Machine Learning HW2

0560206 蔡孝謙

1.(a)

1. (a) $S_B = \sum_{k=1}^{K} N_k (m_k - m)(m_k - m)^T$, $m_k = \frac{1}{N_k} \sum_{n \in C_k} x_n$, $m = \frac{1}{N} \sum_{i=1}^{K} N_i m_i$

$J(w) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{w^T S_B w}{w^T S_W w}$ where $S_B = (m_2 - m_1)(m_2 - m_1)^T$, $S_W = \sum_{n \in C_1} (x_n - m_1)(x_n - m_1)^T + \sum_{n \in C_2} (x_n - m_2)(x_n - m_2)^T$

$\because S_W$ is Linear combination of positive-definite matrix $\Rightarrow w^T S_W w > 0$

令 $\nabla J(w) = 0$ $\quad (w^T S_B w) S_W w = (w^T S_B w) S_B w$

$$\Rightarrow w^T S_W w (S_B w) - w^T S_B w (S_B w) = 0 \Rightarrow \frac{\cancel{w^T S_W w} (S_B w)}{\cancel{w^T S_W w}} - \frac{w^T S_B w (S_W w)}{w^T S_W w} = 0$$

$$\Rightarrow S_B w - \frac{w^T S_B w}{w^T S_W w} (S_W w) = 0 \Rightarrow S_B w = \lambda S_W w \quad \left(\lambda = \frac{w^T S_B w}{w^T S_W w} \right)$$

If S_W is full-rank $\Rightarrow S_W^{-1} S_B w = \lambda w$, $\text{rank}(S_W^{-1} S_B) = \min(\text{rank}(S_W^{-1}), \text{rank}(S_B)) = \text{rank}(S_B)$

$$S_B = \begin{bmatrix} | & & | \\ (m_1 - m) & \cdots & (m_k - m) \\ | & & | \end{bmatrix} \begin{bmatrix} \frac{N_1}{N} & & 0 \\ & \ddots & \\ 0 & & \frac{N_k}{N} \end{bmatrix} \begin{bmatrix} -(m_1 - m)^T- \\ \vdots \\ -(m_k - m)^T- \end{bmatrix} \Rightarrow \text{rank}(S_B) = \text{rank} \begin{bmatrix} (m_1 - m) \cdots (m_k - m) \end{bmatrix}$$

$$\Rightarrow \sum_{i=1}^{K} N_i (m_i - m) = 0 \Rightarrow \text{rank}(S_B) \le k-1$$

1.(b)(HW2_1_b.m)
Mathodology
Training by generative model
我把三個class分別用不同的matrix 存起來，並獲得以下變數

- N_k為class k的資料量
- Class k的π，其中 $\pi_k = N_k/N$
- Class k的μ，每個μ都是1*4的vector，因為有四個attribute
- Class k的S，用課本的公式算出來的
- 全部共用的$S = \sum N_k/N * S_k$

有了這些值，就可以算Maximum likelihood，將x_n帶入每個class的pdf ($N(x_n | \mu_k, S)$)
找出比較大的pdf就是那個要的class
PCA
先將上面的S取eigenvector，並將原本的資料乘上前n大的eigenvalue對應到的eigenvector，就可以降維了
LDA
照課本的公式算出Sw和Sb，然後將inv(Sw)*Sb取eigenvector，再用跟PCA一樣的方法就能降維了

Result
[Generative model]
[Training data]

True\Predict	SET	VIR	VER	Total	Accuracy
SET	39	0	0	39	100%
VIR	0	40	1	41	98%
VER	0	2	38	40	95%
Total	39	42	39	120	

[Testing data]

True\Predict	SET	VIR	VER	Total	Accuracy
SET	11	0	0	11	100%
VIR	0	9	0	9	100%
VER	0	0	10	10	100%
Total	11	9	10	30	

[PCA Reduce dimension with generative model]
[Training data, Dimension =3]

True\Predict	SET	VIR	VER	Total	Accuracy
SET	39	0	0	39	100%
VIR	0	40	1	41	98%
VER	0	2	38	40	95%
Total	39	42	39	120	

[Testing data, Dimension =3]

True\Predict	SET	VIR	VER	Total	Accuracy
SET	11	0	0	11	100%
VIR	0	9	0	9	100%
VER	0	0	10	10	100%
Total	11	9	10	30	

