

香 港 大 學
THE UNIVERSITY OF HONG KONG

Optimality Conditions for Unconstrained Programs

Prof. Li Jiayang

DASE7113 Optimization Methods for Intelligent Systems

Optimality Condition

Definition

Optimality conditions are mathematical conditions that can equivalently characterize the solution to optimization problems.

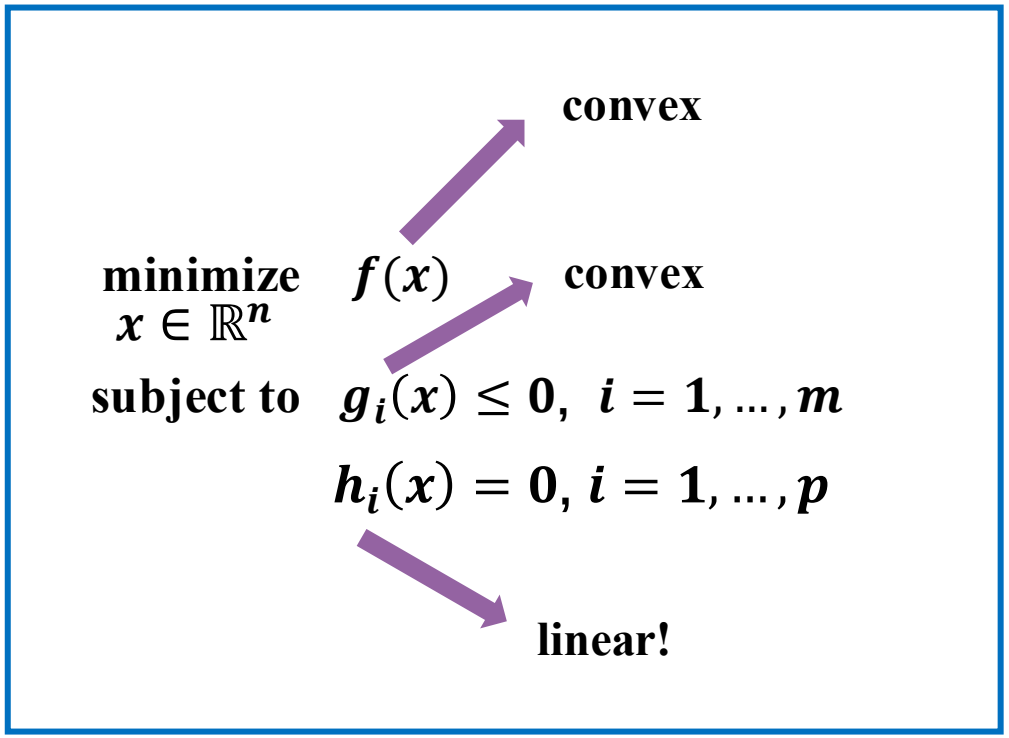
x^* solves the optimization problem

if and only if xxxx

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \ i = 1, \ldots, m \\ & h_i(x) = 0, \ i = 1, \ldots, p \end{aligned}$$

Optimality Condition

For most optimization problems, there are no straightforward optimality conditions.



But convex programs are exceptions…

Optimality Condition

Unconstrained Convex Programs

Given a differentiable convex function $f: \mathbb{R} \to \mathbb{R}$, x^* is the solution to the following optimization problem if and only if $f'(x^*) = 0$.

$$\begin{array}{ll} \text{minimize} & f(x) \\ x \in \mathbb{R} & \end{array}$$



Since $f(x)$ is convex, for x^* and all $x \in \mathbb{R}$, we have

$$f(x) \geq f(x^*) + f'(x^*) \cdot (x - x^*).$$

If $f'(x^*) = 0$, we will then have $f(x) \geq f(x^*)$ for all $x \in \mathbb{R}$.

Optimality Condition

Unconstrained Convex Programs

Given a differentiable convex function $f: \mathbb{R} \to \mathbb{R}$, x^* is the solution to the following optimization problem if and only if $f'(x^*) = 0$.

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &x \in \mathbb{R} \end{aligned}$$

Suppose that $f(x) \geq f(x^*)$ for all $x \in \mathbb{R}$

Right-hand derivative $$f'(x^{*,+}) = \lim_{x \to x^{*,+}} \frac{f(x) - f(x^*)}{x - x^*} \geq 0$$

Left-hand derivative $$f'(x^{*,-}) = \lim_{x \to x^{*,-}} \frac{f(x) - f(x^*)}{x - x^*} \leq 0$$

$$\Rightarrow \quad f'(x^*) = 0$$

Optimality Condition

Exercise

Consider $f: \mathbb{R} \to \mathbb{R}$ defined as $f(x) = a \cdot (e^x + a) - x$, where $a \in \mathbb{R}^{++}$ is a positive constant.

