Machine Learning HW1

0560206 蔡孝謙

1、Likelihood: $p(t|x,\underline{w},\beta) = \mathcal{N}(t|y(x,\underline{w}),\beta^{-1})$

Prior: $p(\underline{w}|\alpha) = \mathcal{N}(\underline{w}|0,\alpha^{-1}I)$

Posterior: $p(\underline{w}|X,t) \propto p(t|X,\underline{w},\beta)\times p(\underline{w}|\alpha) = \left(\frac{\beta}{2\pi}\right)^{\frac{N}{2}} e^{-\frac{\beta}{2}\sum_{n=1}^{N}(t_n-w^T\phi(x_n))^2}\left(\frac{\alpha}{2\pi}\right)e^{\frac{-\alpha w^T w}{2}}$

$$= C\cdot e^{-\frac{1}{2}\underline{w}^T(\beta\Phi^T\Phi+\alpha I)\underline{w}+\underline{w}^T\beta\Phi^T t} \qquad \Phi = \begin{bmatrix} -\phi(x_1)- \\ -\phi(x_N)- \end{bmatrix}$$

令 $\beta\Phi^T\Phi+\alpha I = S^{-1}$ ⇒ Posterior $= C\cdot e^{-\frac{1}{2}\underline{w}^T S^{-1}\underline{w}+\underline{w}^T\beta\Phi^T t} = C\times N(\underline{w}|S\beta\Phi^T t, S^{-1})$

by Hint $p(t|\underline{w}) = N(t|A\underline{w}+b, L^{-1})$ ⇒ $A=\phi(x)^T$ $\quad b=0 \quad L=\beta$

$p(\underline{w}) = N(\underline{w}|\mu,\Lambda^{-1})$ ⇒ $\mu = S\beta\Phi^T t \quad \Lambda = S^{-1}$

⇒ $p(t) = N(t|\Phi(x)^T S\beta\Phi^T t, \beta^{-1}+\phi(x)^T S\phi(x))$

2. (1.108) $p(x) = e^{(-1+\lambda_1+\lambda_2 x+\lambda_3(x-\mu)^2)}$ $\quad$ (1.109) $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{\{-\frac{(x-\mu)^2}{2\sigma^2}\}}$

$\tilde{H} = L = -\int p(x)\ln p(x)dx + \lambda_1\left(\int p(x)dx-1\right) + \lambda_2\left(\int x\,p(x)dx-\mu\right) + \lambda_3\left(\int (x-\mu)^2 p(x)dx-\sigma^2\right)$

⇒ $p(x) = \underset{p(x)}{\arg\min}\ \tilde{H}$

對 p(x) 微分 ⇒ $\int(-\ln p(x)-1+\lambda_1+\lambda_2 x+\lambda_3(x-\mu)^2)dx = 0$ ⇒ $p(x) = e^{(-1+\lambda_1+\lambda_2 x+\lambda_3(x-\mu)^2)}$

式 1.108 得證

$p(x) = e^{-(\lambda_3 x^2+(\lambda_2-2\lambda_3\mu)x+\lambda_3\mu^2+\lambda_2-\lambda_1)}$

代λ $\begin{cases} \lambda_3 = \frac{1}{2\sigma^2} \\ \lambda_2 = 0 \\ \lambda_1 = 1-\frac{1}{2}\ln(2\pi\sigma^2) \end{cases}$ ⇒ $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(x-\mu)^2}{\sigma^2}}$

3.(hw1_3.m)
(1)
Mathodology
[M=2]
$w = [w0, w1, w2, \ldots, w11, w12, \ldots, w44]^T$
$phi\_x(i) = [1, x1, x2, \ldots, x1^2, x1*x2, \ldots, x4^2]^T$, i=1~400, phi_x is [400 * 21]
$train\_y = [y1, y2, \ldots, y400]^T$, train_y is [400 * 1]
w* = pinv(phi_x'*phi_x)*phi_x'*train_y

[M=3]
$w = [w0, w1, w2, \ldots, w11, w12, \ldots, w44, w111, w112, \ldots, w444]^T$
$phi\_x(i) = [1, x1, \ldots, x1^2, x1*x2, \ldots, x4^2, x1\text{^}3, x1^2*x2, \ldots, , x4\text{^}3]^T$, i=1~400, phi_x is [400 * 85]
$train\_y = [y1, y2, \ldots, y400]^T$, train_y is [400 * 1]
w* = pinv(phi_x'*phi_x)*phi_x'*train_y

Result
[M=2]
- Erms of training set in M=2 is 3.8952
- Erms of testing set in M=2 is 4.1974

[M=3]
- Erms of training set in M=3 is 3.8796
- Erms of testing set in M=3 is 4.1991

