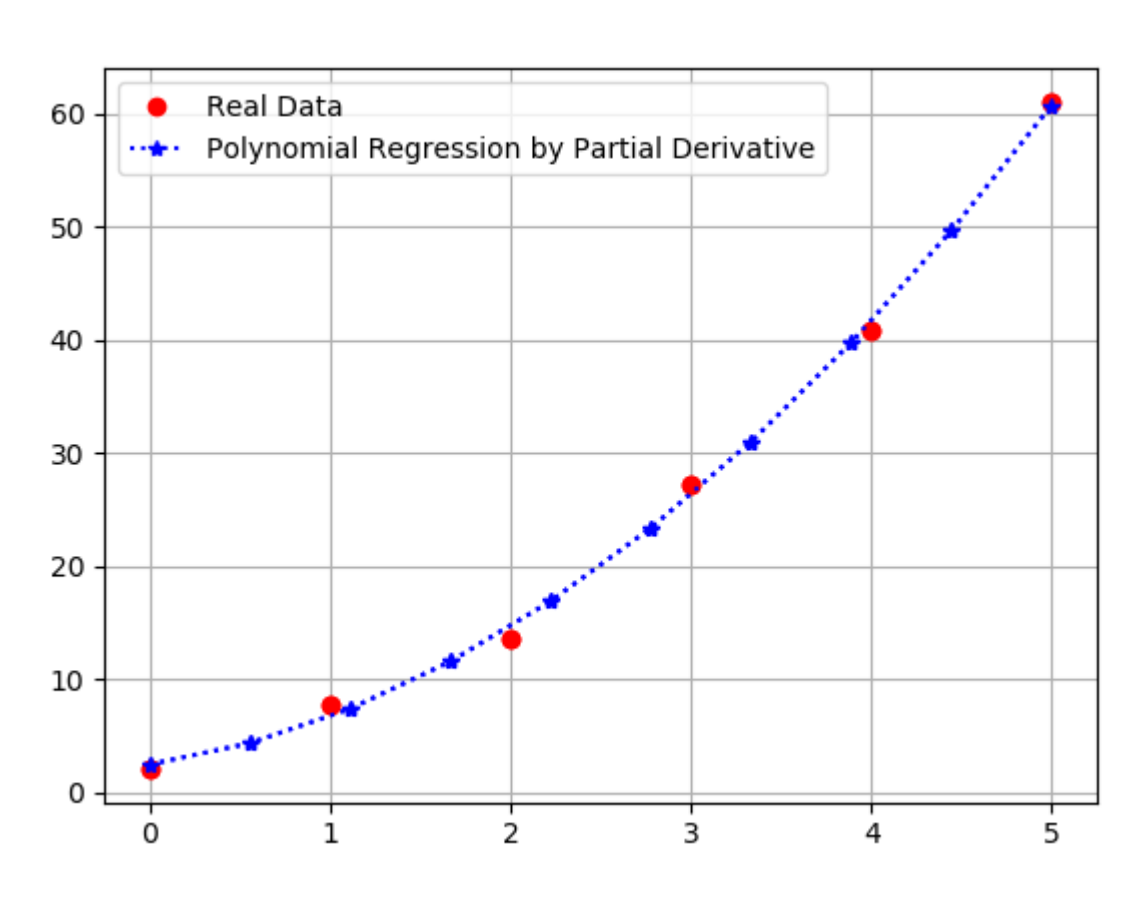
- 편미분으로 다항회귀 해보기

```
# 편미분을 이용한 다항 회귀
a=np.array([[np.size(x), np.sum(x), np.sum(x**2)], [np.sum(x), np.sum(x**2),
np.sum(x**3) ], [np.sum(x**2), np.sum(x**3), np.sum(x**4) ] ])
#array([[  6,  15,  55],
#       [ 15,  55, 225],
#       [ 55, 225, 979]])
b=np.array([np.sum(y), np.sum(x*y), np.sum(x**2*y)])
# array([ 152.6,  585.6, 2488.8])
t=np.linalg.solve(a, b)
# array([2.47857143, 2.35928571, 1.86071429])

x1=np.linspace(0,5,10)
y1=t[2]*x1**2+t[1]*x1+t[0]
# derived from ploynomial regression
plt.figure(2)
plt.plot(x,y,'ro', x1, y1, 'b*:')
plt.legend(['Real Data','Polynomial Regression by Partial Derivative'])
plt.grid()
```

https://github.com/SCKIMOSU/Numerical-Analysis/blob/master/regression_polynomial.py

다항회귀: 편미분으로 Curve Fitting 해보기

- 편미분으로 다항회귀 해보기



https://github.com/SCKIMOSU/Numerical-Analysis/blob/master/regression_polynomial.py

np.polyfit()메소드의 다항 회귀

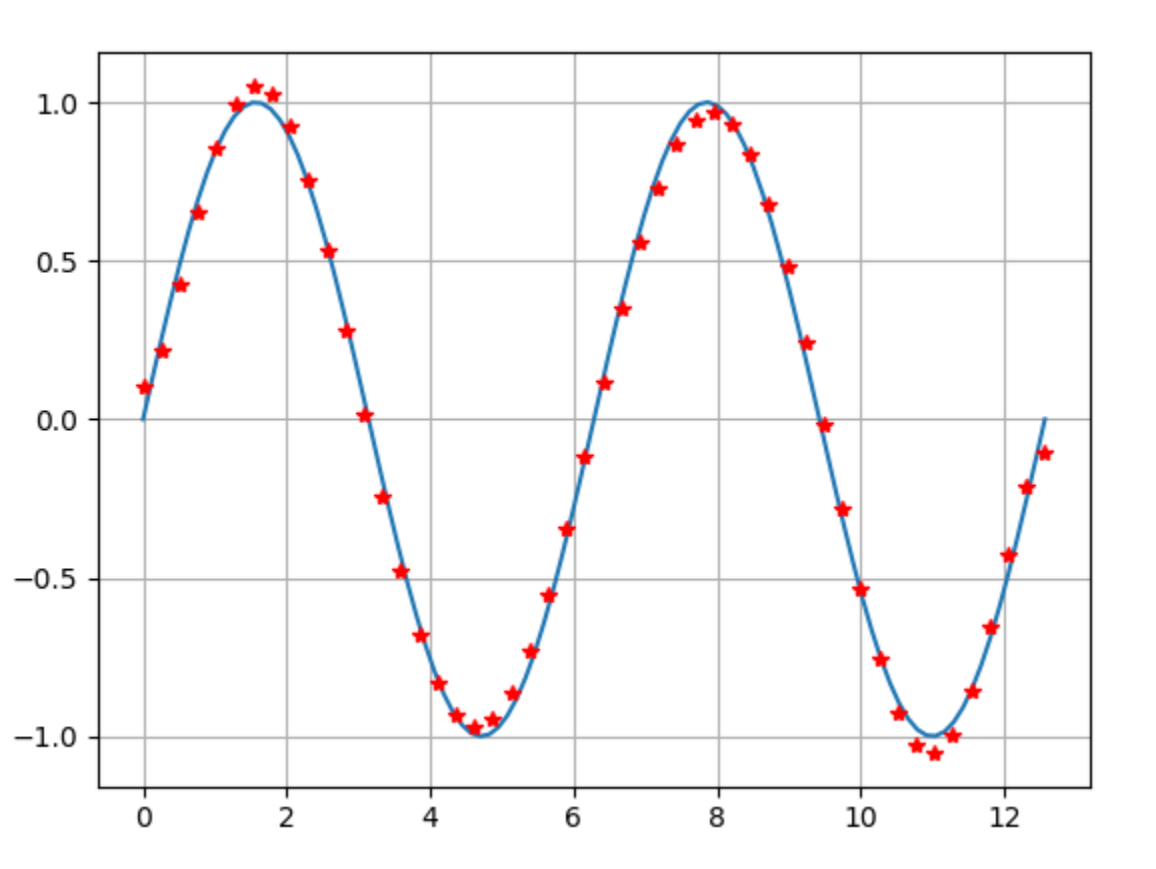
- np.polyfit(x,y,7) 에 도전해보자 (7항)

```
import numpy as np
import matplotlib.pyplot as plt

# 구간 [0,4*pi]에 사인 곡선을 따라 균일한 간격의 점 100개를 생성합니다.
x = np.linspace(0,4*np.pi,100)
y=np.sin(x)

# 구간 [0,4*pi]에 따라 사인 곡선을 시각화한다
plt.figure(1)
plt.plot(x, y)
# np.polyfit을 사용하여 이들 점에 7차 다항식을 피팅합니다.
p = np.polyfit(x,y,7)
# 좀 더 촘촘한 그리드에서 다항식을 계산하고 결과를 플로팅합니다.
x1 = np.linspace(0, 4*np.pi)
y1 = np.polyval(p,x1)
plt.figure(2)
plt.plot(x, y, x1, y1, 'r*')
plt.grid()
```