Prove that $f(x) > 2\log(a) + 1.5$.

$f''(x) = a \cdot e^x > 0$ for all $x \in \mathbb{R}$ ⇒ $f(x)$ is a convex function

Setting $f'(x^*) = a \cdot e^{x^*} - 1 = 0$, we get $x^* = -\log(a)$

$f(x) \geq f(-\log(a)) = 1 + a^2 + \log(a)$

… …

Optimality Condition

Unconstrained Convex Programs

Given a differentiable convex function $f: \mathbb{R}^n \to \mathbb{R}$, x^* is the solution to the following optimization problem if and only if $\nabla f(x^*) = 0$.

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &x \in \mathbb{R}^n \end{aligned}$$

Optimality Condition

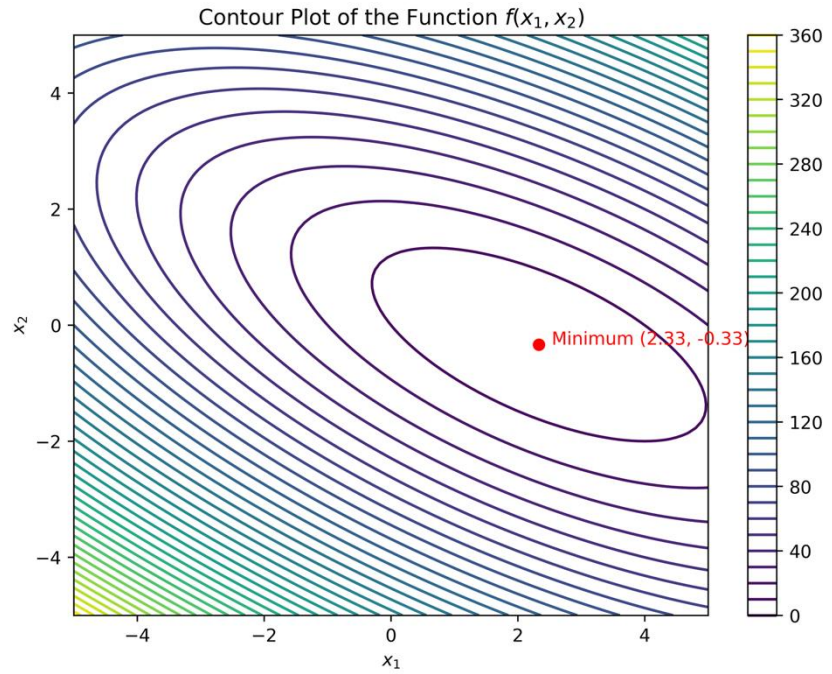
Exercise

Consider $f: \mathbb{R}^2 \to \mathbb{R}$ defined as $f(x_1, x_2) = 2x_1^2 + 4x_1x_2 + 5x_2^2 - 8x_1 - 6x_2 + 10$.

Where does the function attain its minimum?