在M=3的時候因為feature更多了，所以Erms比較小，但是也沒小很多

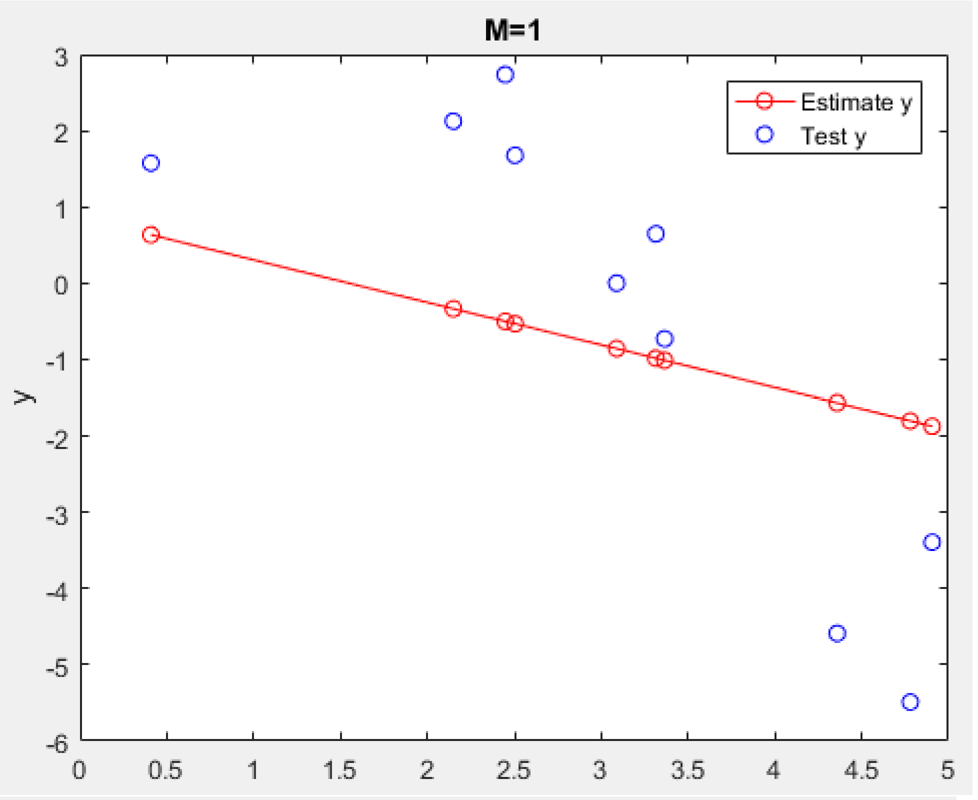
在M=3時如果使用inv或\，而不是pinv的話matlab會出現warning而且Erms會爆掉非常非常多(2.6e+42)，所以我使用pinv

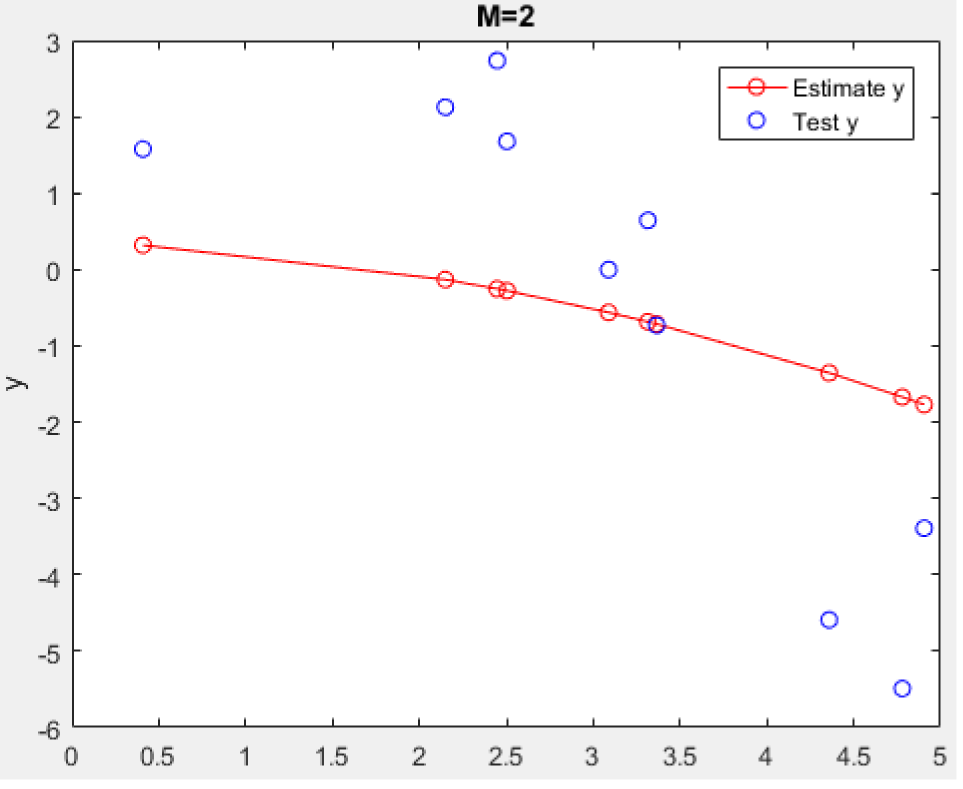
(2)
將每一個attribute拿掉，用另外三個attribute去train，比較training 跟 testing 的Erms，如果比較高，那代表那個欄位影響的比例大

