2025-09-20 13:52

Status:

Tags:

## Homework 2

110612025　魏于翔

### Part A - Normal equation

(1)

$$X = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}, y = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 5 \\ 4 \end{bmatrix}$$

(2) 首先我們先展開MSE function:

$$\begin{aligned} J(\theta) &= \frac{1}{n}||X\theta - y||_2^2 \\ &= \frac{1}{n}\sum(X\theta - y)^2 \\ &= \frac{1}{n}(X\theta - y)^T(X\theta - y) \\ &= \frac{1}{n}((X\theta)^T - y^T)(X\theta - y) \\ &= \frac{1}{n}(X^T\theta^T X\theta - X\theta y^T - X^T\theta^T y + y^T y) \\ &= \frac{1}{n}(X^T\theta^T X\theta - 2X^T\theta^T y + y^T y) \end{aligned}$$

， `θ` 是 `2*1` `X` 是 `5*2` `y` 是 `5*1` 矩陣

$$(X\theta y^T)^T = X^T\theta^T y = X\theta y^T$$

這是一個純量(`1*1`矩陣)所以上式成立而MSE展開後對`θ`偏微分得到

$$\begin{aligned}\frac{\partial}{\partial\theta}J(\theta) &= \frac{1}{n}(\partial/\partial(x))(X^T\theta^TX\theta - 2X^T\theta^Ty + y^Ty) \\ &= \frac{1}{n}((X^TX + X^TX)\theta - 2X^Ty\theta^T) \\ &= \frac{1}{n}(2X^TX\theta - 2X^Ty) \\ &= 0\end{aligned}$$

移項後得到，$X^TX\theta = X^Ty$

*對於向量偏微分的補充*

**公式**：對於二次型 $\theta^TA\theta$，有`∂/∂θ (θᵀAθ) = (A + Aᵀ)θ`，因此`∂/∂θ (θᵀXᵀXθ) = (XᵀX + XᵀX)θ = 2XᵀXθ`

* 對於線性項 `θᵀb`，有`∂/∂θ (θᵀb) = b`，因此`∂/∂θ (-2θᵀXᵀy) = -2Xᵀy`成立

(3)

$$X^TX = \begin{bmatrix}1 & 1 & 1 & 1 & 1\\ 0 & 1 & 2 & 3 & 4\end{bmatrix}\begin{bmatrix}1 & 0\\ 1 & 1\\ 1 & 2\\ 1 & 3\\ 1 & 4\end{bmatrix} = \begin{bmatrix}5 & 10\\ 10 & 30\end{bmatrix}$$

，而

$$X^Ty = \begin{bmatrix}1 & 1 & 1 & 1 & 1\\ 0 & 1 & 2 & 3 & 4\end{bmatrix}\begin{bmatrix}1\\ 3\\ 2\\ 5\\ 4\end{bmatrix} = \begin{bmatrix}15\\ 38\end{bmatrix}$$

(4) 對Xθ = y使用高斯消去法，得

$$\begin{bmatrix}15\\ 38\end{bmatrix} = \begin{bmatrix}5\theta_0 + 10\theta_1\\ 10\theta_0 + 10\theta_1\end{bmatrix}$$

，根據簡單的二一次方程計算，我們得到

$$\theta_0 = 1.4, \theta_1 = 0.8, \theta = \begin{bmatrix}1.4\\ 0.8\end{bmatrix}$$

(5) 從第四小題得

$$\begin{aligned}\hat{y} &= \theta_0 + \theta_1 x \\ &= 1.4 + 0.8x\end{aligned}$$

## Part B - Prediction & MSE

(1)
根據

$$\hat{y} = 1.4 + 0.8x$$

帶入下表格

| $i$ | $x$ | $y$ | $\hat{y}_i$ | $residual(r_i = y_i - \hat{y}_i)$ |
|---|---|---|---|---|
| 1 | 0 | 1 | 1.4 | -0.4 |
| 2 | 1 | 3 | 2.2 | 0.8 |
| 3 | 2 | 2 | 3 | -1 |
| 4 | 3 | 5 | 3.8 | 1.2 |
| 5 | 4 | 4 | 4.6 | -0.6 |

