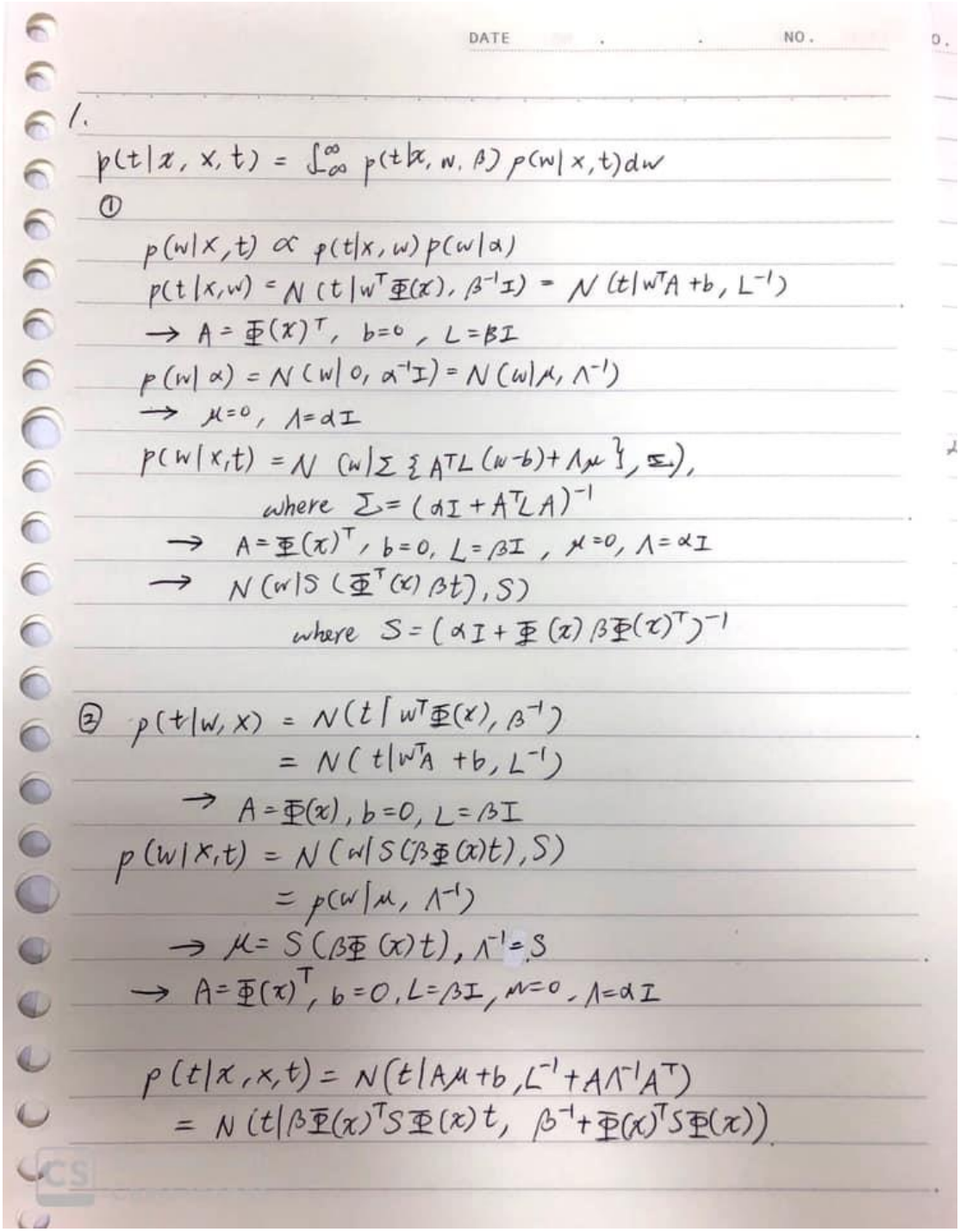
Machine Learning Homework 1

0853412 資管碩 吳宛儒

1.



