ML_Hw1

電機乙 0850736 楊登宇

1. Feature selection

(a)

M = 1

這邊我對於 M=1 的實作方法為將 E(w)直接進行微分求極值。推倒過程如下圖(一)：

1) All input data is training data / Curve fitting (Erms = 4.120)

這部分將所有的 data_X 都作為 training data，接著再算全部資料的誤差值。

```
Train data from 1 to 1096
Valid data from 1 to 1096
1
   (a) M = 1 , Erms = 4.120
```

2) 將第 960～1096 筆資料作為 valid data(Erms = 4.944)

縮減 training data 量，可以發現誤差值提升。

```
Train data from 1 to 959
Valid data from 960 to 1096
1
   (a) M = 1 , Erms = 4.944
```

3) 將第 823～1096 筆資料作為 valid data (Erms = 5.805)

繼續縮減 training data 數量，誤差值明顯提升。

```
Train data from 1 to 822
Valid data from 823 to 1096
1
   (a) M = 1 , Erms = 5.805
```

由 2～3 可以確定，當給定的訓練資料量增大時，RMS 的準度會增加是無庸置疑的，只是當資料量一減少，誤差值的差距在 M=1 的模型會有顯著的差異。

M = 2

這裡因為考慮到更高維度的參數眾多(171 個)，直接對 E(w)作微分求極值的方法較為困難。我雖然有先試過直接微分的方式來試著做看看，但參數過多的關係，光是要建立聯立方程式 Ax=b 的 A 矩陣就要計算許久，所以最後選擇直接使用老師上課所提到的方法：

$$W^* = (\Phi^T\Phi)^{-1}\Phi t$$

直接找出Φ矩陣來做運算，對於程式整體的運算效率與速度會更加的優秀，因此決定改用此方法來實作。

而 M = 2 的部分可以看到若是作為 curve fitting 時 RMS 明顯變小，但是當 training data 逐漸變小時，M = 2 的 RMS 變化會比 M = 1 的還要劇烈，可以推測應該是 M = 2 有 overfitting 的現象發生。

1) All input data is training data / Curve fitting (Erms = 3.376)

```
Train data from 1 to 1096
Valid data from 1 to 1096
1
   (a) M = 2 , Erms = 3.376
```

2) 將第 960～1096 筆資料作為 valid data (Erms = 5.996)

```
Train data from 1 to 959
Valid data from 960 to 1096
1
   (a) M = 2 , Erms = 5.996
```

3) 將第 823～1096 筆資料作為 valid data (Erms = 7.513)

```
Train data from 1 to 822
Valid data from 823 to 1096
1
   (a) M = 2 , Erms = 7.513
```

(b)

```
w0      w1      w2      w3      w4      w5      w6      w7      w8      w9      w10     w11     w12     w13     w14     w15     w16     w17
-7.469  0.003   20.410  25.108  -22.874 1.425   1.908   -1.716  0.020   0.419   -0.852  0.052   0.467   -20.568 0.045   -0.041  1.958   -3.681
m0      m1      m2      m3      m4      m5      m6      m7      m8      m9      m10     m11     m12     m13     m14     m15     m16     m17
1.000   23.832  1.847   0.381   0.130   2.730   14.047  16.762  29.111  37.175  0.170   72.855  2.267   1.978   133.130 132.646 2.367   1.835

m0*w0   m1*w1   m2*w2   m3*w3   m4*w4   m5*w5   m6*w6   m7*w7   m8*w8   m9*w9   m10*w10 m11*w11 m12*w12 m13*w13 m14*w14 m15*w15 m16*w16 m17*w17
-7.469  0.078   37.703  9.561   -2.981  3.892   26.803  -28.760 0.596   15.584  -0.145  3.821   1.059   -40.675 5.984   -5.449  4.634   -6.756
mean T = 17.480
```

上圖 w0~w17 為參數 w 向量值，m0~m17 為 17 筆資料的平均值。原先我要進行分析時，只考慮到 w 的部份，所以會認為 w 越大表示此項的重要性越高。但當我將此想法套用到第二題，根據我認為重要的 w 帶入後會發現計算出的 rms 效果不如我想像中的好。所以我又回過頭來看，認為$y(x,w) = W^T\Phi$所預測出的結果是 weight 與 base function 線性組合而成，應該要考量兩者相乘後的數值來做判斷。因此，我將各項資料(x1~x17)的平均值與 weight 相乘，根據這項來作為判斷標準。根據此標準，我認為有以下幾組可能：