https://github.com/SCKIMOSU/Numerical-Analysis/blob/master/regression_muti_polyfit.py

np.polyfit()메소드의 다항 회귀

- np.polyfit(x,y,7) 에 도전해보자 (7항)



다변수 선형 회귀

텐서플로우를 통해 다변수 선형 회귀를 이해한다

https://github.com/SCKIMOSU/Numerical-Analysis/blob/master/regression_multiple.py

국민대학교 소프트웨어학부

Multiple Linear Regression (다중 선형 회귀 분석)

- 두 개 변수(x, y 또는 x_1, x_2)의 1차 방정식을 사용하여 곡선 접합 수행
- 또는 다변수 선형회귀라고도 한다

15.2 MULTIPLE LINEAR REGRESSION

Another useful extension of linear regression is the case where y is a linear function of two or more independent variables. For example, y might be a linear function of x_1 and x_2, as in

$$y = a_0 + a_1 x_1 + a_2 x_2 + e$$

Such an equation is particularly useful when fitting experimental data where the variable being studied is often a function of two other variables. For this two-dimensional case, the regression "line" becomes a "plane" (Fig. 15.3).

As with the previous cases, the "best" values of the coefficients are determined by formulating the sum of the squares of the residuals:

$$S_r = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_{1,i} - a_2 x_{2,i})^2 \qquad (15.4)$$

and differentiating with respect to each of the unknown coefficients:

$$\frac{\partial S_r}{\partial a_0} = -2 \sum (y_i - a_0 - a_1 x_{1,i} - a_2 x_{2,i})$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum x_{1,i} (y_i - a_0 - a_1 x_{1,i} - a_2 x_{2,i})$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum x_{2,i} (y_i - a_0 - a_1 x_{1,i} - a_2 x_{2,i})$$

$$y = a_0 + a_1 x_1 + a_2 x_2 + e$$

다항 회귀와 다변수 회귀의 비교

- $y_i - a_0 - a_1 x_i - a_2 {x_i}^2$ 형태의 다항회귀 계수 유도와 비슷하게 $y_i - a_0 - a_1 x_i - \boldsymbol{a_2} z_i$ 형태의 다변수 회귀의 계수 유도 함.
- 다항회귀 계수 유도
  - $S = min \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 {x_i}^2)^2$
  - $\frac{\partial S}{\partial a_0} = -2 \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 {x_i}^2)$
  - $\frac{\partial S}{\partial a_1} = -2 \sum_{i=1}^{n} x_i (y_i - a_0 - a_1 x_i - a_2 {x_i}^2)$
  - $\frac{\partial S}{\partial a_2} = -2 \sum_{i=1}^{n} {x_i}^2 (y_i - a_0 - a_1 x_i - a_2 {x_i}^2)$

$$\begin{bmatrix} n & \sum x_i & \sum {x_i}^2 \\ \sum x_i & \sum {x_i}^2 & \sum {x_i}^3 \\ \sum {x_i}^2 & \sum {x_i}^3 & \sum {x_i}^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum {x_i}^2 y_i \end{bmatrix}$$

- 다항회귀 연립방정식
  - $(n) a_0 + (\sum_{i=1}^{n} x_i)\, a_1 + (\sum_{i=1}^{n} {x_i}^2)\, a_2 = \sum_{i=1}^{n} y_i$
  - $(\sum_{i=1}^{n} x_i)\, a_0 + (\sum_{i=1}^{n} {x_i}^2)\, a_1 + (\sum_{i=1}^{n} {x_i}^3)\, a_2 = \sum_{i=1}^{n} x_i y_i$
  - $(\sum_{i=1}^{n} {x_i}^2)\, a_0 + (\sum_{i=1}^{n} {x_i}^3)\, a_1 + (\sum_{i=1}^{n} {x_i}^4)\, a_2 = \sum_{i=1}^{n} {x_i}^2 y_i$

다항 회귀와 다변수 회귀의 비교

- $y_i - a_0 - a_1 x_i - a_2 x_i^2$ 형태의 다항회귀 계수 유도와 비슷하게 $y_i - a_0 - a_1 x_i - \boldsymbol{a_2} z_i$ 형태의 다변수 회귀의 계수 유도 함.
- 다변수회귀 계수 유도
  - $S = min \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 z_i)^2$
  - $\frac{\partial S}{\partial a_0} = -2 \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 z_i)$
  - $\frac{\partial S}{\partial a_1} = -2 \sum_{i=1}^{n} x_i (y_i - a_0 - a_1 x_i - a_2 z_i)$
  - $\frac{\partial S}{\partial a_2} = -2 \sum_{i=1}^{n} z_i (y_i - a_0 - a_1 x_i - a_2 z_i)$
- 다변수회귀 연립방정식
  - $n a_0 + (\sum x_i) a_1 + (\sum z_i) a_2 = \sum y_i$
  - $(\sum x_i) a_0 + (\sum x_i^2) a_1 + (\sum x_i \cdot z_i) a_2 = \sum x_i \cdot y_i$
  - $(\sum z_i) a_0 + (\sum x_i \cdot z_i) a_1 + (\sum z_i^2) a_2 = \sum_{i=1}^{n} z_i \cdot y_i$

Multi-variable Regression (다변수 회귀)

