Polynomial Regression (Handwriting Assignment)

Name: 신다예

Student ID: 20210858

Instructor: Professor Kyungjae Lee

October 9, 2023

Introduction

In the mid-term project, we will look at a polynomial regression algorithm which can be used to fit non-linear data by using a polynomial function. The polynomial Regression is a form of regression analysis in which the relationship between the independent variable x and the dependent variable y is modeled as an nth degree polynomial in x.

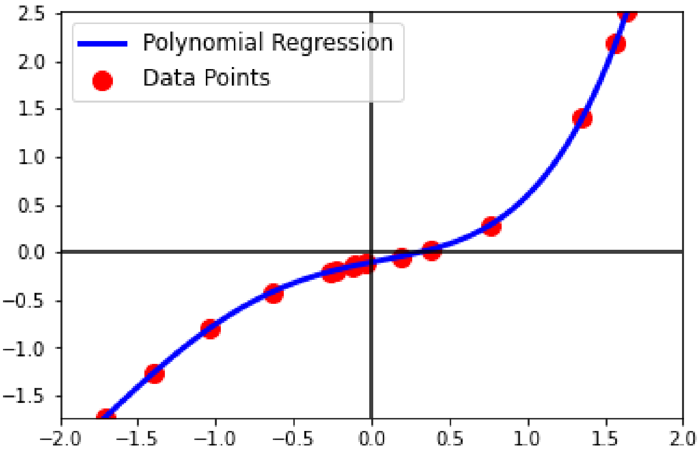


Figure 1: Example of Polynomial Regression

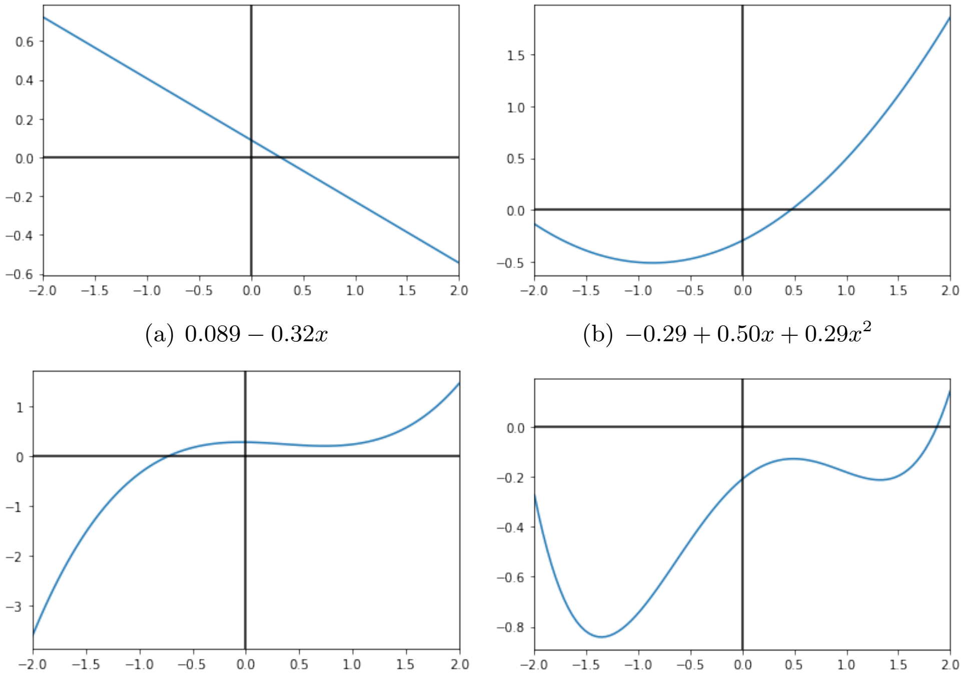
First, what is a regression? we can find a definition from the book as follows: *Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent and independent variable.* Actually, this definition is a bookish definition, in simple terms the regression can be defined as *finding a function that best explain data which consists of input and output pairs.* Let assume that we have 100 data points,

$$(x_1, y_1),\ (x_2, y_2),\ (x_3, y_3),\ \cdots (x_{98}, y_{98}),\ (x_{99}, y_{99}),\ (x_{100}, y_{100}).$$

