Machine Learning (HW01)

Course： NCTU-ECM5094-ML

***ID：**309505002 ***Name：**鄭紹文

Q1：Bayesian Linear Regression

1. Bayesian Linear Regression.

(1). Linearity is often a good assumption when many inputs influences the output. Some natural law are approximately linear $f = ma$, but in general, it's rather likely that a true function is linear. The simplest linear model for regression is one that involves a linear combination of the input variables

$$y(x, w) = w_0 + w_1 x_1 + \dots + w_D x_D, \quad \text{with } x = (x_1, \dots, x_D)^T$$ known as linear regression

we immediately extend the class of models by considering linear combinations of fixed nonlinear functions of the input variables

$$y(x, w) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(x).$$

functions $\phi_j(x)$ of the input x are known as basis function. *

(2)

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

其中 $\begin{cases} p(t|x, \vec{w}) = \mathcal{N}(t|y(x, \vec{w}), \beta^{-1}) = \mathcal{N}(t|w^T \phi(x), \beta^{-1}) \\ p(w) = \mathcal{N}(w|0, \alpha^{-1} I) \end{cases}$

$$\because p(\vec{w}|\vec{x}, \vec{t}) \propto p(t|\vec{x}, \vec{w}) \times p(\vec{w}) \propto \prod_{n=1}^{N} \mathcal{N}(t_n|w^T \phi(x_n), \beta^{-1}) \cdot \mathcal{N}(w|0, \alpha^{-1} I)$$

$$\propto \exp\left[-\frac{\beta}{2}\left((t_1 - w^T\phi(x_1))^2 + (t_2 - w^T\phi(x_2))^2 + \dots + (t_N - w^T\phi(x_N))^2\right)\right] \exp\left(\frac{-\alpha}{2} w^T w\right)$$

$$= \exp\left[\frac{-\beta}{2} \sum_{n=1}^{N} \left(t_n^2 + w^T \phi(x_n)\phi(x_n)^T w - 2 w^T \phi(x_n) t_n\right) - \frac{\alpha}{2} w^T w\right]$$

$$\propto \frac{-1}{2} w^T \left(\beta \sum_{n=1}^{N} \phi(x_n)\phi^T(x_n) + \alpha I\right) w - 2\beta w^T \sum_{n=1}^{N} (\phi(x_n) t_n)$$

$$\Rightarrow S_N^{-1} = \alpha I + \beta \sum_{n=1}^{N} \phi(x_n)\phi^T(x_n), \quad m_N = S\left(\sum_{n=1}^{N} \beta \phi(x_n) t_n\right)$$

$$\therefore p(\vec{w}|\vec{x}, \vec{t}) = \mathcal{N}(w|m_N, S_N)$$

$$p(t|x, \vec{x}, \vec{t}) = \int_{-\infty}^{\infty} p(t|x, \vec{w}) p(\vec{w}|\vec{x}, \vec{t}) \, dw \propto \int_{-\infty}^{\infty} \exp\left(\frac{-\beta}{2}(t - w^T\phi(x))^2\right) \cdot \exp\left(\frac{-1}{2}(w - m_N)^T S_N^{-1} (w - m_N)\right)$$

$$\propto \int_{-\infty}^{\infty} \exp\left(\frac{-\beta}{2}\left(t^2 - 2(w^T\phi(x))t + (w^T\phi(x))^2\right]\right) \cdot \exp\left[\frac{-1}{2}\left(w^T S_N^{-1} w - 2 w^T S_N^{-1} m_N + m_N^T S_N^{-1} m_N\right)\right) dw$$

$$\propto \int_{-\infty}^{\infty} \exp\left[\frac{-1}{2}\left(\beta t^2 - 2\beta w^T \phi(x) t + \beta w^T \phi(x) \phi^T(x) w\right) + (w^T S_N^{-1} w) - (2 w^T S_N^{-1} m_N)\right] dw$$

$$= \int_{-\infty}^{\infty} \exp\left\{\frac{-1}{2}\left[w^T(\beta\phi(x)\phi^T(x) + S_N^{-1}) w - 2w^T(\phi(x) t\beta + S_N^{-1} m_N) + \beta t^2\right]\right\} dw$$

compare with $\frac{-1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu) = \frac{-1}{2}(x^T \Sigma^{-1} - 2x^T \Sigma^{-1} \mu + \mu^T \Sigma^{-1} \mu)$