- 편미분으로 Multi-variable Regression (다변수 회귀) 유도하기

$$\frac{\partial S}{\partial \boldsymbol{a_0}} = \frac{\partial}{\partial \boldsymbol{a_0}} \sum_{i=1}^{n} (y_i - \boldsymbol{a_0} - a_1 x_i - a_2 z_i)^2 = 0$$

$$y = \{f(x)\}^n$$

$$y' = n \cdot \{f(x)\}^{n-1} \cdot f'(x)$$

$$= 2 \cdot \sum_{i=1}^{n} (y_i - \boldsymbol{a_0} - a_1 x_i - a_2 z_i)^{2-1} \cdot \frac{\partial (y_i - \boldsymbol{a_0} - a_1 x_i - a_2 z_i)}{\partial \boldsymbol{a_0}}$$

$$= 2 \cdot \sum_{i=1}^{n} (y_i - \boldsymbol{a_0} - a_1 x_i - a_2 z_i) \cdot (\boldsymbol{-1})$$

$$= \boldsymbol{-2} \cdot \sum_{i=1}^{n} (y_i - \boldsymbol{a_0} - a_1 x_i - a_2 z_i) = \boldsymbol{0}$$

$$= \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} a_0 - \sum_{i=1}^{n} a_1 x_i - \sum_{i=1}^{n} a_2 z_i = 0$$

$\boldsymbol{a_0}$ 에 대해 편미분해서 $\frac{\partial S}{\partial \boldsymbol{a_0}} = \boldsymbol{0}$이 되는 $\boldsymbol{a_0}$ 값을 구해야 한다

- $S = min \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 z_i)^2$
- $\frac{\partial S}{\partial a_0} = -2 \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 z_i)$
- $\frac{\partial S}{\partial a_1} = -2 \sum_{i=1}^{n} x_i (y_i - a_0 - a_1 x_i - a_2 z_i)$
- $\frac{\partial S}{\partial a_2} = -2 \sum_{i=1}^{n} z_i (y_i - a_0 - a_1 x_i - a_2 z_i)$

Multi-variable Regression (다변수 회귀)

- 편미분으로 Multi-variable Regression (다변수 회귀) 유도하기

$$\sum_{i=1}^{n} y_i - \sum_{i=1}^{n} a_0 - \sum_{i=1}^{n} a_1 x_i - \sum_{i=1}^{n} a_2 z_i = 0$$

계속 전개 하면

$$\sum_{i=1}^{n} a_0 + \sum_{i=1}^{n} a_1 x_i + \sum_{i=1}^{n} a_2 z_i = \sum_{i=1}^{n} y_i$$

$$n \cdot a_0 + \sum_{i=1}^{n} a_1 x_i + \sum_{i=1}^{n} a_2 z_i = \sum_{i=1}^{n} y_i$$

$$\sum_{i=1}^{n} a_0 = a_0 \cdot \sum_{i=1}^{n} 1 = a_0 \cdot (1 + 1 + \cdots + 1) = a_0 \cdot n = n \cdot a_0$$

$$\boldsymbol{n a_0 + \left(\sum x_i\right) a_1 + \left(\sum z_i\right) a_2 = \sum y_i} \quad ①$$

a_0 에 대해 ① 번 식이 유도 된다

Multi-variable Regression (다변수 회귀)

- 편미분으로 Multi-variable Regression (다변수 회귀) 유도하기

$$\frac{\partial S}{\partial \boldsymbol{a_1}} = \frac{\partial}{\partial a_1}\sum_{i=1}^{n}(y_i - a_0 - \boldsymbol{a_1}x_i - a_2 z_i)^2 = 0$$

$$y = \{f(x)\}^n$$

$$y' = n \cdot \{f(x)\}^{n-1} \cdot f'(x)$$

$$= 2 \cdot \sum_{i=1}^{n}(y_i - a_0 - \boldsymbol{a_1}x_i - a_2 z_i)^{2-1} \cdot \frac{\partial(y_i - a_0 - \boldsymbol{a_1}x_i - a_2 z_i)}{\partial \boldsymbol{a_1}}$$

$$= -2\boldsymbol{x_i} \cdot \sum_{i=1}^{n}(y_i - a_0 - \boldsymbol{a_1}x_i - a_2 z_i) = 0$$

$$= \boldsymbol{x_i} \cdot \sum_{i=1}^{n} y_i - \boldsymbol{x_i} \cdot \sum_{i=1}^{n} a_0 - \boldsymbol{x_i} \cdot \sum_{i=1}^{n} a_1 x_i - \boldsymbol{x_i} \cdot \sum_{i=1}^{n} a_2 z_i = 0$$

$$= \sum_{i=1}^{n} \boldsymbol{x_i} \cdot y_i - \sum_{i=1}^{n} \boldsymbol{x_i} \cdot a_0 - \sum_{i=1}^{n} a_1 \boldsymbol{x_i} \cdot x_i - \sum_{i=1}^{n} a_2 \boldsymbol{x_i} \cdot z_i = 0$$

$\boldsymbol{a_1}$ 에 대해 편미분해서 $\frac{\partial S}{\partial \boldsymbol{a_1}} = \boldsymbol{0}$이 되는 $\boldsymbol{a_1}$ 값을 구해야 한다

$\boldsymbol{x_i}$ 를 $\sum_{i=1}^{n}$ 안에 넣어서 계산해서 전개한다

Multi-variable Regression (다변수 회귀)

$$\sum_{i=1}^{n} \boldsymbol{x_i} \cdot y_i - \sum_{i=1}^{n} \boldsymbol{x_i} \cdot a_0 - \sum_{i=1}^{n} a_1 \boldsymbol{x_i} \cdot x_i - \sum_{i=1}^{n} a_2 \boldsymbol{x_i} \cdot z_i = 0$$

전개한다

$$\sum_{i=1}^{n} \boldsymbol{x_i} \cdot y_i - \sum_{i=1}^{n} \boldsymbol{a_0} x_i - \sum_{i=1}^{n} a_1 {x_i}^2 - \sum_{i=1}^{n} a_2 \boldsymbol{x_i} \cdot z_i = 0$$

$$\sum_{i=1}^{n} \boldsymbol{a_0} x_i + \sum_{i=1}^{n} a_1 {x_i}^2 + \sum_{i=1}^{n} a_2 \boldsymbol{x_i} \cdot z_i = \sum_{i=1}^{n} \boldsymbol{x_i} \cdot y_i$$

$$\left(\sum \boldsymbol{x_i}\right) \boldsymbol{a_0} + \left(\sum \boldsymbol{x_i}^2\right) \boldsymbol{a_1} + \left(\sum \boldsymbol{x_i} \cdot z_i\right) \boldsymbol{a_2} = \sum \boldsymbol{x_i} \cdot y_i \quad ②$$

$\boldsymbol{a_1}$ 에 대해 ② 번 식이 유도 된다

Multi-variable Regression (다변수 회귀)

- 지금까지 나온 ① 식과 ② 식을 같이 적어 보면 다음과 같다

$$n a_0 + \left(\sum x_i\right) a_1 + \left(\sum z_i\right) a_2 = \sum y_i \quad ①$$

$$\left(\sum x_i\right) a_0 + \left(\sum x_i^2\right) a_1 + \left(\sum x_i \cdot z_i\right) a_2 = \sum x_i \cdot y_i \; ②$$

Multi-variable Regression (다변수 회귀)