Gradient: $\begin{bmatrix} 4x_1 + 4x_2 - 8 \\ 4x_1 + 10x_2 - 6 \end{bmatrix}$

Hessian: $\begin{bmatrix} 4 & 4 \\ 4 & 10 \end{bmatrix}$

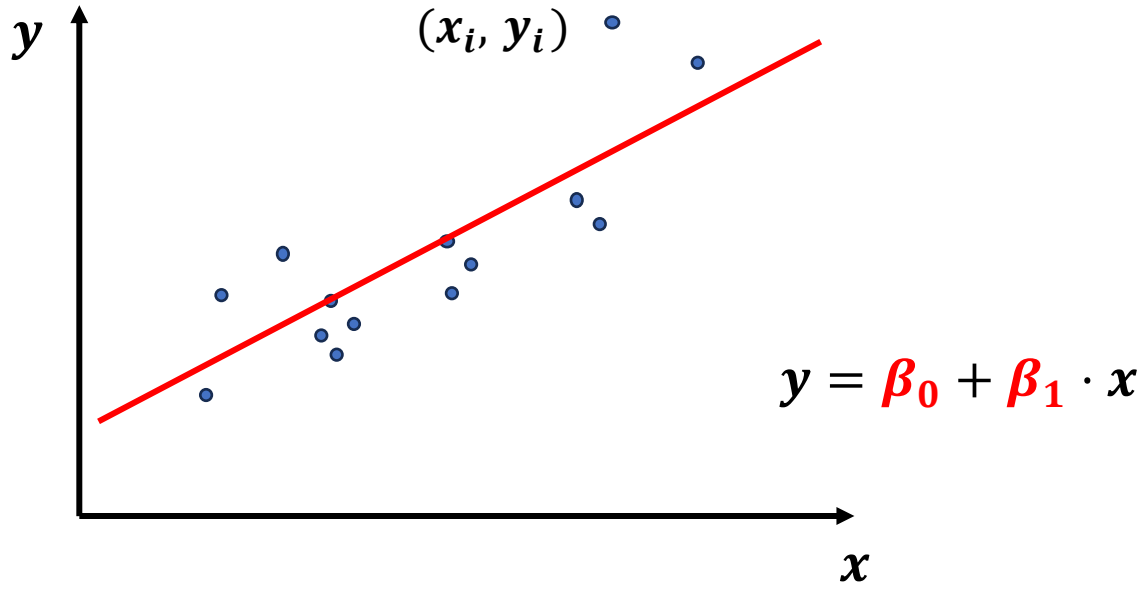


Linear Regression

Definition

A model $f(x; \beta)$ for fitting input-output data pairs is called a linear regression model

if $f(x; \beta)$ is a linear function in β.

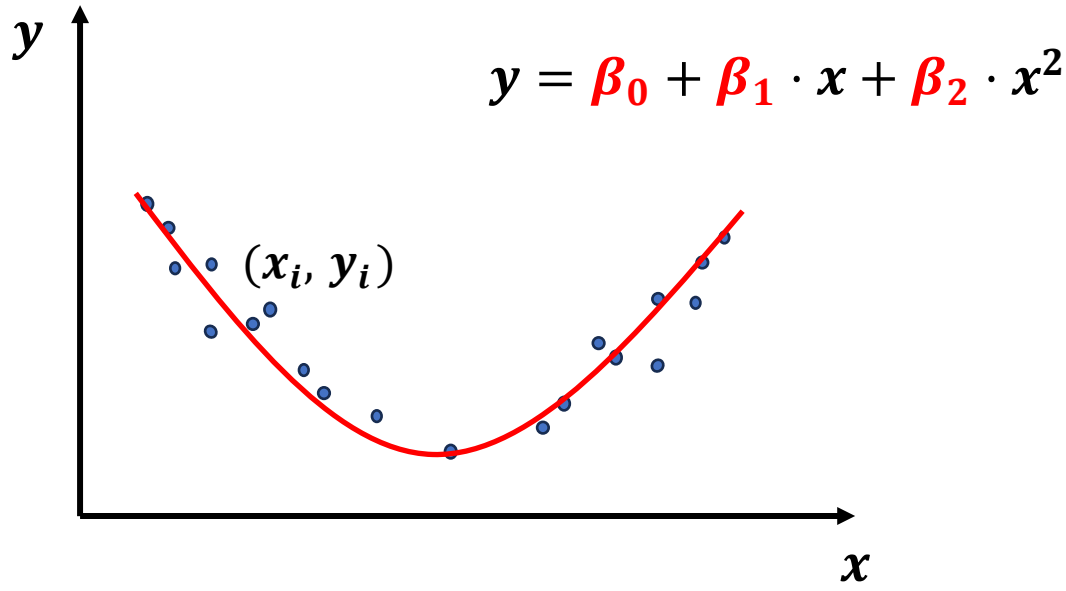


Linear Regression

Definition

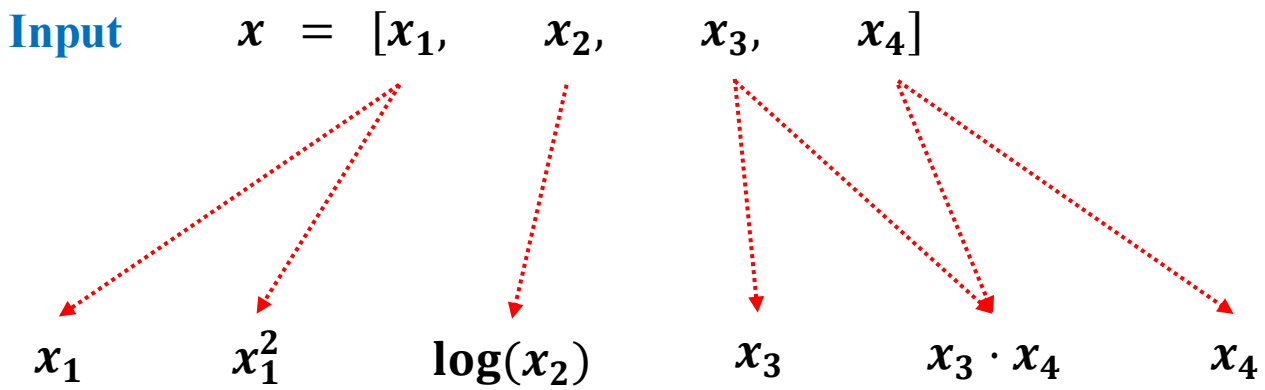
A model $f(x; \beta)$ for fitting input-output data pairs is called a linear regression model