The goal of regression is to find a function $\hat{f}$ such that

$$\hat{f}(x_1) = y_1,\ \hat{f}(x_2) = y_2,\ \hat{f}(x_3) = y_3,\ \cdots,\ \hat{f}(x_{99}) = y_{100},\ \hat{f}(x_{100}) = y_{100}.$$

This is the simplest definition of the regression problem. Note that many details about regression analysis are omitted here, but, you will learn more rigorous definition in other courses such as



(a) $0.089 - 0.32x$

(b) $-0.29 + 0.50x + 0.29x^2$

(c) $0.28 - 0.03x - 0.34x^2 + 0.32x^3$

(d) $-0.21 + 0.34x - 0.35x^2 - 0.06x^3 + 0.1x^4$

Figure 2: Examples of polynomial functions

machine learning or statistics. Then, the polynomial regression is the regression framework that employs the polynomial function to fit the data.

So, what is the polynomial function? I guess you may remember, from high school, the following functions:

$$
\begin{aligned}
\text{Degree of } 0 &: f(x) = w_0 \\
\text{Degree of } 1 &: f(x) = w_1 \cdot x + w_0 \\
\text{Degree of } 2 &: f(x) = w_2 \cdot x^2 + w_1 \cdot x + w_0 \\
\text{Degree of } 3 &: f(x) = w_3 \cdot x^3 + w_2 \cdot x^2 + w_1 \cdot x + w_0 \\
&\vdots \\
\text{Degree of } d &: f(x) = \sum_{i=0}^{d} w_i \cdot x^i,
\end{aligned}
$$

where $w_0, w_1, \cdots, w_d$ are a coefficient of polynomial and d is called a degree of a polynomial. So, we can determine a polynomial function $f(x)$ by deciding its degree d and corresponding coefficients $\{w_0, w_1, \cdots, w_d\}$. Figure 2 illustrates some examples of polynomial functions.

Then, the polynomial regression is a regression problem to find the best polynomial function to fit the given data points. Especially, the polynomial function is determined by coefficients (let just assume that d is fixed). We can restate the polynomial regression as *finding coefficients of polynomials such that, for all data point,* (x_i, y_i), $y_i = \hat{f}(x_i)$ *holds* (if we have noise free data). Figure 1 shows the example of polynomial regression. In the following problems, you have to study how to compute the coefficients of the polynomial to fit the data points.

Problems

1. (80 pt. in total)

Assume that we have n data points, $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$. Let the degree of polynomial be d. Then, we want to find $w_0, w_1, w_2, \cdots, w_d$ of the polynomial such that

$$
\begin{aligned}
\hat{f}(x_1) &= w_0 + w_1 x_1 + w_2 x_1^2 + \cdots + w_d x_1^d = y_1, \\
\hat{f}(x_2) &= w_0 + w_1 x_2 + w_2 x_2^2 + \cdots + w_d x_2^d = y_2, \\
\hat{f}(x_3) &= w_0 + w_1 x_3 + w_2 x_3^2 + \cdots + w_d x_3^d = y_3, \\
\hat{f}(x_4) &= w_0 + w_1 x_4 + w_2 x_4^2 + \cdots + w_d x_4^d = y_4, \\
\hat{f}(x_5) &= w_0 + w_1 x_5 + w_2 x_5^2 + \cdots + w_d x_5^d = y_5, \\
&\vdots \\
\hat{f}(x_n) &= w_0 + w_1 x_n + w_2 x_n^2 + \cdots + w_d x_n^d = y_n.
\end{aligned}
$$

Now, we reformulate the equations into the vector and matrix form. First, let $\mathbf{w} = [w_0, w_1, \cdots, w_d]^T$ and $\mathbf{y} = [y_1, y_2, \cdots, y_n]^T$. Then, the above equations can be rewritten as

$$
\hat{f}(x_1) = [1, x_1, x_1^2, x_1^3, \cdots, x_1^d] \cdot \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_d \end{bmatrix} = [1, x_1, x_1^2, x_1^3, \cdots, x_1^d]\mathbf{w} = y_1
$$