令 $\Sigma^{-1} = \beta\phi(x)\phi^T(x) + S_N^{-1}, \quad \mu = \Sigma(\beta\phi(x)t + S_N^{-1} m_N)$

$$= \int_{-\infty}^{\infty} \exp\left\{\frac{-1}{2}\left[(w^T \Sigma^{-1} w - 2 w^T \Sigma^{-1} \mu + \mu^T \Sigma^{-1} m) - m^T \Sigma^{-1} \mu + \beta t^2\right]\right\} dw$$

$$= \cancel{\int_{-\infty}^{\infty} \exp\left\{\frac{-1}{2}[w - \mu]^T \Sigma^{-1} (w - \mu)\right\}}_{1} \cdot \exp\left\{\frac{-1}{2}(\beta t^2 - \mu^T \Sigma^{-1} \mu)\right\} dw$$

$$= \exp\left\{\frac{-1}{2}(\beta t^2 - \mu^T \Sigma^{-1} \mu)\right\} = \exp\left\{\frac{-1}{2}\left(\beta t^2 - (\Sigma(\beta\phi(x)) + S_N^{-1} m_N)^T \cancel{\Sigma^{-1}\Sigma}(\beta\phi(x)t)\right)\right\}$$

$$= \exp\left\{\frac{-1}{2}\left(\beta t^2 - (\beta\phi(x)t + S_N^{-1} m_N)^T \underbrace{\Sigma^T}_{=\Sigma} (\beta\phi(x)t + S_N^{-1} m_N)\right)\right\}.$$

$$\propto \exp\left\{\frac{-1}{2}\left[\beta t^2 - \beta^2\phi(x)^T\Sigma\phi(x)t^2 + 2(S_N^{-1}m_N)^T\Sigma\beta\phi(x)t\right]\right\}.$$

$$= \exp\left\{\frac{-1}{2}\left[(\beta - \beta^2\phi(x)^T\Sigma\phi(x))t^2 - 2(S_N^{-1}m_N)^T\Sigma\beta\phi(x)t\right]\right\}.$$

$$\propto \exp\left\{\frac{-1}{2}(\beta - \beta^2\phi(x)^T\Sigma\phi(x))\left(t - \frac{S_N^{-1}m_N^T\Sigma\beta\phi(x)t}{\beta - \beta^2\phi(x)^T\Sigma\phi(x)}\right)^2\right\}.$$

$$\rightarrow s^2(x) = (\beta - \beta^2\phi(x)^T\Sigma\phi(x))^{-1}$$

$$\Sigma = (\Sigma^{-1})^{-1} = [S_N^{-1} + \beta\phi(x)\phi^T(x)]^{-1} = S_N - \frac{S_N\beta\phi(x)^TS_N}{1+\phi(x)^TS_N(\beta\phi(x))}$$

$$\rightarrow s^2(x) = \beta - \beta^2\phi(x)^T\left(S_N - \frac{S_N\beta\phi(x)\phi^T(x)S_N}{1+\phi(x)^TS_N\beta\phi(x)}\right)\phi(x))^{-1}$$

$$= (\beta - \beta^2\phi(x)^TS_N\left(I - \frac{\beta\phi(x)\phi^T(x)S_N}{1+\beta\phi(x)^TS_N\phi(x)}\right)\phi(x))^{-1}$$

$$= (\beta - \beta^2\phi(x)^TS_N\frac{\phi(x) + \beta\phi(x)\phi(x)^TS_N\phi(x) - \beta\phi(x)\phi(x)^TS_N\phi(x)}{1+\beta\phi(x)^TS_N\phi(x)})^{-1}$$

$$= \left(\beta - \beta^2\frac{\phi(x)^TS_N\phi(x)}{1+\beta\phi(x)^TS_N\phi(x)}\right)^{-1} = \left[\beta\left(1-\beta\frac{\phi(x)^TS_N\phi(x)}{1+\beta\phi(x)^TS_N\phi(x)}\right)\right]^{-1}$$

$$= \left(\beta\cdot\frac{1}{1+\beta\phi(x)^TS_N\phi(x)}\right)^{-1} = \frac{1+\beta\phi(x)^TS_N\phi(x)}{\beta}$$

$$\rightarrow s^2(x) = \beta^{-1} + \phi(x)^TS_N\phi(x). \quad S_N^{-1} = \alpha I + \sum_{n=1}^{N}\phi(x_n)\phi(x_n)^T$$

$$m(x) = y(x, m_N) = \phi(x)^Tm_N = \phi(x)^T\left[S_N\left(\sum_{n=1}^{N}\beta\phi(x_n)t_n\right)\right)$$