(2)計算MSE，將(1)表格結果帶進公式

$$\begin{aligned} MSE &= \sum_i r_i^2 \\ &= \frac{1}{5}((-0.4)^2 + (0.8)^2 + (-1)^2 + (1.2)^2 + (-0.6)^2) \\ &= 0.72 \end{aligned}$$

(3)解釋斜率與截距

- 截距(intercept)=1.4:
  - 代表在新型號剛推出時(第0年)，預期會有1.4起事故報告，代表基礎事故率，可能反應了用戶初期不熟悉新車型，或是設計瑕疵
- 斜率(slope)=0.8:
  * 代表隨著使用年限一年年增加，預期事故報告數每年會增加0.3起，可能因為隨著推移，車輛零件老化跟磨損導致事故發生率上升
  (4)根據第一小題的表格我們得知point4有最大的殘差$|r_4| = 1.2$，實際事故發生率遠比預期來的高，可能是因為有特定的設計缺陷在使用3年後開始暴露，或是數據太少，有可能數據多一點就會符合預期了。

### Part C - Gradient Descent & LMS updates

(1)用partA-2的結果推導∇θ J(θ)

$$(X^T X)\theta - X^T y = 0$$

這個公式是在

$$\nabla_\theta J(\theta) = \frac{1}{n}(2X^T X\theta - 2X^T y) = 0$$

的條件下成立的，因此我們可以得到

$$\nabla_\theta J(\theta) = \frac{2}{n}((X^T X)\theta - X^T y) = \frac{2}{n}X^T(X\theta - y)$$

此式成立

(2) Full batch gradient descent(跑過所有的training data後才更新參數): learning rate `α = 0.1` and initial `θ (0) = [0, 0]T`，根據gradient descent公式推導下一個`θ (1)`:

$$\begin{aligned}\nabla_\theta J(\theta) &= \frac{2}{n}X^T(X\theta^{(0)} - y) \\ &= \frac{2}{5}\begin{bmatrix}1 & 1 & 1 & 1 & 1\\ 0 & 1 & 2 & 3 & 4\end{bmatrix}(\begin{bmatrix}1 & 0\\ 1 & 1\\ 1 & 2\\ 1 & 3\\ 1 & 4\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix}) - \begin{bmatrix}1\\3\\2\\5\\4\end{bmatrix}) \\ &= \begin{bmatrix}-6\\-15.2\end{bmatrix}\end{aligned}$$

，帶入gradient descent公式得

$$\begin{aligned}\theta^{(1)} &= \theta^{(0)} - \alpha\nabla J(\theta^{(0)}) \\ &= \begin{bmatrix}0\\0\end{bmatrix} - 0.1\begin{bmatrix}-6\\-15.2\end{bmatrix} \\ &= \begin{bmatrix}0.6\\1.52\end{bmatrix}\end{aligned}$$

(3)LMS/SGD step(跑過一組trainign data就更新一次參數):

- Per-example gradient 公式為:

$$\nabla_\theta l_i = 2(\hat{y}_i - y_i)\bar{x}_i$$

，依題是以i=3(x=2,y=2)跟`θ = [0, 0]T`作為例子，我們先計算

$$\begin{aligned}\bar{x}_i &= [1 \quad x_i]^T \\ &= \begin{bmatrix}1\\2\end{bmatrix}\end{aligned}$$

$$\begin{aligned}\hat{y}_i &= \theta^T \bar{x}_i \\ &= [0 \quad 0]\begin{bmatrix}1\\2\end{bmatrix} \\ &= 0\end{aligned}$$

帶入LMS update公式當α(Learning rate)= 0.1時:

$$\theta^* \leftarrow \theta - \alpha \nabla l_3$$
$$\theta^* = \begin{bmatrix} 0 & 0 \end{bmatrix} - 0.1 \begin{bmatrix} -4 \\ -8 \end{bmatrix}$$
$$= \begin{bmatrix} 0.4 \\ 0.8 \end{bmatrix}$$

(4)