1. w2、w3、w4、w5、w6、w7、w9、w11、w12、w13、w14、w15、w16、w17(abs>1)
2. w2、w3、w4、w5、w6、w7、w9、w11、w13、w14、w15、w16、w17(abs>2)
3. w2、w3、w6、w7、w9、w13、w14、w15、w17(abs>5)
4. w2、w6、w7、w9、w13(abs>10)
5. w2、w3、w6、w9、w14(>5)

單就直覺而言，我認為選擇絕對值數值越大的加權項對於整體 model 會越有貢獻，因為當數值越大表示此項只要一有變動便很容易去影響到模型。只是要如何去抉擇 threshold 的部份，我尚且沒有什麼概念，所以我考慮將每項的數值來取絕對值，再根據不同的 threshold 來做分類。另外，考量到或許也有可能此模型只要考慮到正向數值即可，所以另外做一類數值大於 5 的類別。因此，我總共考量 5 種可能。
但是我將此想法落實到第二題來作為我選擇的依據時，會發現並非如此…詳細的討論我將寫在第二題。

2. Maximum likelihood approach

此題我使用的基底函數為高斯分佈模型，根據老師上課所提到的分析，可以得知：

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} X_n$$

$$\sigma_{ML}^2 = \frac{1}{N}\sum_{n=1}^{N} (X_n - \mu_{ML})^T (X_n - \mu_{ML})$$

$$\phi_n(X_n) = exp\left\{-\frac{(X_n - \mu_{ML})^2}{2S^2}\right\}$$

根據上面的方程式，可以快速的計算出 Gaussian maximum likelihood 的平均值與標準差還有基底函數。首先，我考量所有的資料來做分析，可以得到如下圖的結果(hw2-1_Output_.txt)：

```
Fold = 8
Using particular weights: W9 W10 W1 W15 W12 W3 W8 W13 W2 W5 W11 W7 W4

(N = 1)
Train data from 138 to 1096
Valid data from 1 to 137
Gaussian model , Erms = 14.473
Gaussian model with 13 weights, Erms = 15.076

(N = 2)
Train data from 1 to 136 and 275 to 1096
Valid data from 137 to 274
Gaussian model , Erms = 10.788
Gaussian model with 13 weights, Erms = 10.269

(N = 3)
Train data from 1 to 273 and 412 to 1096
Valid data from 274 to 411
Gaussian model , Erms = 8.872
Gaussian model with 13 weights, Erms = 8.820

(N = 4)
Train data from 1 to 410 and 549 to 1096
Valid data from 411 to 548
Gaussian model , Erms = 8.581
Gaussian model with 13 weights, Erms = 8.719
```

```

(N = 5)
Train data from 1 to 547 and 686 to 1096
Valid data from 548 to 685
Gaussian model , Erms = 9.444
Gaussian model with 13 weights, Erms = 8.979

(N = 6)
Train data from 1 to 684 and 823 to 1096
Valid data from 685 to 822
Gaussian model , Erms = 10.113
Gaussian model with 13 weights, Erms = 10.026

(N = 7)
Train data from 1 to 821 and 960 to 1096
Valid data from 822 to 959
Gaussian model , Erms = 8.102
Gaussian model with 13 weights, Erms = 7.778

(N = 8)
Train data from 1 to 958
Valid data from 959 to 1096
Gaussian model , Erms = 8.708
Gaussian model with 13 weights, Erms = 8.723

```

我將 data_X 分為 8 份，來進行 train 與 valid。其中的 particular weights 是我將全部 data 都作為 training data 時，一次取一項資料去除後重新再 train 過來計算 rms，計算每項並比較 rms 變化，若變化越劇烈的，就將其列入 particular weights，我這邊只取到 8.650 作為 threshold。

其中各項數值可參考下表：

original	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
8.646	8.815	8.652	8.666	8.647	8.650	8.647	8.648	8.657	9.109	9.550	8.650	8.680	8.655	8.647	8.685	8.646	8.646

由此可發現，除了 N=1,4,8 外，只取特定 weights 來做預測的話，rms 的數值會較全部資料都拿來做預測的還要好。在此可以推測全部都資料都拿來作為高斯分佈模型的話，會有 overfitting 的可能性。而根據結果來看，在 N=7 時，整體的 rms 都是最小的，在 N=1 時 rms 最大。不過將高斯分布跟第一題的多項式擬合來比較的話，高斯分布的整體 rms 均較大，這個部份我在得到結果時感到滿驚訝的，原先認為高斯模型很常在生活中出現，應該是效果滿好的一組模型，但是得出的結果卻沒有單純用多項式擬合的好。推測可能的原因可能是因為高斯模型較複雜，使得整體擬合結果為 overfitting，導致 rms 較大，但是若是直接將全部的資料都進行訓練再驗證時，rms 仍然比多項式的結果還要大，因此我也不確定到底是哪個部分導致這個模型效果比多項式擬合差的原因。不過使用高斯作為模型時，會發現高斯用有限資料來做預測的效果還是會比多項式擬合來的更好。