- 편미분으로 Multi-variable Regression (다변수 회귀) 유도하기

$$\frac{\partial S}{\partial a_2} = \frac{\partial}{\partial a_2}\sum_{i=1}^{n}(y_i - a_0 - a_1 x_i - a_2 z_i)^2 = 0$$

$$y = \{f(x)\}^n$$

$$y' = n \cdot \{f(x)\}^{n-1} \cdot f'(x)$$

$$= 2 \cdot \sum_{i=1}^{n}(y_i - a_0 - a_1 x_i - a_2 z_i)^{2-1} \cdot \frac{\partial(y_i - a_0 - a_1 x_i - a_2 z_i)}{\partial a_2}$$

$$= -2z_i \cdot \sum_{i=1}^{n}(y_i - a_0 - a_1 x_i - a_2 z_i) = 0$$

$$= z_i \cdot \sum_{i=1}^{n} y_i - z_i \cdot \sum_{i=1}^{n} a_0 - z_i \cdot \sum_{i=1}^{n} a_1 x_i - z_i \cdot \sum_{i=1}^{n} a_2 z_i = 0$$

$$= \sum_{i=1}^{n} z_i \cdot y_i - \sum_{i=1}^{n} z_i \cdot a_0 - \sum_{i=1}^{n} a_1 z_i \cdot x_i - \sum_{i=1}^{n} a_2 {z_i}^2 = 0$$

a_2 에 대해 편미분해서 $\frac{\partial S}{\partial a_2} = 0$이 되는 a_2 값을 구해야 한다

${x_i}^2$ 을 $\sum_{i=1}^{n}$ 안에 넣어서 계산해서 전개한다

Multi-variable Regression (다변수 회귀)

$$\sum_{i=1}^{n} z_i \cdot y_i - \sum_{i=1}^{n} z_i \cdot a_0 - \sum_{i=1}^{n} a_1 z_i \cdot x_i - \sum_{i=1}^{n} a_2 {z_i}^2 = 0$$

$$\sum_{i=1}^{n} z_i \cdot y_i - \sum_{i=1}^{n} a_0 z_i - \sum_{i=1}^{n} a_1 z_i \cdot x_i - \sum_{i=1}^{n} a_2 {z_i}^2 = 0$$

$$\sum_{i=1}^{n} a_0 z_i + \sum_{i=1}^{n} a_1 z_i \cdot x_i + \sum_{i=1}^{n} a_2 {z_i}^2 = \sum_{i=1}^{n} z_i \cdot y_i$$

$$\left(\sum z_i\right) a_0 + \left(\sum z_i \cdot x_i\right) a_1 + \left(\sum {z_i}^2\right) a_2 = \sum_{i=1}^{n} z_i \cdot y_i \quad ③$$

전개한다

a_2 에 대해 ③번 식이 유도된다

Multi-variable Regression (다변수 회귀)

- 미지의 수 a_0, a_1, a_2에 대하여, 세 개의 방정식 ①, ②, ③ 을 $\frac{\partial S}{\partial a_0} = 0, \frac{\partial S}{\partial a_1} = 0, \frac{\partial S}{\partial a_2} = 0$ 으로 구했다.
- 여기서 $S = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 z_i)^2$이다. 모든 실제 값과 모델 값의 에러의 합이다.

$$n a_0 + \left(\sum x_i\right) a_1 + \left(\sum z_i\right) a_2 = \sum y_i \quad ①$$

$$\left(\sum x_i\right) a_0 + \left(\sum x_i^2\right) a_1 + \left(\sum x_i \cdot z_i\right) a_2 = \sum x_i \cdot y_i \quad ②$$

$$\left(\sum z_i\right) a_0 + \left(\sum z_i \cdot x_i\right) a_1 + \left(\sum z_i^2\right) a_2 = \sum_{i=1}^{n} z_i \cdot y_i \quad ③$$

행렬 전개를 통한 해 구하기

- 각 ① , ② , ③ 식에 곱해서 통합 정리해서 구하는 기존의 방법은 해 (a_0, a_1, a_2) 를 구하기가 굉장히 어려움 → 행렬 전개를 통한 해를 구함

$$na_0 + \left(\sum x_i\right)a_1 + \left(\sum {x_i}^2\right)a_2 = \sum y_i \quad ①$$

$$\left(\sum x_i\right)a_0 + \left(\sum {x_i}^2\right)a_1 + \left(\sum {x_i}^3\right)a_2 = \sum_{i=1}^{n} x_i \cdot y_i \quad ②$$

$$\left(\sum {x_i}^2\right)a_0 + \left(\sum {x_i}^3\right)a_1 + \left(\sum {x_i}^4\right)a_2 = \sum_{i=1}^{n} {x_i}^2 \cdot y_i \quad ③$$

$$\begin{bmatrix} n & \sum x_i & \sum {x_i}^2 \\ \sum x_i & \sum {x_i}^2 & \sum {x_i}^3 \\ \sum {x_i}^2 & \sum {x_i}^3 & \sum {x_i}^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum {x_i}^2 y_i \end{bmatrix}$$

$$na_0 + \left(\sum x_i\right)a_1 + \left(\sum z_i\right)a_2 = \sum y_i \quad ①$$

$$\left(\sum x_i\right)a_0 + \left(\sum {x_i}^2\right)a_1 + \left(\sum x_i \cdot z_i\right)a_2 = \sum x_i \cdot y_i \quad ②$$

$$\left(\sum z_i\right)a_0 + \left(\sum z_i \cdot x_i\right)a_1 + \left(\sum {z_i}^2\right)a_2 = \sum_{i=1}^{n} z_i \cdot y_i \quad ③$$

$$\begin{bmatrix} n & \sum x_i & \sum z_i \\ \sum x_i & \sum {x_i}^2 & \sum x_i \cdot z_i \\ \sum z_i & \sum x_i \cdot z_i & \sum {z_i}^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum z_i y_i \end{bmatrix}$$

다변수 선형 회귀와 텐서플로우

- **weight가 1개가 아닌 여러 개(w1, w2)를 학습시킨다**
- **GradientDecentOptimizer() 안에 행렬 전개를 통한 해를 구하는 과정이 포함되어 있다. 추상화가 잘 되어 있다.**

```
import tensorflow as tf
#data
x1_data = [1,0,3,0,5]
x2_data = [0,2,0,4,0]
y_data = [1,2,3,4,5]
# W1=[1.0000001] W2=[1.0000001] b=[-2.9742586e-07]
W1 = tf.Variable(tf.random_uniform([1],-1.0,1.0))
W2 = tf.Variable(tf.random_uniform([1],-1.0,1.0))
b = tf.Variable(tf.random_uniform([1],-1.0,1.0))
#hypothesis
hypothesis = W1 * x1_data + W2 * x2_data + b
cost = tf.reduce_mean(tf.square(hypothesis - y_data))
#minimize
a = tf.Variable(0.1) #alpha, learning rate
optimizer = tf.train.GradientDescentOptimizer(a)
train = optimizer.minimize(cost)
```

다변수 선형 회귀와 텐서플로우

- **weight가 1개가 아닌 여러 개(w1, w2)를 학습시킨다**
- **GradientDecentOptimizer() 안에 행렬 전개를 통한 해를 구하는 과정이 포함되어 있다. 추상화가 잘 되어 있다.**

```
# before starting, initialize variables
init = tf.initialize_all_variables()
#launch
sess = tf.Session()
sess.run(init)

# fit the line
for step in range(2001):
    sess.run(train)
    if step % 20 == 0:
        print (step, sess.run(cost), sess.run(W1),
        sess.run(W2), sess.run(b) )
```