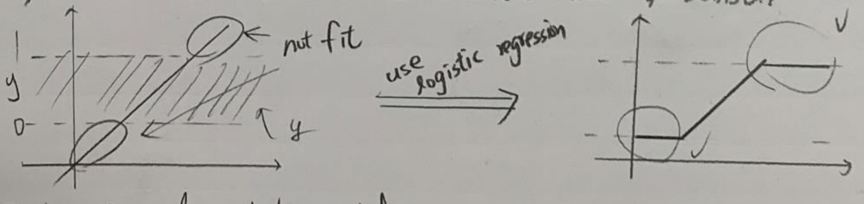
$$= \beta\phi(x)^TS_N\sum_{n=1}^{N}\phi(x_n)t_n.$$

$$\therefore p(t|x, \vec{x}, \vec{t}) = N(t|m(x), s^2(x)), \quad \begin{cases} m(x) = \beta\phi(x)^TS_N\sum_{n=1}^{N}\phi(x_n)t_n \\ s^2(x) = \beta^{-1} + \phi(x)^TS_N\phi(x) \end{cases}$$ ※

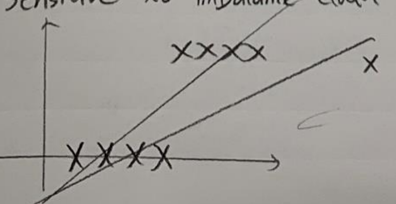
(3). No.

We know that in the classification scenario the target value y is either 0 or 1, It works on the probabilistic basis where we have thresold value. Here's get some problem where we trying to fit "linear regression" in classification.

① Predicted Values are continuous not probabilistic



② Sensitive to imbalance data



can't certainly classify

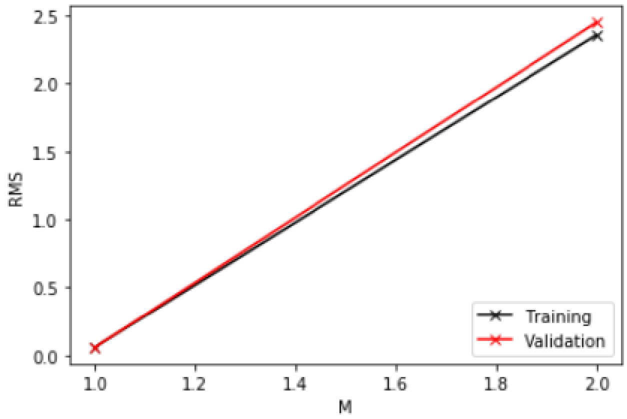
∴ Not use linear regression for classification ※

Q2：Linear Regression

1. Feature select

(a).RMS error:

```
［Training data result］:
M =  1     RMS =  0.05966376021822569
M =  2     RMS =  2.355572200243743
［Testing data result］:
M =  1     RMS =  0.05899539053729578
M =  2     RMS =  2.4481968767263704
```



計算需將 weight 先求出來，利用公式$\omega_{ML} = (\varphi^T\varphi)^{-1}\varphi^T t$，其中的$\varphi$可由

$$\mathbf{y(x, w)} = \sum_{j=0}^{M-1} \omega_j\varphi_j(x) = w^T\,\varphi(x)$$

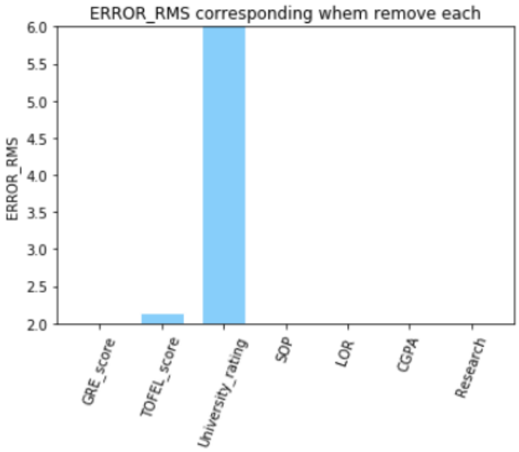
推得，最後代入 $\mathrm{E(w)} = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, w) - t_n\}^2$ 求得 RMS，並進行比較。