接著，回頭來看在第一題所推估的幾類來作為 particular weights 的話，rms 變化狀況：

1) w2、w3、w4、w5、w6、w7、w9、w11、w12、w13、w14、w15、w16、w17(abs>1)

從圖片可以看出，取這幾項對於整體模型並沒有更加改善，反而使得大部分的誤差更大。推測有可能是因為 overfitting 的關係，導致 rms 增加。

```
Fold = 8
Using particular weights: W2 W3 W4 W5 W6 W7 W9 W11 W12 W13 W14 W15 W16 W17

(N = 1)
Train data from 138 to 1096
Valid data from 1 to 137
Gaussian model , Erms = 14.473
Gaussian model with 14 weights, Erms = 14.797

(N = 2)
Train data from 1 to 136 and 275 to 1096
Valid data from 137 to 274
Gaussian model , Erms = 10.788
Gaussian model with 14 weights, Erms = 10.533

(N = 3)
Train data from 1 to 273 and 412 to 1096
Valid data from 274 to 411
Gaussian model , Erms = 8.872
Gaussian model with 14 weights, Erms = 9.670

(N = 4)
Train data from 1 to 410 and 549 to 1096
Valid data from 411 to 548
Gaussian model , Erms = 8.581
Gaussian model with 14 weights, Erms = 9.977
```

```

(N = 5)
Train data from 1 to 547 and 686 to 1096
Valid data from 548 to 685
Gaussian model , Erms = 9.444
Gaussian model with 14 weights, Erms = 9.617

(N = 6)
Train data from 1 to 684 and 823 to 1096
Valid data from 685 to 822
Gaussian model , Erms = 10.113
Gaussian model with 14 weights, Erms = 11.946

(N = 7)
Train data from 1 to 821 and 960 to 1096
Valid data from 822 to 959
Gaussian model , Erms = 8.102
Gaussian model with 14 weights, Erms = 9.479

(N = 8)
Train data from 1 to 958
Valid data from 959 to 1096
Gaussian model , Erms = 8.708
Gaussian model with 14 weights, Erms = 8.927
```

2) w2、w6、w7、w9、w13(abs>10)

這組參數的效果也不好，而且 rms 明顯加大，推測可能是因參數過少的關係，導致整體擬合方程式並沒有非常貼合資料。

```
Fold = 8
Using particular weights: W2 W6 W7 W9 W13

(N = 1)
Train data from 138 to 1096
Valid data from 1 to 137
Gaussian model , Erms = 14.473
Gaussian model with 5 weights, Erms = 18.663

(N = 2)
Train data from 1 to 136 and 275 to 1096
Valid data from 137 to 274
Gaussian model , Erms = 10.788
Gaussian model with 5 weights, Erms = 10.123

(N = 3)
Train data from 1 to 273 and 412 to 1096
Valid data from 274 to 411
Gaussian model , Erms = 8.872
Gaussian model with 5 weights, Erms = 12.995

(N = 4)
Train data from 1 to 410 and 549 to 1096
Valid data from 411 to 548
Gaussian model , Erms = 8.581
Gaussian model with 5 weights, Erms = 13.390
```

```

(N = 5)
Train data from 1 to 547 and 686 to 1096
Valid data from 548 to 685
Gaussian model , Erms = 9.444
Gaussian model with 5 weights, Erms = 6.383

(N = 6)
Train data from 1 to 684 and 823 to 1096
Valid data from 685 to 822
Gaussian model , Erms = 10.113
Gaussian model with 5 weights, Erms = 15.326

(N = 7)
Train data from 1 to 821 and 960 to 1096
Valid data from 822 to 959
Gaussian model , Erms = 8.102
Gaussian model with 5 weights, Erms = 8.376

(N = 8)
Train data from 1 to 958
Valid data from 959 to 1096
Gaussian model , Erms = 8.708
Gaussian model with 5 weights, Erms = 9.853
```

3) w2、w3、w6、w7、w9、w13、w14、w15、w17(abs>5)