DATE . . NO.

1.

$p(t|x, X, t) = \int_{-\infty}^{\infty} p(t|x, w, \beta)\, p(w|X, t)\, dw$

①

$p(w|X,t) \propto p(t|X,w)\, p(w|\alpha)$

$p(t|X,w) = N(t|w^T \phi(x), \beta^{-1} I) = N(t|w^T A + b, L^{-1})$

$\rightarrow A = \phi(x)^T,\ b=0,\ L=\beta I$

$p(w|\alpha) = N(w|0, \alpha^{-1} I) = N(w|\mu, \Lambda^{-1})$

$\rightarrow \mu=0,\ \Lambda=\alpha I$

$p(w|X,t) = N(w|\Sigma\{A^T L(w-b)+\Lambda\mu\}, \Sigma)$,

where $\Sigma = (\alpha I + A^T L A)^{-1}$

$\rightarrow A=\phi(x)^T,\ b=0,\ L=\beta I,\ \mu=0,\ \Lambda=\alpha I$

$\rightarrow N(w|S(\phi^T(x)\beta t), S)$

where $S = (\alpha I + \phi(x)\beta\phi(x)^T)^{-1}$

② $p(t|w,X) = N(t|w^T\phi(x), \beta^{-1})$

$= N(t|w^T A + b, L^{-1})$

$\rightarrow A=\phi(x),\ b=0,\ L=\beta I$

$p(w|X,t) = N(w|S(\beta\phi(x)t), S)$

$= p(w|\mu, \Lambda^{-1})$

$\rightarrow \mu = S(\beta\phi(x)t),\ \Lambda^{-1} = S$

$\rightarrow A=\phi(x)^T,\ b=0,\ L=\beta I,\ \mu=0,\ \Lambda=\alpha I$

$p(t|x,X,t) = N(t|A\mu+b, L^{-1}+A\Lambda^{-1}A^T)$

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

2.

Functions defined:

Function Name	meanings
loadData()	讀取資料
splitData(x,t)	將資料切為 train & val
M1Phi(x)	M=1 一階的Φ
M2Phi(x)	M=2 二階的Φ
Weight(phi,t)	由Φ和 t 計算出來的權重
RMSE(pred_ans,x,t)	計算 RMS error
dropFeatures(x,t)	單獨去除 feature 之 RMSerror
getImportant(x)	去除最不重要的一個特徵
GaussianPhi(x)	計算高斯Φ
SigmoidalPhi(x)	計算 sigmoidalΦ
KFold(dataset, i, k)	N-fold cross validation
avgRMSE(x, t, k, M)	平均 RMS error
WeightwithLambda(phi, t, _lambda)	計算帶有 lambda 的權重(Maximum a posteriori approach)
avgRMSE_ld(x, t, k, M)	計算帶有 lambda 以後的 RMS error (Maximum a posteriori approach)

1. Feature selection

(a)

M=1：17 + 1 = 18
M=2：C17 取 2(i,j 和 j,i 不重複) + 17 + 1 = 171

RMS error	Train	Validation
M=1	7.87257	13.8423
M=2	4.93709	17.5567

可以看出即使 M=2 在 training 時會有較低的 RMS error，但在 validation 的時候其 error 較高。較為好奇的是 M=1 的時候 validation 也產出了類似 overfitting 的結果，這部分還有待進一步討論。

(b)