다변수 선형 회귀

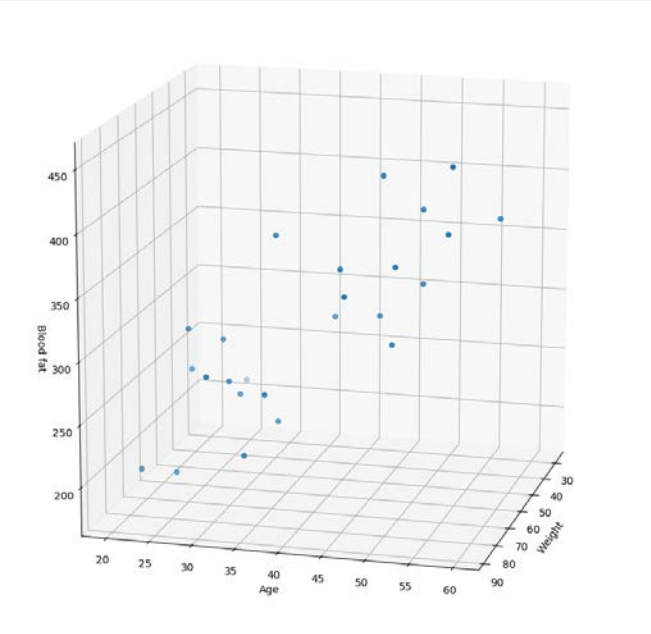
scikit-learn을 통해 다변수 선형 회귀를 이해한다

https://github.com/SCKIMOSU/Numerical-Analysis/blob/master/regression_multivariable.py

국민대학교 소프트웨어학부

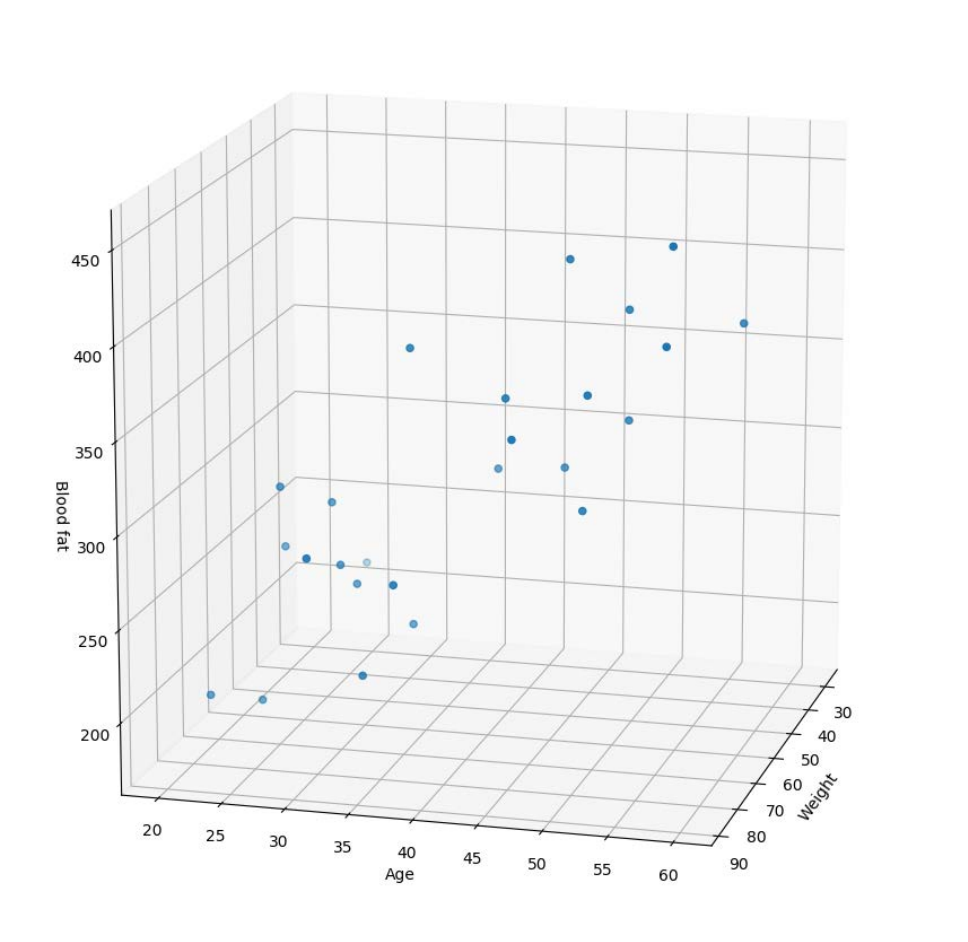
다변수 선형 회귀 (Multiple Linear Regression)

- 단변수 선형 회귀 분석은 하나의 독립변수만을 가지고 종속변수를 예측하기 위한 회귀 모형을 만드는 과정
- 다변수 선형 회귀 분석은 여러 개의 독립변수를 가지고 종속변수를 예측하기 위한 회귀 모형을 만드는 것
- 몸무게, 나이의 독립변수를 가지고 혈중 지방을 예측한다 .



다변수 선형 회귀 (Multiple Linear Regression)

- 몸무게, 나이, 그리고 혈중 지방 함량 데이터이다 .



다변수 선형 회귀 (Multiple Linear Regression)

- scikit-learn을 통해 다변수 선형 회귀를 이해한다

```
import numpy as np
import matplotlib.pyplot as plt
import sklearn.linear_model
from mpl_toolkits.mplot3d import Axes3D

raw_data = np.genfromtxt('x09.txt', skip_header=36)

xs = np.array(raw_data[:,2], dtype=np.float32)
ys = np.array(raw_data[:,3], dtype=np.float32)
zs = np.array(raw_data[:,4], dtype=np.float32)

fig = plt.figure(figsize=(12,12))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(xs, ys, zs)
ax.set_xlabel('Weight')
ax.set_ylabel('Age')
ax.set_zlabel('Blood fat')
ax.view_init(15, 15)

plt.show()
```

다변수 선형 회귀 (Multiple Linear Regression)

- scikit-learn.linear_model.LinearRegression() 통해 다변수 선형 회귀를 이해한다.
- fit()메소드를 통해 다변수 선형회귀 분석을 한다.

```
X = np.array(raw_data[:,2:4], dtype=np.float64)
# 두 개의 변수가 사용된다.
# ax.set_xlabel('Weight')
# ax.set_ylabel('Age')
y = np.array(raw_data[:,4], dtype=np.float64)

model = sklearn.linear_model.LinearRegression()
model.fit(X, y)
print(model)
print('Est [100,40] : ', model.predict([[100,40]]))
print('Est [60,25] : ', model.predict([[60,25]]))
knn = sklearn.neighbors.KNeighborsRegressor(n_neighbors=3)
knn.fit(X, y)
print(knn)
print('Est [100,40] : ', knn.predict([[100,40]]))
print('Est [60,25] : ', knn.predict([[60,25]]))
```

콘솔 디버깅: np.genfromtxt () 메소드

- raw_data = np.genfromtxt('x09.txt', skip_header=36)
- 몸무게(Weight, kilograms), 나이 (Age, Years), 혈중 지방(Blood fat content) 의 데이터 집합을 raw_data로 가져온다.