- 先做Full batch gradient:

$$\nabla_\theta J(\theta) = \frac{2}{n} X^T (X\theta^{(1)} - y)$$

- $$= \frac{2}{5} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \end{bmatrix} ( \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} 0.6 \\ 1.52 \end{bmatrix} ) - \begin{bmatrix} 1 \\ 3 \\ 2 \\ 5 \\ 4 \end{bmatrix} )$$
$$= \begin{bmatrix} 1.28 \\ 5.44 \end{bmatrix}$$

，帶入gradient descent 公式:

$$\theta^{(2)} = \theta^{(1)} - \alpha \nabla J(\theta^{(1)})$$
$$= \begin{bmatrix} 0.6 \\ 1.52 \end{bmatrix} - 0.1 \begin{bmatrix} 1.28 \\ 5.44 \end{bmatrix}$$
$$= \begin{bmatrix} 0.472 \\ 0.976 \end{bmatrix}$$

- 再來做LMS/SGD step:取下一個樣本i=4(x=3,y=5)，先計算

$$\bar{x}_i = \begin{bmatrix} 1 & x_i \end{bmatrix}^T$$
$$= \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

$$\hat{y}_i = \theta^T \bar{x}_i$$
$$= \begin{bmatrix} 0.4 & 0.8 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$
$$= 2.8$$

，帶入LMS update公式當α(Learning rate)= 0.1時:

$$\theta^* \leftarrow \theta - \alpha \nabla l_3$$
$$\theta^* = \begin{bmatrix} 0.4 & 0.8 \end{bmatrix} - 0.1 \begin{bmatrix} -4.4 \\ -13.2 \end{bmatrix}$$
$$= \begin{bmatrix} 0.84 \\ 2.12 \end{bmatrix}$$

- Closed-form solution: `θ* = [1.4, 0.8]ᵀ`，計算
  *Parameter error(為了衡量當前參數與最優參數間的距離，反應我們離正確答案有多遠)*
  For full batch:

$$||\theta^{(2)} - \theta||_2 = \sqrt{(0.472 - 1.4)^2 + (0.976 - 0.8)^2} \approx 0.945$$

  *For LMS step:*

$$||\theta^{(2)} - \theta||_2 = \sqrt{(0.84 - 1.4)^2 + (2.12 - 0.8)^2} \approx 1.434$$

  Objective(衡量當前解的好壞程度):
  *For full batch:*

$$\begin{aligned}
J(\theta^{(k)}) &= \frac{1}{n}\sum_i (\theta^{(k)}X - y)^2 \\
&= \frac{1}{n}\sum_i ((0.472 + 0.976X_i) - y)^2 \\
&= \frac{1}{5}((-0.528)^2 + (-1.532)^2 + (0.424)^2 + (-1.6)^2 + (0.376)^2) \\
&= 1.114
\end{aligned}$$

  For LMS step:

$$\begin{aligned}
J(\theta^{(k)}) &= \frac{1}{n}\sum_i (\theta^{(k)}X - y)^2 \\
&= \frac{1}{n}\sum_i ((0.84 + 2.12X_i) - y)^2 \\
&= \frac{1}{5}((1.12)^2 + (-0.04)^2 + (3.08)^2 + (2.2)^2 + (5.32)^2) \\
&= 8.531
\end{aligned}$$

- Gradient norm(衡量當前參數的平坦程度，梯度為0是找到最優解的必要條件，而當梯度越小代表越接近臨界點，也反應參數收斂程度):
- For full batch:

$$\begin{aligned}
\nabla_\theta J(\theta^{(2)}) &= \frac{2}{n}X^T(X\theta^{(2)} - y) \\
&= \frac{2}{5}\begin{bmatrix}1 & 1 & 1 & 1 & 1\\ 0 & 1 & 2 & 3 & 4\end{bmatrix}\left(\begin{bmatrix}1 & 0\\ 1 & 1\\ 1 & 2\\ 1 & 3\\ 1 & 4\end{bmatrix}\begin{bmatrix}0.472\\ 0.976\end{bmatrix}) - \begin{bmatrix}1\\ 3\\ 2\\ 5\\ 4\end{bmatrix}\right) \\
&= \begin{bmatrix}-1.154\\ -1.607\end{bmatrix}
\end{aligned}$$