因顧慮原資料有照某種固定依據排序，所以先將 dataset 切割並重新排序，Training set 佔了 dataset_X 的 80%，Validation set 佔 dataset_X 的 20%。

由圖可知，在 Polynomial order M=1 時，training set RMS error 約為 0.059663，validation set RMS error 約為 0.05899839；在 Polynomial order M=2 時，training set RMS error 約為 2.3555722，validation set RMS error 約為 2.4481968。可知，當 Polynomial order M=2 時的 training set RMS error 較 Polynomial order M=1 時的 training set RMS error 小，表示較複雜的模型(Polynomial order M=2)對 training set 的 data 能有較好的學習結果。然而，Polynomial order M=2 時的 validation set RMS error 較 Polynomial order M=1 時的 validation set RMS error 大，表示較複雜的模型(Polynomial order M=2)的模型出現 overfitting 的現象。

(b).Select the most contribute attribute:

```
RMS of training REMOVE sth.
[1.181061578254035, 2.1360553662089106, 2202591784.9089675, 0.5218509619496206, 0.6612499210097312, 0.7858752738384913, 0.05832
1150460022286]
The most contributive attribte is: University_rating
```



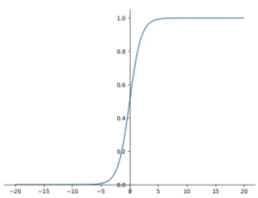
將 input training data 分別單獨 remove 其中一種 feature，並保留其他的 attribute 再重複算得 RMS，再算出在 Polynomial order M=1 時對應的 Training set RMS error 和 Validation set RMS error。

由圖可知，在移除 University Rating 時，training set RMS error、Validation set RMS error 皆較移除其他 attribute 高出許多，由此可知，University Rating 對訓練結果影響最大，所以 most contributive attribute 為 University Rating。

2.Maximum likelihood approach

(a).使用 Sigmoidal basis function.

$$\textbf{Sigmoid output} = \frac{\mathbf{1}}{\mathbf{1} + e^{-x}}$$



使用其原因為其在 ML、DL 為常見的 function，曲線平滑且容易推得。

(b).

```
//===================================//
//              2-2.(a,b)            //
//===================================//
//=== maximum likelihood approach ===//
//===================================//
//===> train_error_ml_1d = 0.06256452393106708
//===> validation_error_ml_1d = 0.05417647185084512
//==================================================//
//===> train_error_ml_2d = 0.060428075264598716
//===> validation_error_ml_2d = 0.04914900534691331
//==================================================//
```

同樣的切割dataset使Training set佔dataset_X的80%，Validation set佔dataset_X的20%。使用的basis function為Sigmoidal function，圖為Polynomial order M=1、M=2的RMS error。

由圖可知，在Polynomial order M=1時，training set RMS error約為0.0625645，validation set RMS error約為0.0541764；在Polynomial order M=2時，training set RMS error約為0.06042807，validation set RMS error約為0.049149005。由此可知，當Polynomial order M=2時的training set RMS error、validation set RMS error較Polynomial order M=1時的training set RMS error、validation set RMS error小一些，表示較複雜的模型(Polynomial order M=2)對training set 的data能有較好的學習結果。

(c).

```
//=================================//
//            2-2.(c)              //
//=================================//
//===  N-fold cross-validation  ===//
//=================================//
//=> train_error_ml_1d_nfold = 0.062145397707754575
//=> validation_error_ml_1d_nfold = 0.09873281010138148
--------------------------------------------------
//=> train_error_ml_1d_nfold = 0.0806742914955343
//=> validation_error_ml_1d_nfold = 0.07267986447451384
--------------------------------------------------
//=> train_error_ml_1d_nfold = 0.08455363127491465
//=> validation_error_ml_1d_nfold = 0.05128056841817997
--------------------------------------------------
//=> train_error_ml_1d_nfold = 0.07974089792208586
//=> validation_error_ml_1d_nfold = 0.06083479736761294
--------------------------------------------------
//=> train_error_ml_1d_nfold = 0.08400277221268905
//=> validation_error_ml_1d_nfold = 0.04486674564661924
--------------------------------------------------
//=========================================================//
//===> average_training_error = 0.09777924765324461
//===> average_validation_error = 0.08209869650207686
//=========================================================//
```