if $f(x; \beta)$ is a linear function in β.



Linear Regression

A regression model is linear as long as the output is linear in the parameters;
there are no restrictions on how the output and the (original) input are related.



Output $y = \beta_1 \cdot x_1 + \beta_2 \cdot x_1^2 + \beta_3 \cdot \log(x_2) + \beta_4 \cdot x_3 + \beta_5 \cdot x_3 \cdot x_4 + \beta_6 \cdot x_4$

Linear Regression

Question

Consider the following linear regression model

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad l(\boldsymbol{\beta}) = \sum_{i=1}^{n} (\boldsymbol{\beta_0} + \boldsymbol{\beta_1} \cdot x_{i1} + \cdots + \boldsymbol{\beta_m} \cdot x_{im} - y_i)^2.$$

What is the optimal $\boldsymbol{\beta}^*$?

Linear Regression

Question

Consider the following linear regression model

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad l(\boldsymbol{\beta}) = \sum_{i=1}^{n} (\beta_0 + \beta_1 \cdot x_{i1} + \cdots + \beta_m \cdot x_{im} - y_i)^2.$$

What is the optimal $\boldsymbol{\beta}^*$?

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nm} \end{bmatrix} \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_m \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$$l(\boldsymbol{\beta}) = (\boldsymbol{X\beta} - \boldsymbol{y})^{\mathrm{T}}(\boldsymbol{X\beta} - \boldsymbol{y})$$

$$= \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X\beta} - 2\boldsymbol{y}^{\mathrm{T}}\boldsymbol{X\beta} + \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}$$

$$\Rightarrow \nabla l(\boldsymbol{\beta}) = 2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X\beta} - 2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$$

$$\Rightarrow \nabla^2 l(\boldsymbol{\beta}) = \boxed{2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}}$$ A PSD matrix

Linear Regression

Question

Consider the following linear regression model

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad l(\boldsymbol{\beta}) = \sum_{i=1}^{n} (\boldsymbol{\beta_0} + \boldsymbol{\beta_1} \cdot x_{i1} + \cdots + \boldsymbol{\beta_m} \cdot x_{im} - y_i)^2.$$

What is the optimal $\boldsymbol{\beta}^*$?

$$\underset{X}{\begin{bmatrix} 1 & x_{11} & \cdots & x_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nm} \end{bmatrix}} \quad \underset{\boldsymbol{\beta}}{\begin{bmatrix} \boldsymbol{\beta_0} \\ \vdots \\ \boldsymbol{\beta_m} \end{bmatrix}} \quad \underset{y}{\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}}$$

Setting $\nabla l(\boldsymbol{\beta}^*) = 2X^{\mathrm{T}}X\boldsymbol{\beta}^* - 2X^{\mathrm{T}}y = 0$