比起上一個參數過少的 rms，這組的 rms 明顯下降，但整體 rms 仍然比使用 17 個參數還要高。

```
Fold = 8
Using particular weights: W2 W3 W6 W7 W9 W13 W14 W15 W17

(N = 1)
Train data from 138 to 1096
Valid data from 1 to 137
Gaussian model , Erms = 14.473
Gaussian model with 9 weights, Erms = 15.654

(N = 2)
Train data from 1 to 136 and 275 to 1096
Valid data from 137 to 274
Gaussian model , Erms = 10.788
Gaussian model with 9 weights, Erms = 9.906

(N = 3)
Train data from 1 to 273 and 412 to 1096
Valid data from 274 to 411
Gaussian model , Erms = 8.872
Gaussian model with 9 weights, Erms = 10.252

(N = 4)
Train data from 1 to 410 and 549 to 1096
Valid data from 411 to 548
Gaussian model , Erms = 8.581
Gaussian model with 9 weights, Erms = 10.389
```

```

(N = 5)
Train data from 1 to 547 and 686 to 1096
Valid data from 548 to 685
Gaussian model , Erms = 9.444
Gaussian model with 9 weights, Erms = 8.860

(N = 6)
Train data from 1 to 684 and 823 to 1096
Valid data from 685 to 822
Gaussian model , Erms = 10.113
Gaussian model with 9 weights, Erms = 12.203

(N = 7)
Train data from 1 to 821 and 960 to 1096
Valid data from 822 to 959
Gaussian model , Erms = 8.102
Gaussian model with 9 weights, Erms = 9.561

(N = 8)
Train data from 1 to 958
Valid data from 959 to 1096
Gaussian model , Erms = 8.708
Gaussian model with 9 weights, Erms = 9.225
```

4) w2、w3、w4、w5、w6、w7、w9、w11、w13、w14、w15、w16、w17(abs>2)

參數繼續增加，逐漸貼近使用 17 個參數的 rms。但整體效果仍比不上參數全用。

```
Fold = 8
Using particular weights: W2 W3 W4 W5 W6 W7 W9 W11 W13 W14 W15 W16 W17

(N = 1)
Train data from 138 to 1096
Valid data from 1 to 137
Gaussian model , Erms = 14.473
Gaussian model with 13 weights, Erms = 14.795

(N = 2)
Train data from 1 to 136 and 275 to 1096
Valid data from 137 to 274
Gaussian model , Erms = 10.788
Gaussian model with 13 weights, Erms = 10.519

(N = 3)
Train data from 1 to 273 and 412 to 1096
Valid data from 274 to 411
Gaussian model , Erms = 8.872
Gaussian model with 13 weights, Erms = 9.615

(N = 4)
Train data from 1 to 410 and 549 to 1096
Valid data from 411 to 548
Gaussian model , Erms = 8.581
Gaussian model with 13 weights, Erms = 9.993
```

```

(N = 5)
Train data from 1 to 547 and 686 to 1096
Valid data from 548 to 685
Gaussian model , Erms = 9.444
Gaussian model with 13 weights, Erms = 9.569

(N = 6)
Train data from 1 to 684 and 823 to 1096
Valid data from 685 to 822
Gaussian model , Erms = 10.113
Gaussian model with 13 weights, Erms = 11.919

(N = 7)
Train data from 1 to 821 and 960 to 1096
Valid data from 822 to 959
Gaussian model , Erms = 8.102
Gaussian model with 13 weights, Erms = 9.469

(N = 8)
Train data from 1 to 958
Valid data from 959 to 1096
Gaussian model , Erms = 8.708
Gaussian model with 13 weights, Erms = 8.926
```

5) w2、w3、w6、w9、w14(>5)

比起第三組的結果，此組參數量少但是 rms 貼近的程度卻更好。可以推測此模型取參數為正的效果是可行的。

```
Fold = 8
Using particular weights: W2 W3 W6 W9 W14

(N = 1)
Train data from 138 to 1096
Valid data from 1 to 137
Gaussian model , Erms = 14.473
Gaussian model with 5 weights, Erms = 16.709

(N = 2)
Train data from 1 to 136 and 275 to 1096
Valid data from 137 to 274
Gaussian model , Erms = 10.788
Gaussian model with 5 weights, Erms = 8.991

(N = 3)
Train data from 1 to 273 and 412 to 1096
Valid data from 274 to 411
Gaussian model , Erms = 8.872
Gaussian model with 5 weights, Erms = 10.571

(N = 4)
Train data from 1 to 410 and 549 to 1096
Valid data from 411 to 548
Gaussian model , Erms = 8.581
Gaussian model with 5 weights, Erms = 10.565
```

```

(N = 5)
Train data from 1 to 547 and 686 to 1096
Valid data from 548 to 685
Gaussian model , Erms = 9.444
Gaussian model with 5 weights, Erms = 8.494

(N = 6)
Train data from 1 to 684 and 823 to 1096
Valid data from 685 to 822
Gaussian model , Erms = 10.113
Gaussian model with 5 weights, Erms = 12.049

(N = 7)
Train data from 1 to 821 and 960 to 1096
Valid data from 822 to 959
Gaussian model , Erms = 8.102
Gaussian model with 5 weights, Erms = 9.772

(N = 8)
Train data from 1 to 958
Valid data from 959 to 1096
Gaussian model , Erms = 8.708
Gaussian model with 5 weights, Erms = 9.379
```