Similarly, we have,

$$
\begin{aligned}
[1, x_2, x_2^2, x_2^3, \cdots, x_2^d]\mathbf{w} &= y_2, \\
[1, x_3, x_3^2, x_3^3, \cdots, x_3^d]\mathbf{w} &= y_3, \\
[1, x_4, x_4^2, x_4^3, \cdots, x_4^d]\mathbf{w} &= y_4, \\
[1, x_5, x_5^2, x_5^3, \cdots, x_5^d]\mathbf{w} &= y_5, \\
&\vdots \\
[1, x_n, x_n^2, x_n^3, \cdots, x_n^d]\mathbf{w} &= y_n.
\end{aligned}
$$

Then, all equations can be written as the form of linear equation,

$$
A\mathbf{w} = \mathbf{y},
$$

where A is the stack of $[1, x_i, x_i^2, x_i^3, \cdots, x_i^d]$ for $i = 1, \cdots, n$. Under this setting, answer the following questions.

1-(a) What is the size of vector w and y? (10pt)

$$w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} \quad \text{size: } (d+1) \times 1 \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} \quad \text{size: } n \times 1$$

1-(b) What is the size of matrix A? Write A. (10pt)

$$A = \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{d-1} & \lambda_1^d \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{d-1} & \lambda_2^d \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \lambda_n & \lambda_n^2 & \cdots & \lambda_n^{d-1} & \lambda_n^d \end{bmatrix} \quad \text{size: } n \times (d+1)$$

1-(c) Let $d+1=n$, then, A becomes a square matrix. Compute the determinant of A. (40pt in total, Derivation: 30pt, Answer: 10pt)

$d+1=n$, A는 $n \times n$ matrix이다. → square matrix

$$A = \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{d-1} & \lambda_1^d \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{d-1} & \lambda_2^d \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \lambda_n & \lambda_n^2 & & \lambda_n^{d-1} & \lambda_n^d \end{bmatrix}$$

determinant을 구할때 정해지는 원소의 열 또는 행에 0이 많을수록 행렬식을 구할 때 계산이 간단해진다. 한 열의 스칼라배를 다른 열에 더해도 행렬식은 바뀌지 않기 때문에 기본열연산을 통해 1행을 간단하게 해본다

1)

$$\begin{bmatrix} 1 & \lambda_1 & \cdots & \lambda_1^d - \lambda_1 \cdot \lambda_1^{d-1} \\ 1 & \lambda_2 & \cdots & \lambda_2^d - \lambda_1 \cdot \lambda_2^{d-1} \\ \vdots & \vdots & & \vdots \\ 1 & \lambda_n & \cdots & \lambda_n^d - \lambda_1 \cdot \lambda_n^{d-1} \end{bmatrix} = \begin{bmatrix} 1 & \lambda_1 & \cdots & \lambda_1^{d-1} & 0 \\ 1 & \lambda_2 & \cdots & \lambda_2^{d-1} & \lambda_2^{d-1}(\lambda_2-\lambda_1) \\ \vdots & \vdots & & & \vdots \\ 1 & \lambda_n & \cdots & \lambda_n^{d-1} & \lambda_n^{d-1}(\lambda_n-\lambda_1) \end{bmatrix}$$

→ 1행이 $(1\ 0\ 0 \cdots 0)$ 이 될 수 있도록 열연산 반복

$$\sim \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & \lambda_2-\lambda_1 & \cdots & \lambda_2^{d-1}(\lambda_2-\lambda_1) \\ \vdots & \vdots & & \vdots \\ 1 & \lambda_n-\lambda_1 & \cdots & \lambda_n^{d-1}(\lambda_n-\lambda_1) \end{bmatrix}$$

→ 1행을 기준으로 행렬식을 구해보면

$$1 \times \det\left(\begin{bmatrix} \lambda_2-\lambda_1 & \lambda_2(\lambda_2-\lambda_1) & \cdots & \lambda_2^{d-1}(\lambda_2-\lambda_1) \\ \lambda_3-\lambda_1 & \lambda_3(\lambda_3-\lambda_1) & & \lambda_3^{d-1}(\lambda_3-\lambda_1) \\ \vdots & \vdots & & \vdots \\ \lambda_n-\lambda_1 & \lambda_n(\lambda_n-\lambda_1) & \cdots & \lambda_n^{d-1}(\lambda_n-\lambda_1) \end{bmatrix}\right)$$

$$= (\lambda_2-\lambda_1)(\lambda_3-\lambda_1)\cdots(\lambda_n-\lambda_1)\det\left(\begin{bmatrix} 1 & \lambda_2 & \cdots & \lambda_2^{d-1} \\ 1 & \lambda_3 & \cdots & \lambda_3^{d-1} \\ \vdots & \vdots & & \vdots \\ 1 & \lambda_n & \cdots & \lambda_n^{d-1} \end{bmatrix}\right)$$