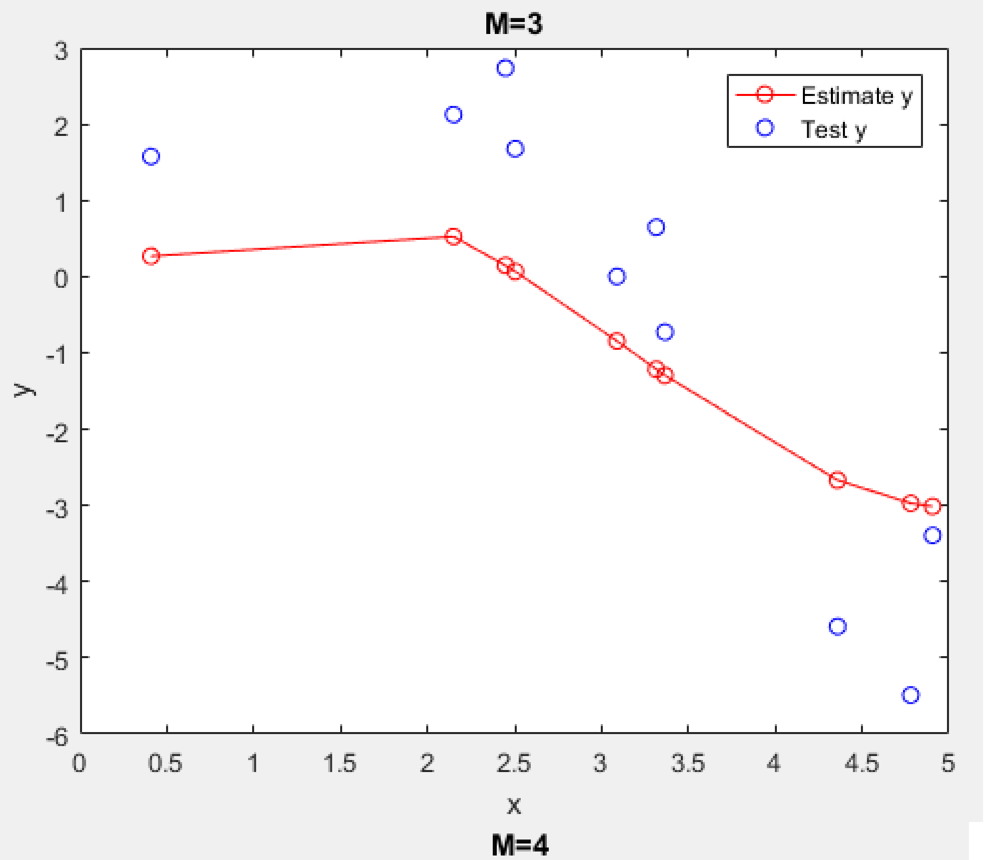
	T	V	AP	RH
Train error	6.892596	4.211406	4.1052	3.986462
Test error	7.293411	4.605487	4.656568	4.350837

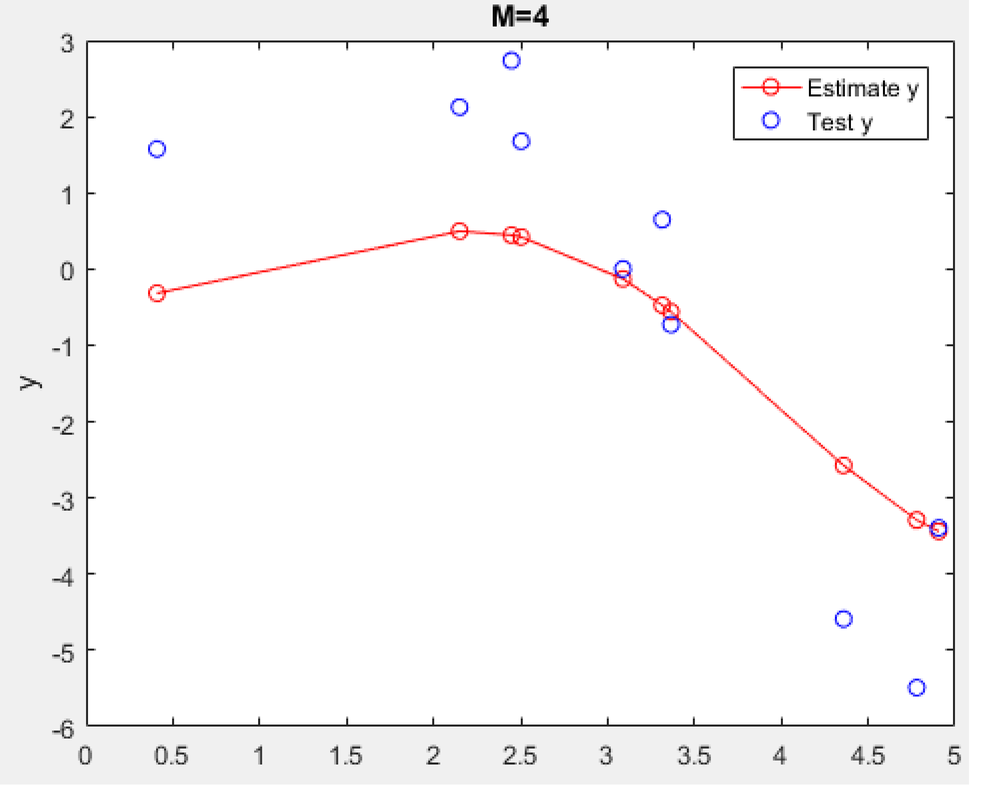
從表格可看出拿掉第一個attribute後的error比較大，故第一個attribute為最contributive的欄位
4.(hw1_4.m)
(a)
將training set分成三組validation set，分別是1-5, 6-10, 11-15，並用另外的10筆資料train出model，其cross validation後的Erms如下表