$$||\nabla J(\theta^{(2)})||_2 = 1.978$$

- For LMS step:

$$\begin{aligned}
\nabla_\theta J(\theta^{(2)}) &= \frac{2}{n} X^T (X\theta^{(2)} - y) \\
&= \frac{2}{5} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \end{bmatrix} ( \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} 0.84 \\ 2.12 \end{bmatrix} ) - \begin{bmatrix} 1 \\ 3 \\ 2 \\ 5 \\ 4 \end{bmatrix} ) \\
&= \begin{bmatrix} 4.16 \\ 13.6 \end{bmatrix} \\
||\nabla J(\theta^{(2)})||_2 &= \sqrt{(4.16)^2 + (13.6)^2} \\
&= 14.22
\end{aligned}$$

- 總結比較:
  - Full batch在三個指標上表現都比較好
  - LMS的目標函數離最優解差距很大，代表預測誤差大
  - LMS的梯度範圍很大，代表離最佳解還有一大段路要走

**Part D - Feature scaling & GD stability**

(1)計算standardized feature

$$\begin{aligned}
z &= \frac{x - \bar{x}}{s_x} \\
\bar{x} &= \frac{0+1+2+3+4}{5} = 2 \\
s_x &= \sqrt{\frac{1}{(n-1)} \sum_i (x_i - \bar{x})^2} \\
&= \sqrt{\frac{1}{4}((-2)^2 + (-1)^2 + 0^2 + 1^2 + 2^2)} \\
&= 1.581 \\
z_1 &= \frac{0-2}{1.581} = -1.265 \\
z_2 &= \frac{1-2}{1.581} = -0.632 \\
z_3 &= 0 \\
z_4 &= \frac{3-2}{1.581} = 0.632 \\
z_5 &= \frac{4-2}{1.581} = 1.265
\end{aligned}$$

(2)重作一次full batch gradient descent

$$
\begin{aligned}
\nabla_\theta J(\theta) &= \frac{2}{n} X^T (X\theta^{(0)} - y) \\
&= \frac{2}{5} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -1.265 & -0.632 & 0 & 0.632 & 1.265 \end{bmatrix} \left( \begin{bmatrix} 1 & -1.265 \\ 1 & -0.632 \\ 1 & 0 \\ 1 & 0.632 \\ 1 & 1.265 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 3 \\ 2 \\ 5 \\ 4 \end{bmatrix} \right) \\
&= \begin{bmatrix} -6 \\ -2.024 \end{bmatrix}
\end{aligned}
$$

帶入gradient descent公式:

$$
\begin{aligned}
\theta_1 &= \theta_0 - \alpha \nabla J(\theta) \\
&= \begin{bmatrix} 0 \\ 0 \end{bmatrix} - 0.1 \begin{bmatrix} -6 \\ -2.042 \end{bmatrix} \\
&= \begin{bmatrix} 0.6 \\ 0.2042 \end{bmatrix}
\end{aligned}
$$

- 用gradient norm來比較了個參數間的好壞
  - Unscaled case:

$$
\begin{aligned}
\nabla J(\theta^{(0)})||_2 &= \sqrt{(-6)^2 + (-15.2)^2} \\
&= 16.34
\end{aligned}
$$

```
* Scaled case:
```