이 행렬에 대해 앞서 $(1\ 0 \cdots 0)$을 만들기 위해 반복한 열연산 수행

2)

$$\begin{bmatrix} 1 & \lambda_2 & \cdots & \lambda_2^{d-2}(\lambda_2-\lambda_2) \\ 1 & \lambda_3 & \cdots & \lambda_3^{d-2}(\lambda_3-\lambda_2) \\ \vdots & \vdots & & \vdots \\ 1 & \lambda_n & \cdots & \lambda_n^{d-2}(\lambda_n-\lambda_2) \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & \lambda_3-\lambda_2 & \cdots & \lambda_3^{d-2}(\lambda_3-\lambda_2) \\ \vdots & \vdots & & \vdots \\ 1 & \lambda_n-\lambda_2 & \cdots & \lambda_n^{d-2}(\lambda_n-\lambda_2) \end{bmatrix}$$

→ 행렬식 계산

$$\left(\begin{bmatrix} \lambda_3-\lambda_2 & \lambda_3(\lambda_3-\lambda_2) & \cdots & \lambda_3^{d-2}(\lambda_3-\lambda_2) \\ \lambda_4-\lambda_2 & \lambda_4(\lambda_4-\lambda_2) & \cdots & \lambda_4^{d-2}(\lambda_4-\lambda_2) \\ \vdots & \vdots & & \vdots \\ \lambda_n-\lambda_2 & \lambda_n(\lambda_n-\lambda_2) & \cdots & \lambda_n^{d-2}(\lambda_n-\lambda_2) \end{bmatrix}\right) = (\lambda_3-\lambda_2)(\lambda_4-\lambda_2)\cdots(\lambda_n-\lambda_2)\det\left(\begin{bmatrix} 1 & \lambda_3 & \cdots & \lambda_3^{d-2} \\ 1 & \lambda_4 & \cdots & \lambda_4^{d-2} \\ \vdots & \vdots & & \\ 1 & \lambda_n & \cdots & \lambda_n^{d-2} \end{bmatrix}\right)$$

3) 앞서 진행한 열연산을 반복해 det(A)값이 스칼라값으로 나올때 까지 반복하면 다음과 같이 규칙이 생겨 정리가 가능하다.

4) $\det(A) = \prod_{1 \le i < j \le n} (\lambda_j - \lambda_i)$

1-(d) What is the condition that makes the determinant of A non-zero? (10pt)

det(A)는 x들의 차이의 곱으로 이루어져 있기 때문에 x의 값들은 서로 달라야 한다.

1-(e) Assume that the determinant of A is non-zero, then, what is the solution of linear equation, $A\mathbf{w} = \mathbf{y}$, with respect to $\mathbf{w}$? (10pt)

det(A)가 0이 아니면 A는 역행렬이 존재해서 $Aw=y$의 해가 존재한다.

$w = A^{-1}y$로 나타낼 수 있다

2. (20pt)

Suppose that $n > d + 1$. Then, we cannot compute the inverse of A since A is not a square matrix. In this case, how can we solve the linear equation $A\mathbf{w} = \mathbf{y}$?

$n>d+1$ 이 되면 A는 정방행렬이 아니라 역행렬이 정의되지못한다. 이런 행렬에서는 역행렬 대신 유사역행렬(pseudo inverse matrix)을 정의할 수 있다.

유사 역행렬은 특이값 분해에 의해 계산된다.

A^+을 A의 유사역행렬이라 했을 때 $A^+ = (A^TA)^{-1}A^T$이다. 여기서 A^TA는 정방행렬이 되어 역행렬이 존재한다

따라서 $Aw=y$

$$A^TAw = A^Ty$$

$$w = (A^TA)^{-1}A^Ty$$

$$w = A^+y$$

// 라고 표현할 수 있다.