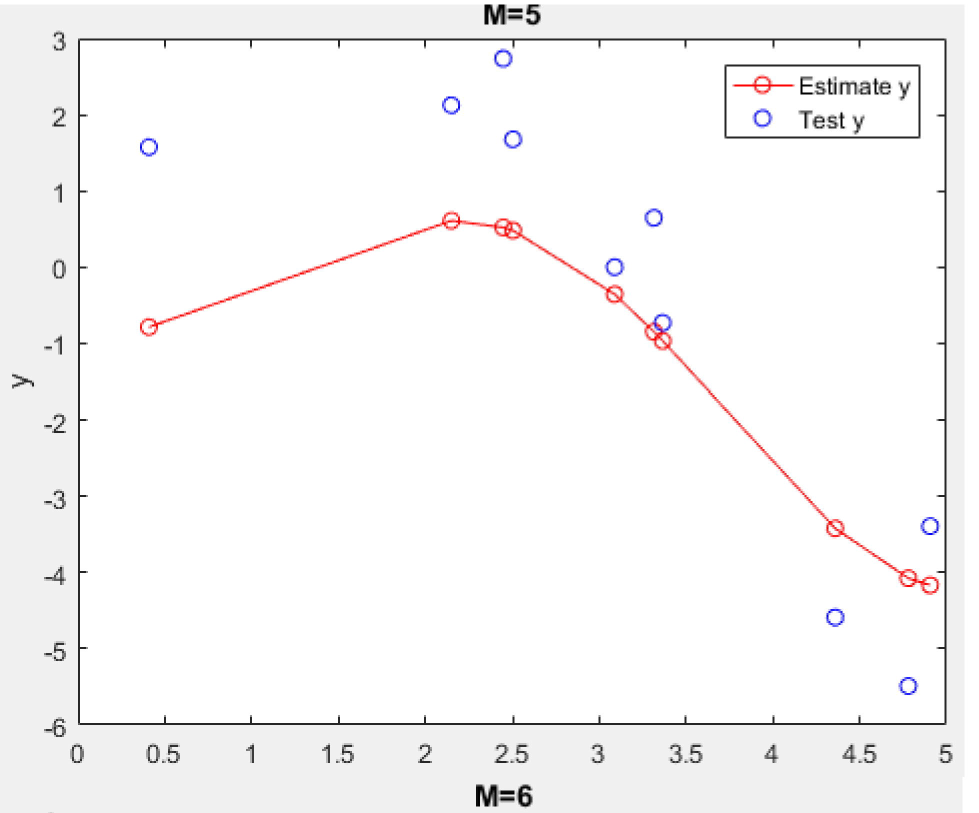
order \ validation set	11-15	6-10	1-5
1	1.63	1.13	0.85
2	1.64	1.83	0.88
3	1.37	1.02	0.64
4	0.98	1.16	0.76
5	1.08	0.69	0.72
6	6.60	23.25	0.85
7	49.72	94.59	0.99
8	204.75	4464.38	0.88
9	185.69	6781.35	18.14