M=1：17 + 1 = 18

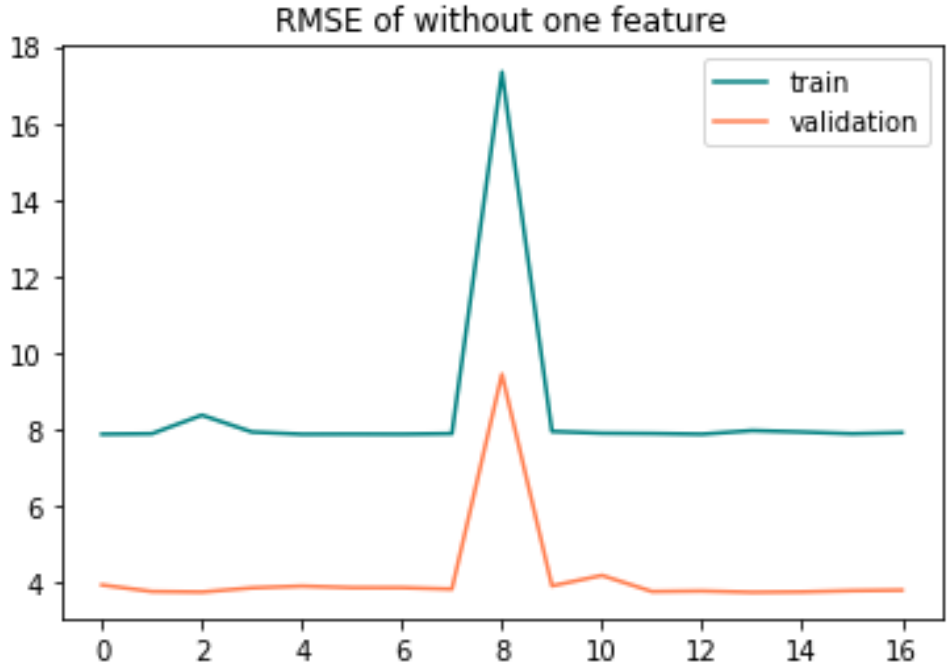
嘗試：每次拿掉一個 feature，看 RMS error 的變化

Training:

拿掉第幾個 feature	RMS error
1	7.87533334
2	7.8876567
3	8.37316749
4	7.93322161
5	7.87397795
6	7.87700168
7	7.87533952
8	7.89043439
9	**17.33946208**
10	7.94730491
11	7.90202827
12	7.89394384
13	7.87563013
14	7.96861402
15	7.93250488
16	7.88779893
17	7.91586493

Validation:

拿掉第幾個 feature	RMS error
1	3.93724
2	3.76602
3	3.75272
4	3.86107
5	3.90678
6	3.86823
7	3.86676
8	3.82958
9	**9.44199**
10	3.9131
11	4.18637
12	3.76605
13	3.77888
14	3.74632
15	3.75603
16	3.78792
17	3.80096



而最小的 RMSerror 發生在：

```
train_idx: [4]
val_idx: [13]
```

以 train 來看，拿掉第五個特徵具有最小的 RMS error，以 validation 來看，拿掉第十四個特徵具有最小的 RMS error。

2. Maximum likelihood approach

(a)

選擇將上一題得出的結果，刪除拿掉以後 RMS error 最小的，也就是相對最不重要的那個 feature，做為接下來的 trainind 和 validation 資料。

a.1 Gaussian

RMS error(Gaussian)	Train	Validation
M=1	6.40391e+08	4.79709e+08
M=2	8.64751629	1941209.41860651

這部分可以看出，Gaussian 所得到的 RMS error 相當不穩定且大，原本考量應該是 function 定義上出了問題，但是幾番確認過後並未找出解決方法與原因。

a.2 Sigmoidal

RMS error(Sigmoidal)	Train	Validation
M=1	2169488.46172904	1666416.15890132
M=2	8.52528087	8.26172646

Sigmoidal 的部分則在 M=1 的時候很不穩定，原本考量應該是 function 定義上出了問題，但是幾番確認過後並未找出解決方法與原因。在 M=2 的時候 error 的數值稍微正常些，training 和 validation 的 error 值則是沒有明顯差距。

(b)

N-fold, N=5

Change hyper-parameter: M=1, M=2 (order)