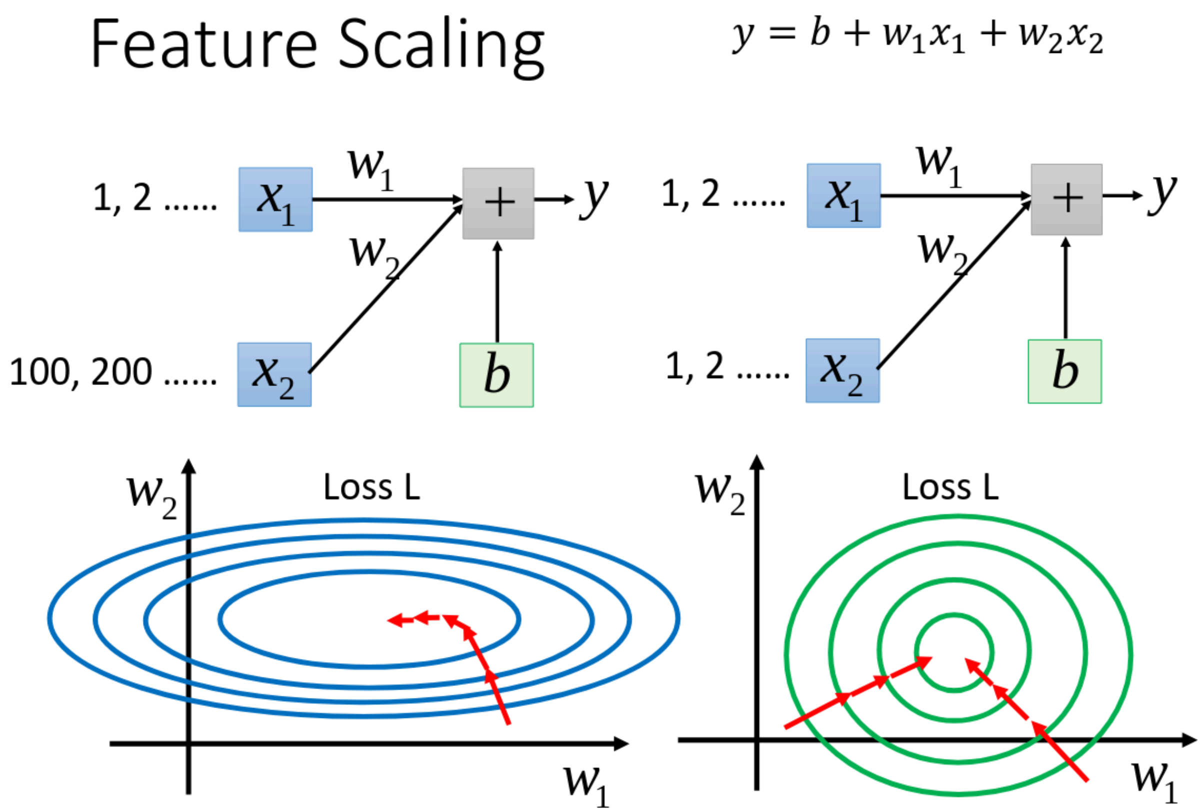
$$
\begin{aligned}
\nabla J(\theta^{(0)})||_2 &= \sqrt{(-6)^2 + (-2.024)^2} \\
&= 6.332
\end{aligned}
$$

- 我們發現梯度明顯減小，代表可以更快的達到最優解。
  (3)

1. 為什麼 scaling features 使得 gradient descent 更快跟跟更穩定?
   Ans:

- 可以避免zig-zag 現象，如下圖所示，當兩個feature單位尺度相差很多時，Loss function會變成一個狹長的橢圓形，較小尺度的那個特徵再有一點小小變化時都會造成loss function有劇烈的變化，很容易因此錯過最佳解，導致gradient的方向要回頭，依此往復循環便會產生鋸齒狀，而feature scaling過後loss function會接近正圓形，各個方向的梯

度更均勻，gradient descent能更直接朝著最優點前進




2. 定義Normalization(rescale to `[0.1]`)跟standardization(mean 0,variance 1)?
   Ans:

- 標準化(Normalization):
  - 將特徵轉成mean為0，variance為1的分佈
  - 公式:

$$z = \frac{x - \mu}{\sigma}$$

  -
  - 數據會呈現標準正態分佈的形狀
- 正規劃(Normalization):
  - 將特徵縮放到 `[0,1]` 區間
  - 公式:

$$x_{norm} = \frac{x - x_{min}}{(x_{max} - x_{min})}$$

  - 所有feature的值都在0到1之間

3. 如果features之間的單位尺度差距很大會有什麼影響跟feaure scaling如何改變？
   Ans:舉例來說，面積`:[50,300]`平方公尺，房間數`[1,5]`個，面積的梯度絕對值遠大於房間數，進而使