同樣的切割dataset使Training set 佔dataset_X的80%，Validation set 佔dataset_X的20%，此文選擇sigmoidal function為basis function。

使用N-fold cross-validation，並將N設定為4，由圖可知，在Polynomial order M=1時，Average training set RMS error約為0.097779，Average validation set RMS error約為0.0820986。

N-fold cross-validation主要作用是要防止因為模型過於複雜所引起的overfitting。

由Average training set RMS error(0.097779)與Average validation set RMS error(0.0820986的差距(0.0156804)可知，比較在沒有做N-fold cross-validation時，所得的training set RMS error(0.0625645)與validation set RMS error(0.0541764)的差距(0.0541764)來得小，故得知N-fold cross-validation可以有效降低over-fitting，使模型有更好的generalization能力。

3. Maximum a posterior approach

(a).

MLE (Maximum Likelihood Estimation)

$$\begin{aligned} &\theta_{MLE} \\ &= arg\ max\ p(X|\theta) \\ &= arg\ max \prod_i p(x_i|\theta) \\ &= arg\ max\ log \prod_i p(x_i|\theta) \\ &= arg\ max \sum_i log\ p(x_i|\theta) \end{aligned}$$

在樣本量較小時，MLE 的結論不可靠。

MAP (Maximum A Posterior)

$$\begin{aligned} &\theta_{MAP} \\ &= arg\ max\ p(X|\theta)p(\theta) \\ &= arg\ max\ log[p(X|\theta)] + log(p(\theta)) \\ &= arg\ max\ log \prod_i p(x_i|\theta) + log(p(\theta)) \\ &= arg\ max \sum_i log\ p(x_i|\theta) + log(p(\theta)) \end{aligned}$$

由以上公式，可以得出，當 prior follows uniform distribution 時，MLE 是 MAP 的一種特殊情況。

(b).

```
train_error_ml_1d_post = 0.07838047839749457
validation_error_ml_1d_post = 0.07197337803780822
=====================================================
train_error_ml_2d_post = 0.06044030126551411
validation_error_ml_2d_post = 0.04914900534691331
=====================================================
```

由2(b)中圖可知，使用maximum likelihood approach，Polynomial order M=1時，training set RMS error約為0.0625645，validation set RMS error約為0.0541764；在Polynomial order M=2時，training set RMS error約為0.06042807，validation set RMS error約為0.049149005。

由3(b)圖可知，使用maximum a posterior approach，Polynomial order M=1時，

training set RMS error約為0.07838，validation set RMS error約為0.07197;在Polynomial order M=2時，training set RMS error約為0.0604403，validation set RMS error約為0.049149。

由結果可得知，使用maximum likelihood approach 和maximum a posterior approach，在training set RMS error和validation set RMS error上差距相似，且在模型較複雜(Polynomial order M=2)時，有出現些微over-fitting的現象。猜測原因可能是訓練資料太少、訓練feature數量太少。

```
train_error_ml_1d_nfold = 0.06238544916062084
validation_error_ml_1d_nfold = 0.11295272263623703
train_error_ml_1d_nfold = 0.07885215740275954
validation_error_ml_1d_nfold = 0.07629873917753936
train_error_ml_1d_nfold = 0.0799256992723335
validation_error_ml_1d_nfold = 0.06194711573123125
train_error_ml_1d_nfold = 0.07756156328669339
validation_error_ml_1d_nfold = 0.072510279672909
train_error_ml_1d_nfold = 0.07844583113056526
validation_error_ml_1d_nfold = 0.06711123227376133
====================================================
average_training_error = 0.09429267506324313
average_validation_error = 0.09770502237291949
====================================================
```

選擇sigmoidal function為basis function，且使用N-fold cross-validation，並將N設定為4。由圖可知，使用maximum likelihood approach，在Polynomial order M=1時，Average training set RMS error約為0.097779，Average validation set RMS error約為0.0820986。

由附圖可知，使用maximum a posterior approach，在Polynomial order M=1時，Average training set RMS error約為0.0942926，Average validation set RMS error約為0.09770502。

由結果可知，兩者變化幅度並不大。

(c).

由(b)結果得知，MLE 在樣本數相對少的情況下，error 較高，與「在樣本量較小時，MLE 的結論不可靠」符合。