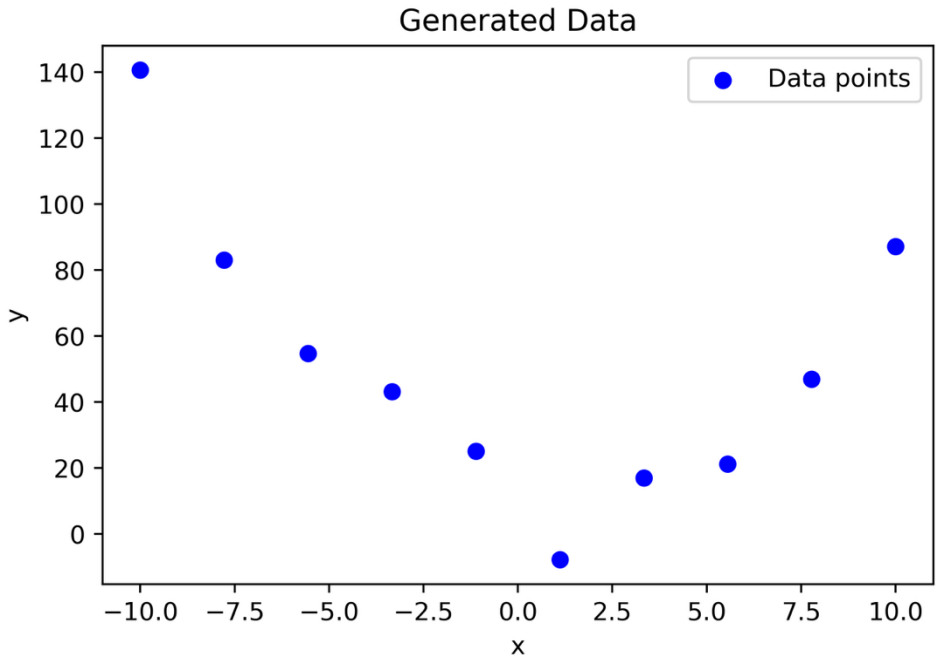
we then get $\boldsymbol{\beta}^* = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y$

Linear Regression

$$\frac{1}{n} \cdot \sum_{i=1}^{n} (\beta_0 + \beta_1 \cdot x_{i1} + \cdots + \beta_m \cdot x_{im} - y_i)^2$$

Coding

Train a linear regression model to fit the data pairs.
Report the coefficients in your model and the mean squared error (MSE).




```
import numpy as np

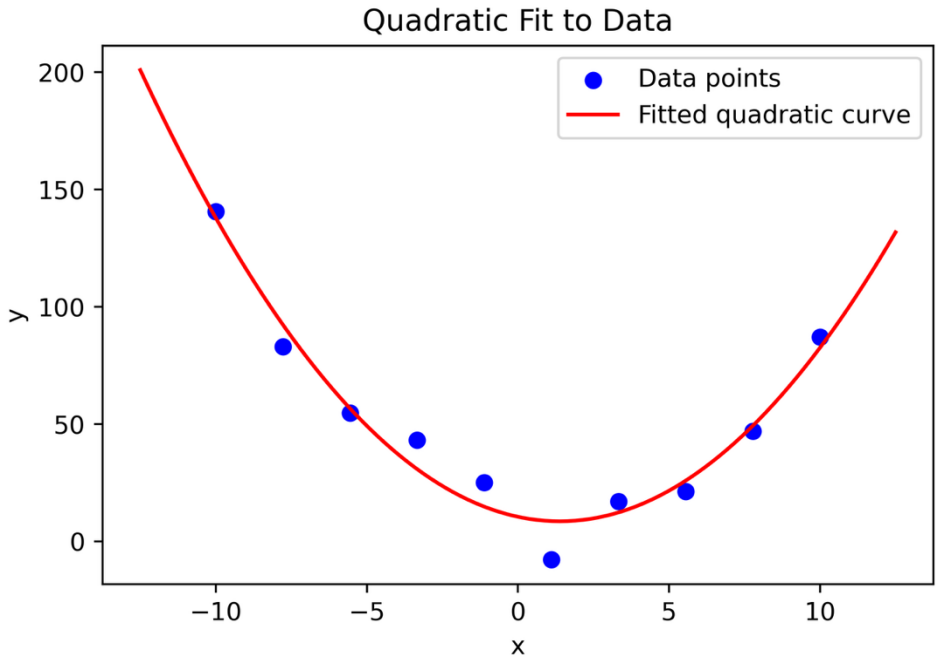
np.random.seed(0)
x = np.linspace(-10, 10, 10)
b2, b1, b0 = 1, -2, 3
y = b2 * x ** 2 + b1 * x + b0
noise = np.random.normal(0, 10, size=x.shape)
y += noise
```

Linear Regression

$$\frac{1}{n} \cdot \sum_{i=1}^{n} (\beta_0 + \beta_1 \cdot x_{i1} + \cdots + \beta_m \cdot x_{im} - y_i)^2$$

Coding

Train a linear regression model to fit the data pairs.
Report the coefficients in your model and the mean squared error (MSE).




```
y = b2 * x**2 + b1 * x + b0
```