- 當學習率太大: 面積方向震盪
- 當學習率太小:臥室方向收斂很慢
  * 而feature scaling就會改變尺度差距大的問題使輸入特徵的梯度相同，更快的達到最佳點。

**Part E - Concept: When to use closed-form vs. gradient methods**

(1)先計算

$$
\begin{aligned}
X^T X &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} \\
&= \begin{bmatrix} 5 & 10 \\ 10 & 30 \end{bmatrix} \\
\det(X^T X) &= 50 > 0
\end{aligned}
$$

代表是invertible的。
(2)計算當all x=2時，

$$
\begin{aligned}
X^T X &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 2 \\ 1 & 2 \\ 1 & 2 \end{bmatrix} \\
&= \begin{bmatrix} 5 & 10 \\ 10 & 20 \end{bmatrix} \\
\det(X^T X) &= 0
\end{aligned}
$$

，如果矩陣invert矩陣的先備條件是**行列式值不等於0**，因此上面那個矩陣不可以invert，而cloded form解:

$$
\theta = X^T y (X^T X)^{-1}
$$
$$
\text{而 } X^T X\text{一定要可以}invert\text{才能求出最佳}\theta
$$

，因此此時closed form equation不能被解出來。
(3)Closed-form跟gradient descent使用時機

- 何時使用closed-form(Normal equation):
  - 當數據集很小時，矩陣儲存不會耗費太多空間
  - 計算(X^T X)^{-1} 的成本可接受，可以得到非常精確的解
- 何時使用gradient descent
  - 當矩陣過大，需要記的元素過多或內存昂貴
  - 無法計算(X^T X)^{-1}
  - 但需要進行參數調整(Learning rate...)

### Part F - Probabilistic interpretation (MLE ↔ least squares)

- Maximum likelihood 以linear regression來舉例就是在預測的線性函數上的每個data放上一個Normal distribution的function，中心是在estimated 的y值上，函數的結果是用來計算實際data對應的高度會在normal distribution上的哪，當預測的y值跟實際的y值靠越近時，結果會越大(越接近1)，反之則越小(越靠近0)，而我們要做的試算出每組data的這個值在相乘，至於要取-log的原因是likelihood數值太小(太接近於0)且為了讓y軸數值是正的。
  (1)對於單個data的likelihood是

$$f(Y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(Y_i-(\theta_0+\theta_1 X_i))}{2\sigma^2}}$$

- 根據獨立性，每個data的likelihood等於分別likelihood相乘

$$\begin{aligned} L(\theta_0,\theta_1,\sigma) &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(Y_i-(\theta_0+\theta_1 X_i))^2}{2\sigma^2}} \\ &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n e^{\sum_{i=1}^{n} \frac{-(Y_i-(\theta_0+\theta_1 X_i))^2}{2\sigma^2}} \\ &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n e^{\sum_{i=1}^{n} \frac{-(Y_i-(\theta_0+\theta_1 X_i))^2}{2\sigma^2}} \end{aligned}$$

  再取log得

$$\log(L(\theta_0,\theta_1,\sigma)) = n\log\frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - (\theta_0+\theta_1)X_i)^2$$

  (2)Negative log-likelihood:

$$
\begin{aligned}
-\log(L(\theta_0,\theta_1,\sigma)) &= -n\log\frac{1}{\sqrt{2\pi\sigma^2}} + \frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i-(\theta_0+\theta_1)X_i)^2 \\
&= constant + \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2 \\
&= constant + \frac{1}{2\sigma^2}SSE
\end{aligned}
$$

(3)Show that maximizing the likelihood over θ is equivalent to minimizing SSE(least square)

$$
\frac{\partial\log(L(\theta_0,\theta_1,\sigma))}{\partial\theta} = -\frac{1}{2\sigma^2}\left(\frac{\partial}{\partial\theta}SSE\right)
$$

根據上式，當likelihood要是maximum時，SSE必然要是minimum。

## Reference


Least square vs maximum likelyhood