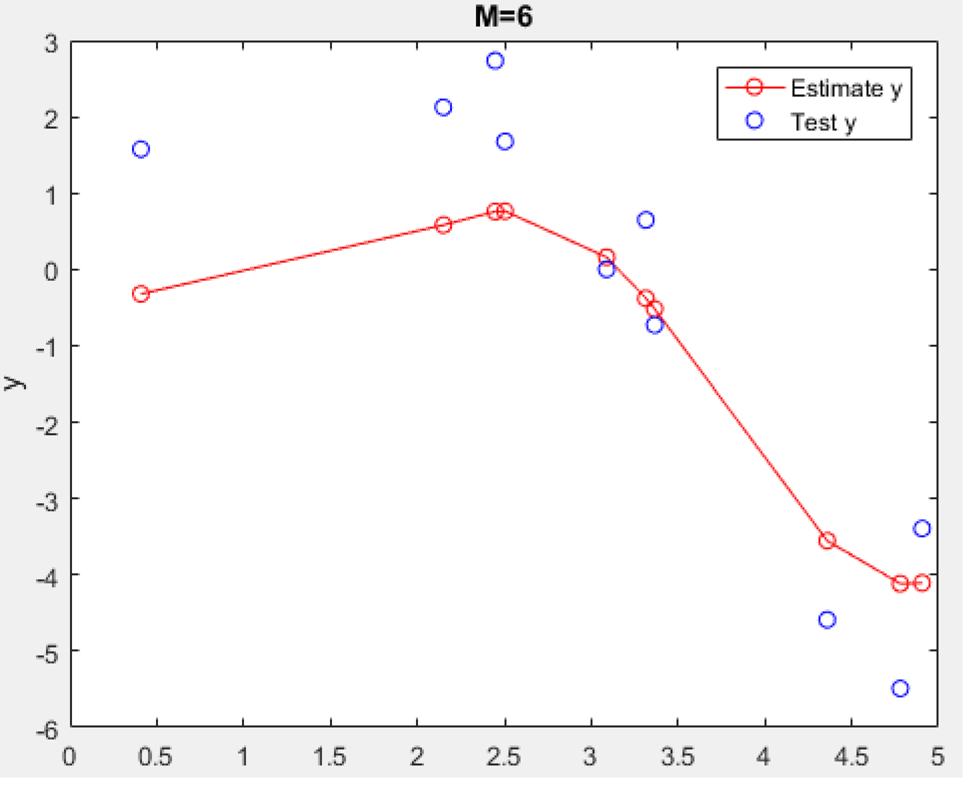

M=1
Estimate y
Test y
y

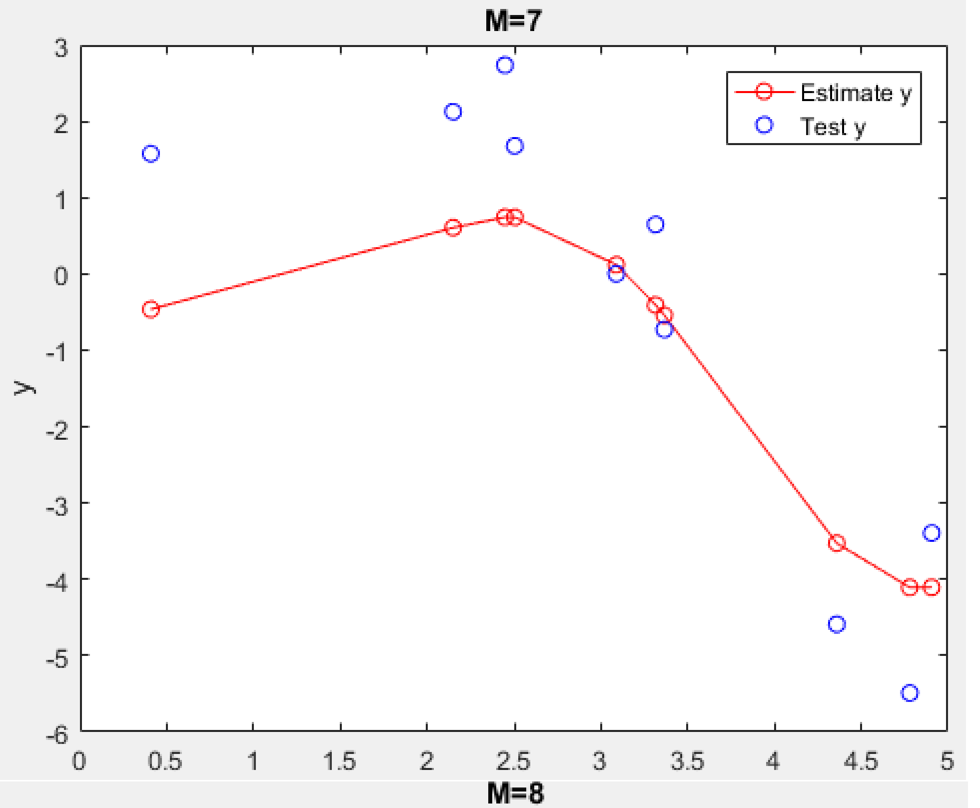

M=2
Estimate y
Test y
y

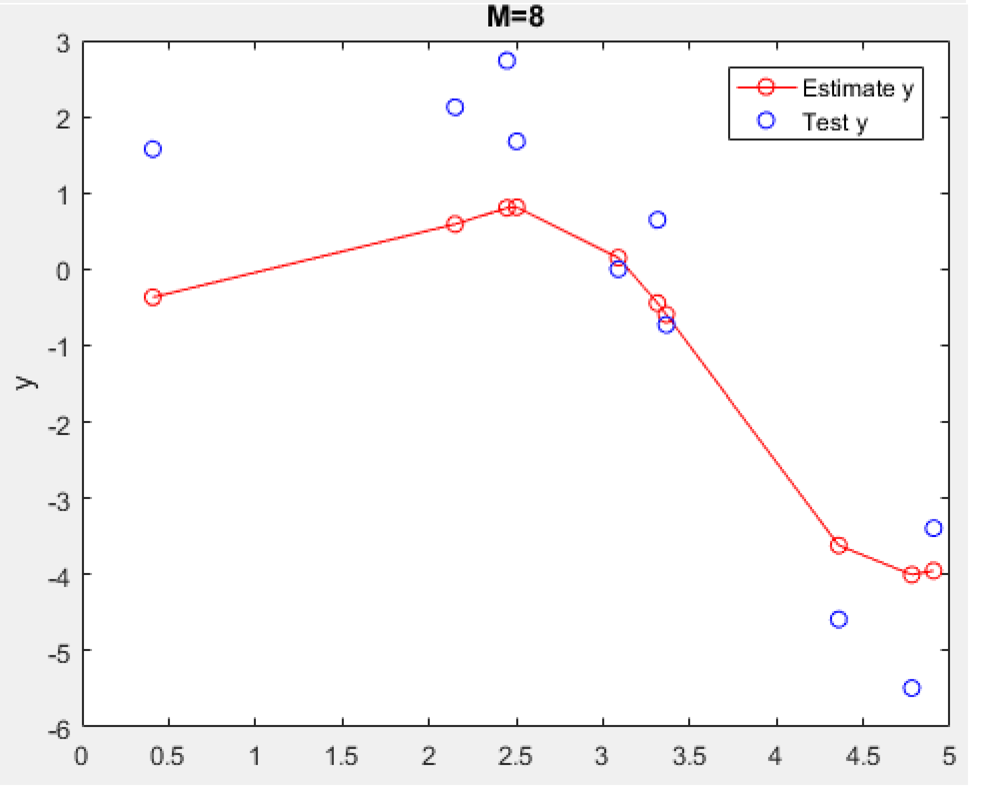

M=3
Estimate y
Test y
y
x