Average RMS error	Train	Validation
M=1	4.00879	0.432871
M=2	0.282576	180271

經過 N-fold cross validation，M=1 時，validation 意外地得到很低的 RMS error，推測原因應該為切 training 和 validation data 的時候，validation 只佔了 15%，也就是較少筆資料，故可能恰巧變異性較小，較不會有離群值影響結果。而 M=2 時，training 時因為參數量過大而有非常小的 RMS error，但在 validation 時則出現了 error 特大的狀況，發生了 overfitting。

3. Maximum a posteriori approach

(a)

Repeat 2(a), 2(b)

Lambda = 0.001, 1, 2

在這個部份除了重複第二小題所做的 Gaussian 與 Sigmoidal 以外，lambda 的值我也取了 0.001,1,2 三個值來做討論。

a.1 Gaussian

RMS error(Gaussian)	Train	Validation
M=1(Lambda = 0.001)	1.22287e+08	9.29358e+07
M=2(Lambda = 0.001)	**8.90603358**	38364964.7541471
M=1(Lambda = 1)	19936259.31439722	19230854.59143642
M=2(Lambda = 1)	**20.8039414**	70079446.6039978
M=1(Lambda = 2)	8137960.44683446	7984582.73422896
M=2(Lambda = 2)	**23.55557115**	27100378.8194697

這部分可以看到，Gaussian 出來的 error 仍是有許多異常的值，我們先假定異常值是因為函數定義問題而有待解決的問題，除去異常值來看，在 training 且 M=2 的時候所有的 lambda 值得出了較好且正常的結果，而且 lambda 值越小，RMS error 的數值也越小。

a.2 Sigmoidal

RMS error(Sigmoidal)	Train	Validation
M=1(Lambda = 0.001)	2.16668e+06	1.66417e+06
M=2(Lambda = 0.001)	**8.52528223**	**8.26058636**
M=1(Lambda = 1)	1948119.82579022	1489663.60458358
M=2(Lambda = 1)	**8.63328611**	**7.99579985**
M=1(Lambda = 2)	1874376.51722522	1437665.05643117
M=2(Lambda = 2)	**8.77173398**	**7.92097963**

Sigmoidal 的部分則是在 M=2 的時候有較為正常的 RMS error 數值，除去異常值來看，不論 lambda 值為何，training 和 validation 的 error 差異並不大，且在各種 lambda 值之間也並沒有明顯的差異，這是比較出乎意料的部分，目前正在嘗試調整 function 來找出發生這個現象的原因。

Repeat 2(a), 2(b)

Lambda = 0.001, 1, 2

N-fold, N=5

Change hyper-parameter: M=1, M=2 (order)

Average RMS error	Train	Validation
M=1(Lambda = 0.001)	**4.00252**	**0.43832**

M=2(Lambda = 0.001)	0.0912425	207068
M=1(Lambda = 1)	4.48192599	0.49879827
M=2(Lambda = 1)	0.09124249	207067.556
M=1(Lambda = 2)	4.63397866	0.50530394
M=2(Lambda = 2)	0.09124249	207067.55580392

這部份則比上述 Gaussian 與 Sigmoidal 來得較為符合預期。如同前面所做過的不具有 lambda 的實驗，以 training 來看，M=1 時，較小的 lambda 值會帶來稍微較低的 error，M=2 時則並無明顯差異；validation 的部分，在 M=1 時，error 異常地小，推測原因如同上面一題所述，可能是切分資料時造成的天生原因，而在 M=2 時，所有不同 lambda 值都帶來了 overfitting，且 lambda 越大時有 error 越大的趨勢。

綜合上述對於 Maximum a posteriori approach，我們推估較小的 lambda 值(=0.001)可能會有比較好的 RMS error 結果。

(b)

針對 Maximum likelihood approach 與 Maximum a posteriori approach 的實驗，首先發現且致力解決的問題在於 Gaussian 以及 Sigmoidal 出現較為大量的異常值可能的原因，並試著透過與同學討論找出解決辦法。另外在各種實驗中看出了階數(M=1,2)、不同 approach(with or without lambda)和不同 lambda 值的效果，皆於上述幾題中討論。