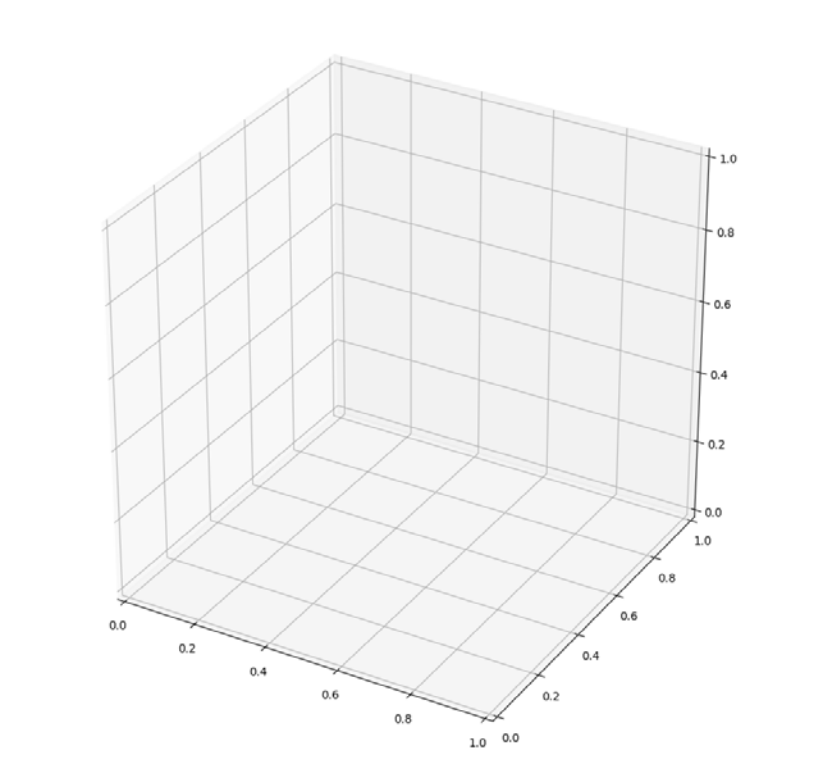
```
array([[  1.,   1.,  84.,  46., 354.],
       [  2.,   1.,  73.,  20., 190.],
       [  3.,   1.,  65.,  52., 405.],
       [  4.,   1.,  70.,  30., 263.],
       [  5.,   1.,  76.,  57., 451.],
       [  6.,   1.,  69.,  25., 302.],
       [  7.,   1.,  63.,  28., 288.],
       [  8.,   1.,  72.,  36., 385.],
       [  9.,   1.,  79.,  57., 402.],
```

콘솔 디버깅: 변수 xs, ys, zs 내용 확인

- 몸무게(Weight, kilograms)를 변수 xs에 저장한다.
  - xs = np.array(raw_data[:,2], dtype=np.float32)
  - array([84., 73., 65., 70., 76., 69., 63., 72., 79., 75., 27., 89., 65.,57., 59., 69., 60., 79., 75., 82., 59., 67., 85., 55., 63.], dtype=float32)
- 나이 (Age, Years) 를 변수 ys에 저장한다
  - ys = np.array(raw_data[:,3], dtype=np.float32)
  - array([46., 20., 52., 30., 57., 25., 28., 36., 57., 44., 24., 31., 52., 23., 60., 48., 34., 51., 50., 34., 46., 23., 37., 40., 30.], dtype=float32)
- 혈중 지방(Blood fat content) 변수 zs에 저장한다
  - zs = np.array(raw_data[:,4], dtype=np.float32)
  - array([354., 190., 405., 263., 451., 302., 288., 385., 402., 365., 209., 290., 346., 254., 395., 434., 220., 374., 308., 220., 311., 181., 274., 303., 244.], dtype=float32)