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
3
2
1
0
-1
-2
-3
-4
-5
-6

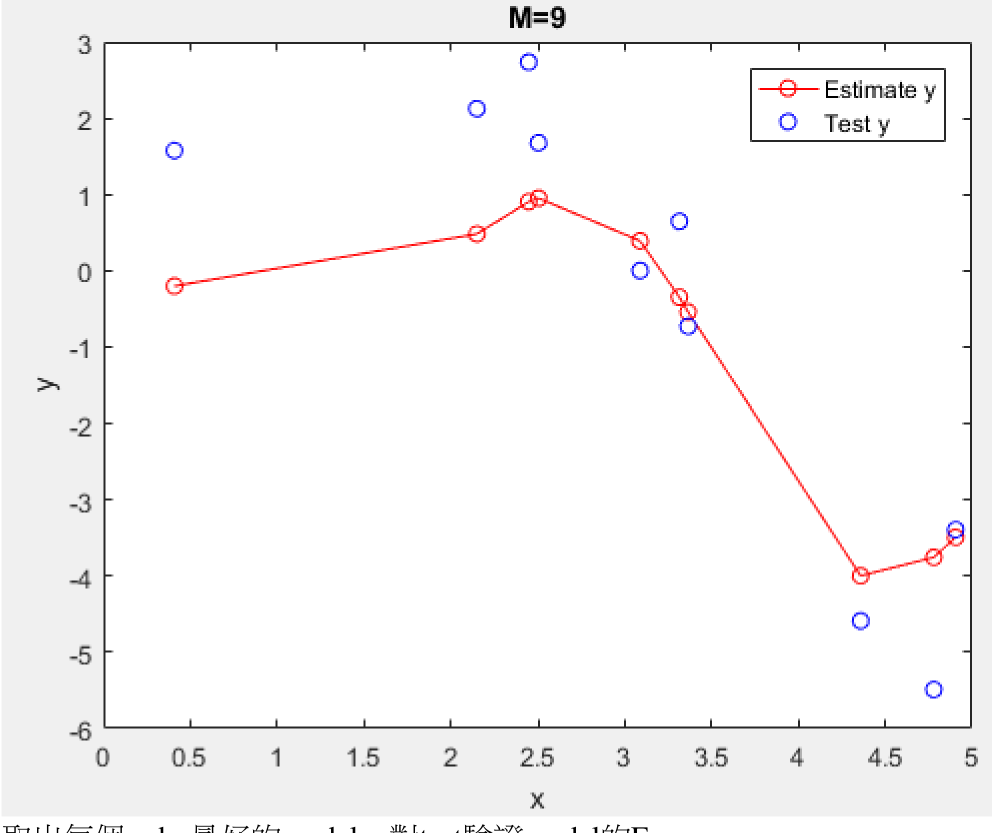

M=4
Estimate y
Test y
y
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
3
2
1
0
-1
-2
-3
-4
-5
-6


