MLHW1_0753748_董律里

2.1.(a)
Please evaluate the corresponding RMS error on the training set and validation set.
In the feature selection stage, please apply polynomials of order M = 1 and M = 2 over the dimension D = 17 of input data.

定義我們的多項式函數：

$$Poly\ (M=2)\ :\ y(x,w) = \omega_0 + \sum_{i=1}^{D} \omega_i^2 x_i^2 + \sum_{i=1}^{D} \sum_{j=1}^{D} x_{ij} x_i x_j$$

$$Error\ :\ E(w) = \frac{1}{2N} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}$$

PHi(x) :

$$\Phi_n j = \phi(x_n) = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & ... & \phi_{M-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & ... & \phi_{M-1}(x_2) \\ . & . & ... & . \\ . & . & ... & . \\ \phi_0(x_N) & \phi_1(x_N) & ... & \phi_{M-1}(x_N) \end{bmatrix}$$

$$W_{ML} = (\Phi^\intercal \Phi)^{-1} \Phi^\intercal Y$$

定義所需要的函數：
1.切割函數：split（n）
2.PHi函數：phi(data, m, n)
3.估計函數（係數估計和RMSE）： estimate(phi(), target)

接下來分別計算M=2、M=1的train & cv之phi值以及RMSE，其中CV選擇5組

1. M=2

	train_RMSE	cv_RMSE
0	[3.3303181800916346]	[4.189775058922326]
0	[3.3093008606901155]	[4.600817032024187]
0	[3.1960605259660264]	[4.6990249675649896]
0	[3.340694611128524]	[4.775224042428947]
0	[3.1859276161398653]	[5.502359841588144]

可以看到train的RMSE介在[3.15 , 3.35]之間，但是cv的RMSE卻明顯比train的部分來得大，表示我們的模型可能存在著over-fitting的問題

2. M=1

	train_RMSE	cv_RMSE
1	4.010777	4.541331
2	4.089115	4.237307
3	4.100977	4.157073
4	4.141310	4.006769
5	4.147419	3.965697

可以看到在M=1的情況下，雖然train的RMSE比M=2的時候高，但cv的結果卻與train一致，因此不存在over-fitting的問題

2.1(b)

Please analyze the weights of polynomial models for M = 1 and select the most contributive attribute which has the lowest RMS error on the Training Dataset.

- 根據a小題的結果我們將最小cv_RMSE的那一組係數取出來

	train_RMSE	cv_RMSE
1	4.010777	4.541331
2	4.089115	4.237307
3	4.100977	4.157073
4	4.141310	4.006769
5	4.147419	3.965697

cv_RMSE平均最小的s為第5組，RMSE = 3.965697415702034

- 係數為:

```
[[-2.54950233e+01]
 [ 4.01963036e-02]
 [ 2.59312214e+01]
 [ 2.11421770e+01]
 [-2.56897128e+01]
 [ 1.35691812e+00]
 [ 1.89115549e+00]
 [-1.68807135e+00]
 [ 2.05822229e-02]
 [ 4.18774094e-01]
 [-9.77538251e-01]
 [ 6.95351852e-02]
 [ 3.87195305e-01]
 [-1.68751089e+01]
 [ 3.75798039e-02]
 [-3.16457581e-02]
 [ 1.79422061e+00]
 [-3.33245043e+00]]
```

2.2
(a) Choose some of air quality measurement in datasetX.csv and design your model.

- 選擇Gaussian basis function，並挑選11個解釋變數進行分析
  變數為:['AMB_TEMP'，'CH4'，'CO'，'NMHC'，'NO'，'NO2'，'NOx', 'O3'，'PM10'，'RAINFALL'，'RH']
- 定義Gaussian basis function 和 design matrix

$$Gaussian : \phi_j(x) = exp(\frac{(x-\mu_j)^2}{2S^2})$$

- 係數估計以及RMSE：

```
[array([[ 11.13008978],
        [  4.32191387],
        [-10.89412996],
        [ 20.65122188],
        [ -1.19395632],
        [  1.62555175],
        [ -1.383968  ],
        [  1.50848435],
        [  0.61093422],
        [  6.69402259],
        [  1.97055026]]), array([10.75947448])]
```

(b) Apply N-fold cross-validation in your training stage to select at least one hyperparameter

- 我所選擇的hyperparameter為 gaussian fuction裡面的S(sigma)，cv一樣為5份
- S 的範圍為 0.1 ~ 0.5 ， 總共測試5個s，透過上面定義的函數來計算 以及各自的RMSE ， 並挑選出最佳的S = 0.1

```
     cv_RMSE
1   8.999457
2   9.035532
3   9.089470
4   9.299276
5  10.592466
RMSE平均最小的s為0.1，RMSE = 8.999457309102066
```

3. Maximum a posteriori(MAP) approach

Use maximum a posteriori approach method and repeat 2 . You could choose Gaussian distribution as a prior.

(a)

Gaussian noise model :

$$\epsilon \sim N(0,\ \beta)$$

透過以下的公式更新我們的參數:

$$p(w|t) = N(w|m_N,\ S_N)\ \text{, where}$$

$$S_N^{-1} = S_0^{-1} + \beta\Phi^T\Phi$$

$$m_N = S_N(S_0^{-1}m_0 + \beta\Phi^T t)$$

建立我們起始的參數:

$$m_0 = \begin{bmatrix} 0 & 0 & ..... & 0 \end{bmatrix}$$

$$S_0 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

$$\beta = 0$$

定義posterior函數：P(w| t) , 並將資料切成100份

接著是 MAP approach ， 並計算出RMSE

$$RMSE = 9.79628501$$

(b)測試不同的S：(0.1 ,0.2 ,0.3 ,0.4, 0.5)並計算RMSE

測試結果

S	RMSE
0.1	10.851152076017991
0.2	9.796285010886661
0.3	9.748188501220547
0.4	9.784211942465895
0.5	9.810886684446048

S = 0.3時，有最小的RMSE = 9.748188501220547

我們發現在MAP的結果中RMSE比MLE的結果來的大一些，但是係數的部分MAP的結果數值較小