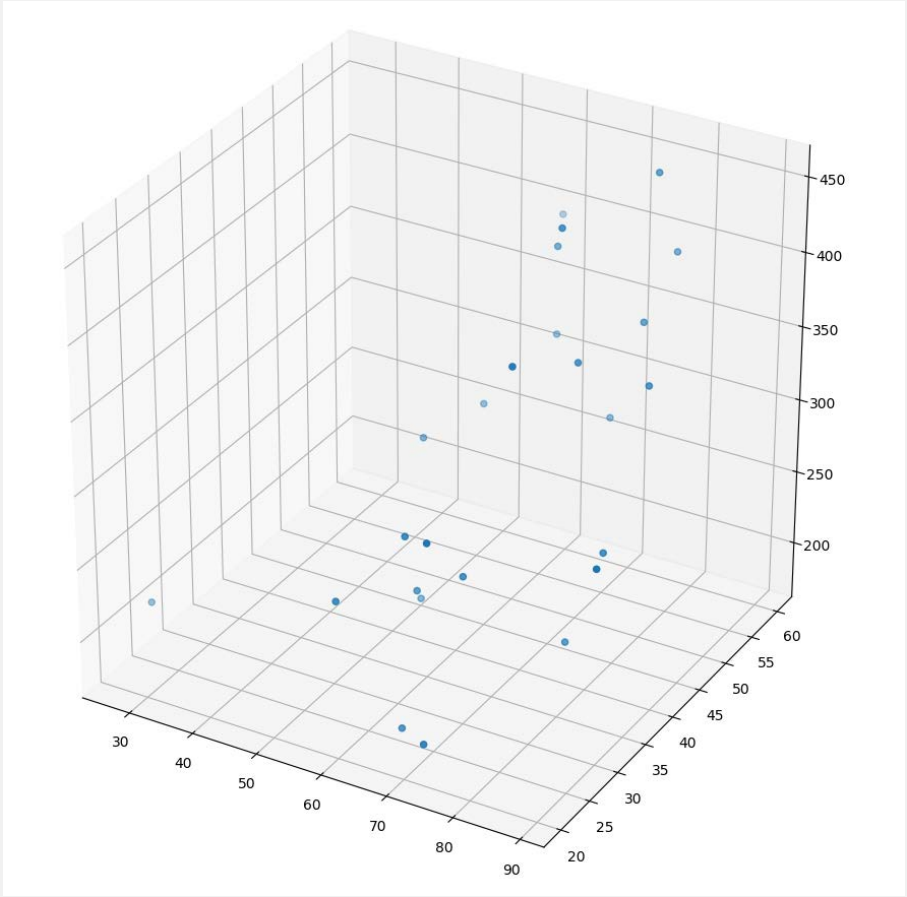
콘솔 디버깅: 3차원 그래프 그리기

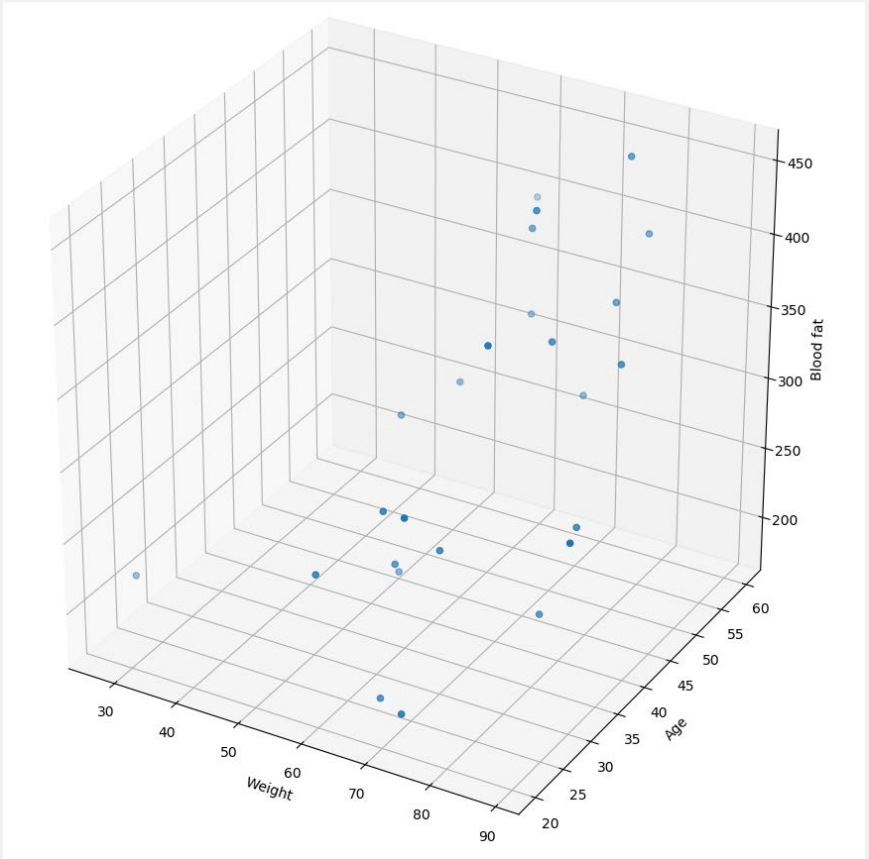
- fig = plt.figure(figsize=(12,12)) 로 그림 사이즈를 지정한다.
- ax = fig.add_subplot(111, projection='3d') 로 3차원 그래프를 그릴 수 있도록 한다.



콘솔 디버깅: 3차원 그래프 그리기

- ax.scatter(xs, ys, zs) 함수로 몸무게 (변수 xs), 나이 (변수 ys), 혈중 지방(변수 zs)를 스캐터링(데이터를 흩어 뿌리는 일) 한다





콘솔 디버깅: 예측 모델 만들기

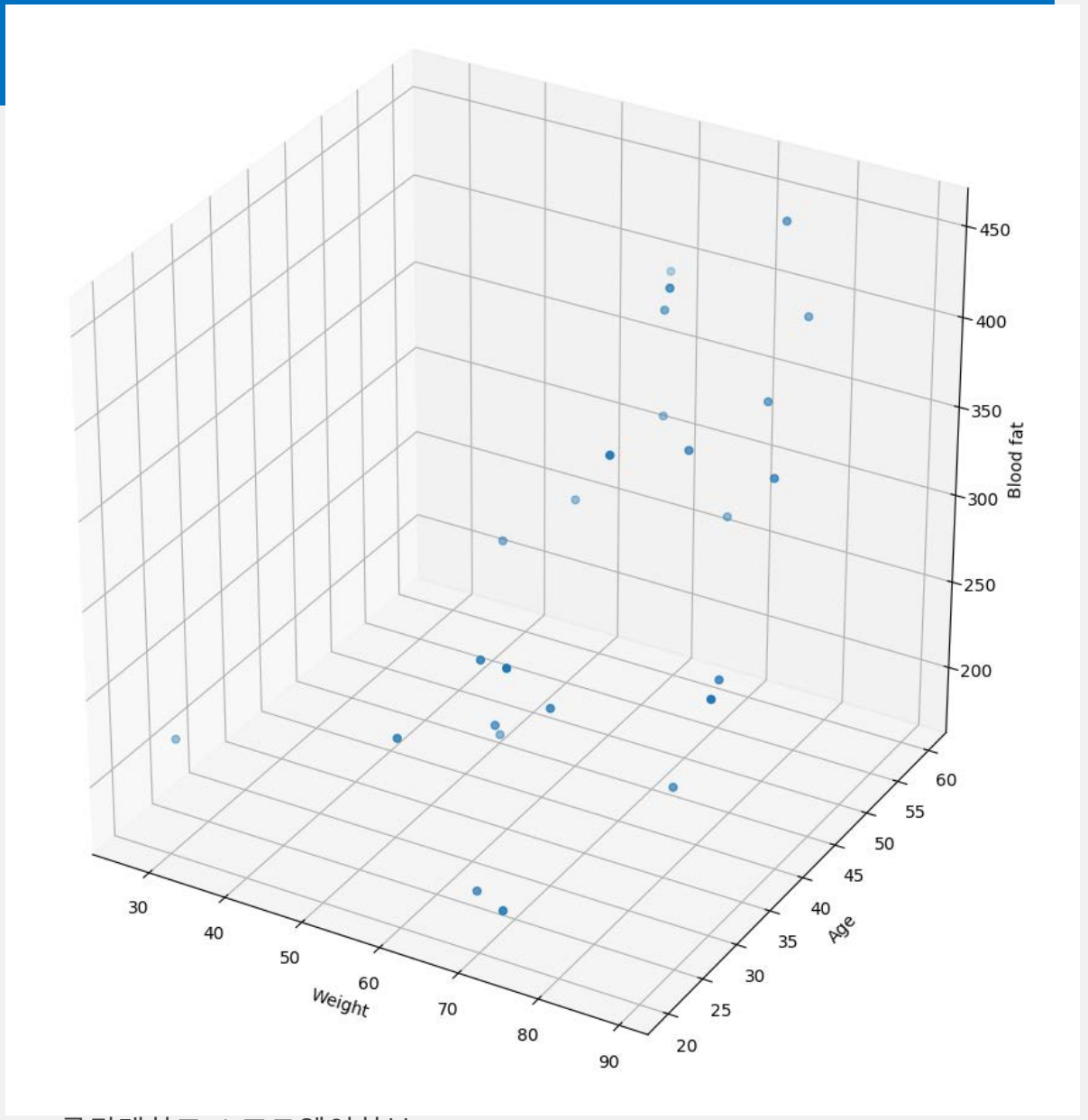
- 3차원 그래프를 통해, 몸무게 (변수 xs)와 나이 (변수 ys)에 따른 혈중 지방(변수 zs)을 예측하려고 한다는 것을 알 수 있다.
- X = np.array(raw_data[:,2:4], dtype=np.float64)
  - 몸무게 (변수 xs)와 나이 (변수 ys), 두 개의 변수가 X 에 사용된다.
  - array([[84., 46.],
  - [73., 20.],
  - [65., 52.],
  - [70., 30.],
- y = np.array(raw_data[:,4], dtype=np.float64)
  - 혈중 지방(변수 zs), 변수가 y 에 사용된다.
  - array([354., 190., 405., 263., 451., 302., 288., 385., 402., 365., 209. 290., 346., 254., 395., 434., 220., 374., 308., 220., 311., 181., 274., 303., 244.])