3. Maximum a posteriori approach

$$w_{MAP}^{*} = (\Phi^{T}\Phi + \lambda \mathrm{I})^{-1}\Phi t \,, \lambda = \frac{\alpha}{\beta}$$

$$assume \;\; \alpha = 1 \,, \beta^{-1} = \beta_{ML}^{-1} = \frac{1}{N}\sum_{n=1}^{N}\{y(X_n, W_{ML}) - t_n\}^2$$

根據上述方法，將原先高斯 maximum likelihood 的結果結合超參數 λ，以計算 MAP 的擬合方程式。下圖為使用 MAP 方法所計算出的 rms(hw3-1_Output.txt)：

```
Fold = 8

(N = 1)
Train data from 138 to 1096
Valid data from 1 to 137
Gaussian model with MAP (λ=95.589), Erms = 15.095

(N = 2)
Train data from 1 to 136 and 275 to 1096
Valid data from 137 to 274
Gaussian model with MAP (λ=93.760), Erms = 9.237

(N = 3)
Train data from 1 to 273 and 412 to 1096
Valid data from 274 to 411
Gaussian model with MAP (λ=95.778), Erms = 9.950

(N = 4)
Train data from 1 to 410 and 549 to 1096
Valid data from 411 to 548
Gaussian model with MAP (λ=92.094), Erms = 9.525

(N = 5)
Train data from 1 to 547 and 686 to 1096
Valid data from 548 to 685
Gaussian model with MAP (λ=90.643), Erms = 8.019

(N = 6)
Train data from 1 to 684 and 823 to 1096
Valid data from 685 to 822
Gaussian model with MAP (λ=87.861), Erms = 11.716

(N = 7)
Train data from 1 to 821 and 960 to 1096
Valid data from 822 to 959
Gaussian model with MAP (λ=84.238), Erms = 7.647

(N = 8)
Train data from 1 to 958
Valid data from 959 to 1096
Gaussian model with MAP (λ=76.197), Erms = 8.191
```

跟使用 maximum likelihood 的結果相比，多加上一個超參數的方法似乎沒有減少多少誤差，反而多增加了些許誤差。除了少數幾個特例可以看出有好一點點外，幾乎是沒有甚麼影響的。我推測可能是因為我的α取值的關係，因此我反覆又測試另α=0.8,0.6,0.4,0.2，可以驗證到當α趨近於 0 時，結果會趨近 maximum likelihood 的方法，但是在α=0.4 時，整體的 rms 會最佳，結果會比第二題使用高斯分布且參數全部使用的 rms 還要低一些，結果如下圖(hw3-1_Output_a=0.4.txt)：

```
Fold = 8

(N = 1)
Train data from 138 to 1096
Valid data from 1 to 137
Gaussian model with MAP (λ=38.236), Erms = 14.772

(N = 2)
Train data from 1 to 136 and 275 to 1096
Valid data from 137 to 274
Gaussian model with MAP (λ=37.504), Erms = 9.698

(N = 3)
Train data from 1 to 273 and 412 to 1096
Valid data from 274 to 411
Gaussian model with MAP (λ=38.311), Erms = 9.520

(N = 4)
Train data from 1 to 410 and 549 to 1096
Valid data from 411 to 548
Gaussian model with MAP (λ=36.838), Erms = 8.996

(N = 5)
Train data from 1 to 547 and 686 to 1096
Valid data from 548 to 685
Gaussian model with MAP (λ=36.257), Erms = 8.454

(N = 6)
Train data from 1 to 684 and 823 to 1096
Valid data from 685 to 822
Gaussian model with MAP (λ=35.144), Erms = 11.001

(N = 7)
Train data from 1 to 821 and 960 to 1096
Valid data from 822 to 959
Gaussian model with MAP (λ=33.695), Erms = 7.725

(N = 8)
Train data from 1 to 958
Valid data from 959 to 1096
Gaussian model with MAP (λ=30.479), Erms = 8.253
```

透過調整α值，的確可以使得整體 rms 更加精準，但並不是將α不斷縮小即可，當α趨近於 0 時會讓結果趨近 maximum likelihood 但卻不是得到最佳的參數，因此找α的過程變成一個相當重要的關鍵。