M=5
Estimate y
Test y
y
3
2
1
0
-1
-2
-3
-4
-5
-6
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

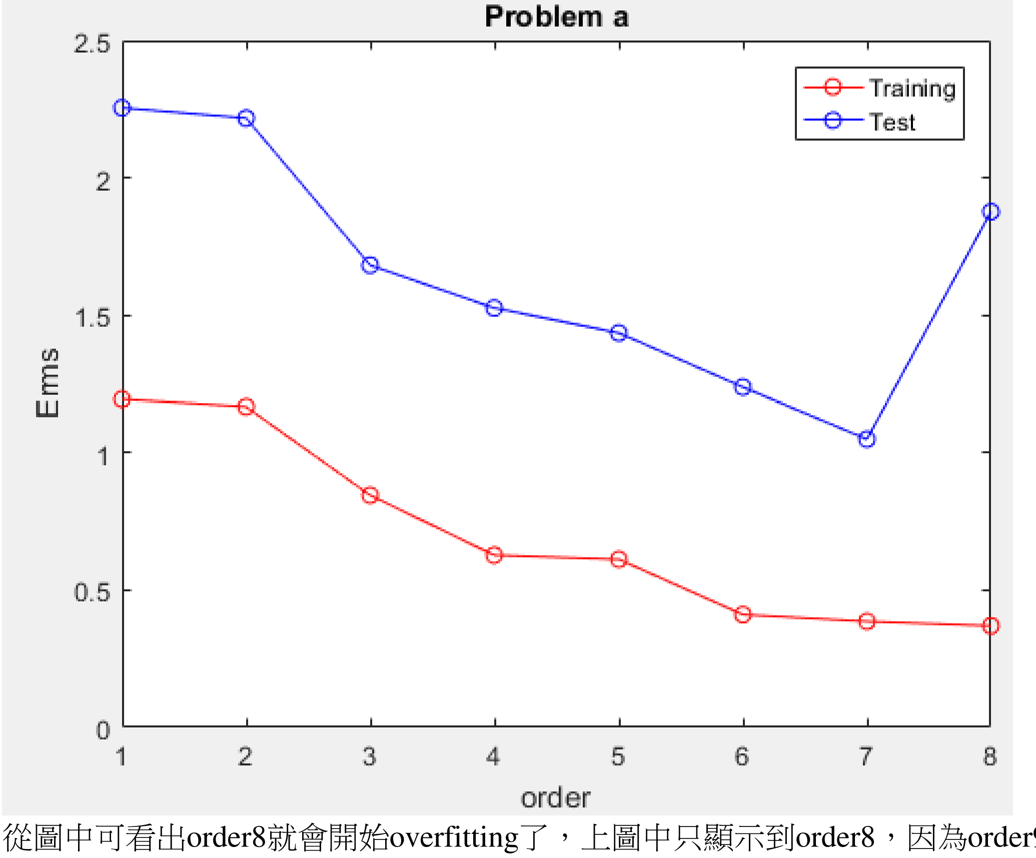

M=6
Estimate y
Test y
y
3
2
1
0
-1
-2
-3
-4
-5
-6
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