콘솔 디버깅: sklearn의 LinearRegression()

- sklearn의 LinearRegression() 메소드를 통해 선형 회귀 객체를 생성한다.
  - model = sklearn.linear_model.LinearRegression()
  - LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
- model.fit(X, y) 메소드를 통해 선형 회귀를 실시한다.

예측

- 몸무게 (변수 xs), 100kg, 나이 (변수 ys), 40세인 사람의 혈중 지방(변수 zs)을 예측을 실시한다
- 몸무게 100kg는 현재 데이터 셋에 나타나 있지 않다.
- 나이 40세는 나타나 있다.



콘솔 디버깅: model.predict() 메소드를 통해 예측

- 몸무게 (변수 xs), 100kg, 나이 (변수 ys), 40세인 사람의 혈중 지방(변수 zs)을 예측을 실시한다
- 예측 결과는? 328.38 이다.
  - print('Est [100,40] : ', model.predict([[100,40]])) 를 통해 예측을 실시한다.
  - Est [100,40] : [328.38238085]
- 몸무게 (변수 xs), 60kg, 나이 (변수 ys), 25세인 사람의 혈중 지방(변수 zs)을 예측을 실시한다
- 예측 결과는? 233. 43 이다.
  - print('Est [60,25] : ', model.predict([[60,25]])) 를 통해 예측을 실시한다.
  - Est [60,25] : [233.43903476]

General Linear Regression (일반 선형 회귀)

일반 선형 회귀를 이해한다

국민대학교 소프트웨어학부

General Linear Regression (일반 선형 회귀)

- $z_1, z_2, \ldots, z_n$ 의 n 개 함수로 곡선 접합 수행

15.3 GENERAL LINEAR LEAST SQUARES

In the preceding pages, we have introduced three types of regression: simple linear, polynomial, and multiple linear. In fact, all three belong to the following general linear least-squares model:

$$y = a_0 z_0 + a_1 z_1 + a_2 z_2 + \cdots + a_m z_m + e \tag{15.7}$$

where $z_0, z_1, \ldots, z_m$ are $m + 1$ basis functions. It can easily be seen how simple linear and multiple linear regression fall within this model—that is, $z_0 = 1, z_1 = x_1, z_2 = x_2, \ldots, z_m = x_m$. Further, polynomial regression is also included if the basis functions are simple monomials as in $z_0 = 1, z_1 = x, z_2 = x^2, \ldots, z_m = x^m$.

Note that the terminology "linear" refers only to the model's dependence on its parameters—that is, the a's. As in the case of polynomial regression, the functions themselves can be highly nonlinear. For example, the z's can be sinusoids, as in

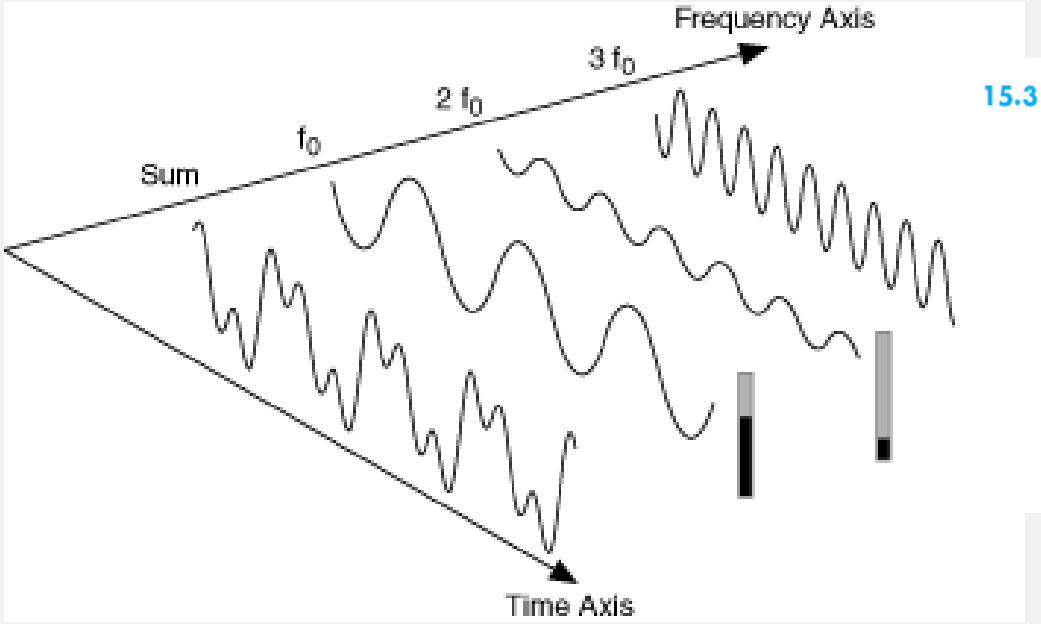
$$y = a_0 + a_1 \cos(\omega x) + a_2 \sin(\omega x)$$

Curve Fitting을 위한 Regression 방법 비교

- Linear Regression (선형 회귀)
  - 1개 변수 x 의 일차 방정식을 사용하여 곡선 접합 수행
  - Numpy, scikit-learn, tensorflow 메소드 이용
- Polynomial Regression (다항 회귀)
  - 1개 변수의 x 의 2차, 3차 방정식(x^2, x^3)을 사용하여 곡선 접합 수행
  - Numpy, scikit-learn, tensorflow 메소드 이용
- Multiple Linear Regression (다변수 선형 회귀)
  - 두 개 이상의 변수 x, y 또는 x_1, x_2 의 일차 방정식을 사용하여 곡선 접합 수행
  - **scikit-learn, tensorflow 메소드 이용**
- General Linear Regression (일반 선형 회귀)
  - 두 개 이상의 함수 $z_1, z_2, \ldots, z_n$ 의 n 개 일차 함수로 곡선 접합 수행

General Linear Regression (일반선형회귀)

- $z_1, z_2, \ldots, z_n$ 의 n 개 함수로 곡선 접합 수행
- **Fourier Analysis는 Curve Fitting을 위하여 Sinusoid 함수를 사용**
- **General Linear Regression 중의 하나다.**



15.3 GENERAL LINEAR LEAST SQUARES

In the preceding pages, we have introduced three types of regression: simple linear, polynomial, and multiple linear. In fact, all three belong to the following general linear least-squares model:

$$y = a_0 z_0 + a_1 z_1 + a_2 z_2 + \cdots + a_m z_m + e \tag{15.7}$$

where $z_0, z_1, \ldots, z_m$ are $m + 1$ basis functions. It can easily be seen how simple linear and multiple linear regression fall within this model—that is, $z_0 = 1, z_1 = x_1, z_2 = x_2, \ldots, z_m = x_m$. Further, polynomial regression is also included if the basis functions are simple monomials as in $z_0 = 1, z_1 = x, z_2 = x^2, \ldots, z_m = x^m$.

Note that the terminology "linear" refers only to the model's dependence on its parameters—that is, the a's. As in the case of polynomial regression, the functions themselves can be highly nonlinear. For example, the z's can be sinusoids, as in

$$y = a_0 + a_1 \cos(\omega x) + a_2 \sin(\omega x)$$