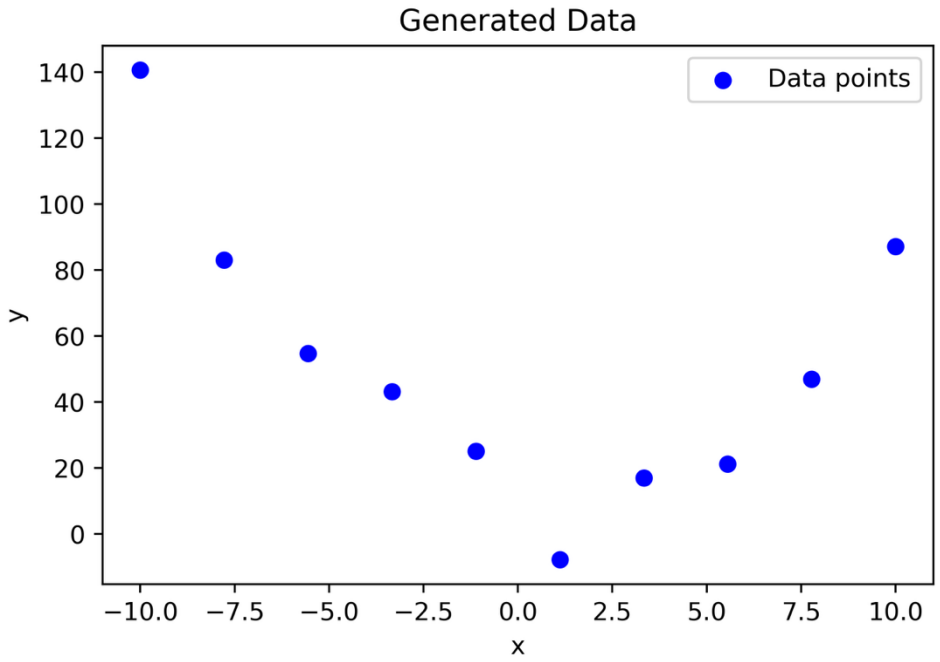
Estimated coeff: 1.00, -2.77, 10.48

MSE: 69.64

Linear Regression

Coding

Now, try a higher-order model, and report the MSE.



2-order: $y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^2$

3-order: $y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^2 + \beta_3 \cdot x^3$

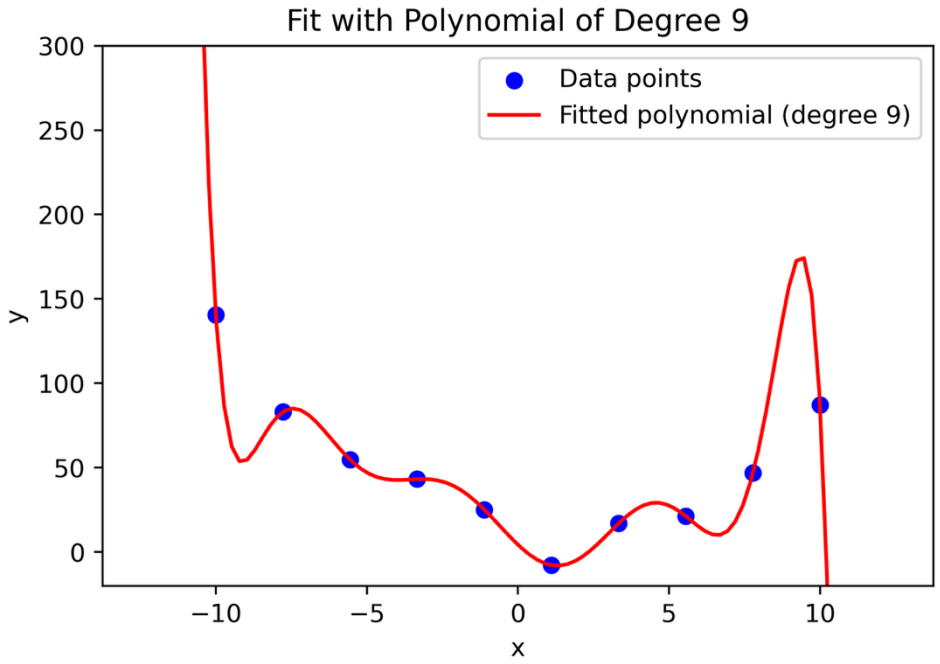
$\vdots$

9-order: $y = \beta_0 + \cdots + \beta_9 \cdot x^9$

Linear Regression

Coding

Now, try a higher-order model, and report the MSE.



2-order: $y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^2$

3-order: $y = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot x^2 + \beta_3 \cdot x^3$

$\vdots$

9-order: $y = \beta_0 + \cdots + \beta_9 \cdot x^9$