M=7
Estimate y
Test y
y

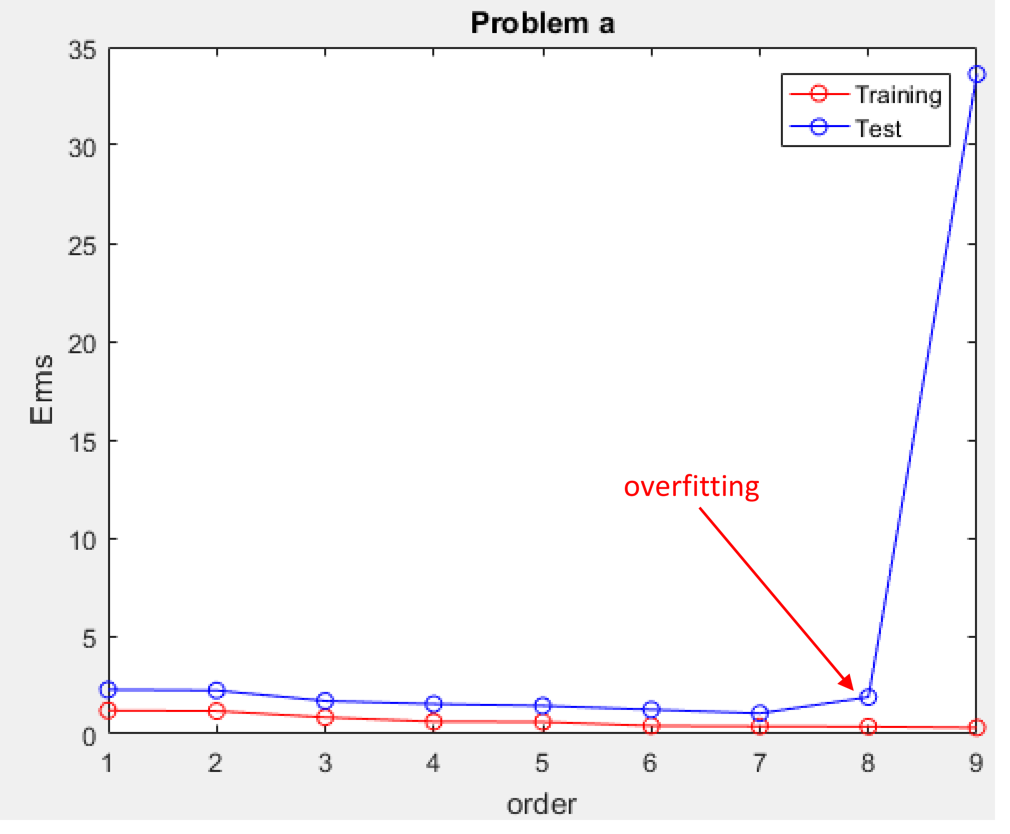

M=8
Estimate y
Test y
y



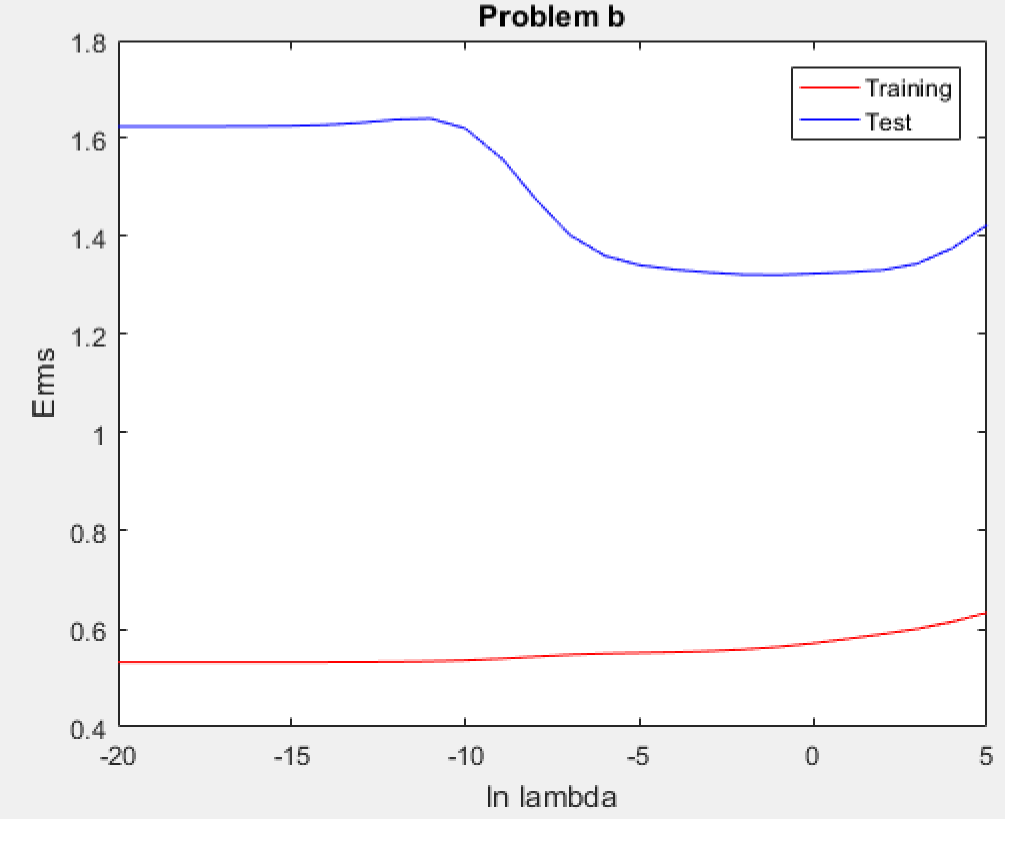
取出每個order最好的model，對test驗證model的Erms



從圖中可看出order8就會開始overfitting了，上圖中只顯示到order8，因為order9會過度overfitting，如下圖



(b)



When lambda = e^(-2), the Testing_Erms-TrainErms has lowest value 0.7416