[Training data, Dimension =2]

True\Predict	SET	VIR	VER	Total	Accuracy
SET	39	0	0	39	100%
VIR	0	33	8	41	80%
VER	0	3	37	40	93%
Total	39	36	45	120	

[Testing data, Dimension =2]

True\Predict	SET	VIR	VER	Total	Accuracy
SET	11	0	0	11	100%
VIR	0	7	2	9	78%
VER	0	3	7	10	70%
Total	11	10	9	30	

[Training data, Dimension =1]

True\Predict	SET	VIR	VER	Total	Accuracy
SET	39	0	0	39	100%
VIR	0	31	10	41	76%
VER	3	5	32	40	80%
Total	42	36	42	120	

[Testing data, Dimension =1]

True\Predict	SET	VIR	VER	Total	Accuracy
SET	11	0	0	11	100%
VIR	0	7	2	9	78%
VER	1	3	6	10	60%
Total	12	10	8	30	

[LDA Reduce dimension with generative model]

[Training data, Dimension =3]

True\Predict	SET	VIR	VER	Total	Accuracy
SET	39	0	0	39	100%
VIR	0	40	1	41	98%
VER	0	2	38	40	95%
Total	39	42	39	120	

[Testing data, Dimension =3]

True\Predict	SET	VIR	VER	Total	Accuracy
SET	11	0	0	11	100%
VIR	0	9	0	9	100%
VER	0	0	10	10	100%
Total	11	9	10	30	

[Training data, Dimension =2]

True\Predict	SET	VIR	VER	Total	Accuracy
SET	39	0	0	39	100%
VIR	0	40	1	41	98%
VER	0	2	38	40	95%
Total	39	42	39	120	

[Testing data, Dimension =2]

True\Predict	SET	VIR	VER	Total	Accuracy
SET	11	0	0	11	100%
VIR	0	7	2	9	78%
VER	0	3	7	10	70%
Total	11	10	9	30	

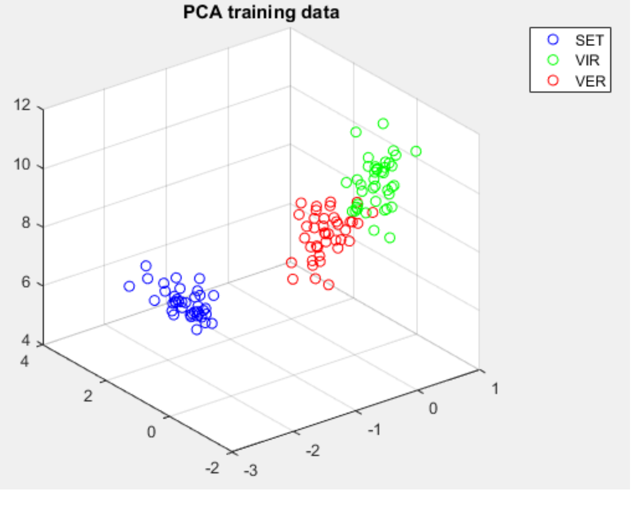
[Training data, Dimension =1]

True\Predict	SET	VIR	VER	Total	Accuracy
SET	39	0	0	39	100%
VIR	0	40	1	41	98%
VER	0	2	38	40	95%
Total	39	42	39	120	

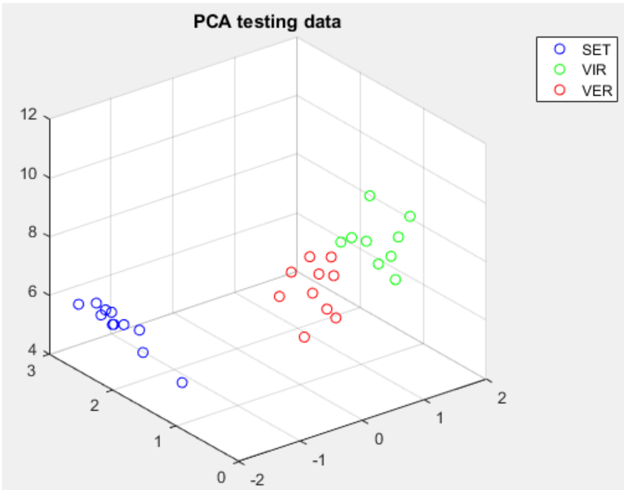
[Testing data, Dimension =1]

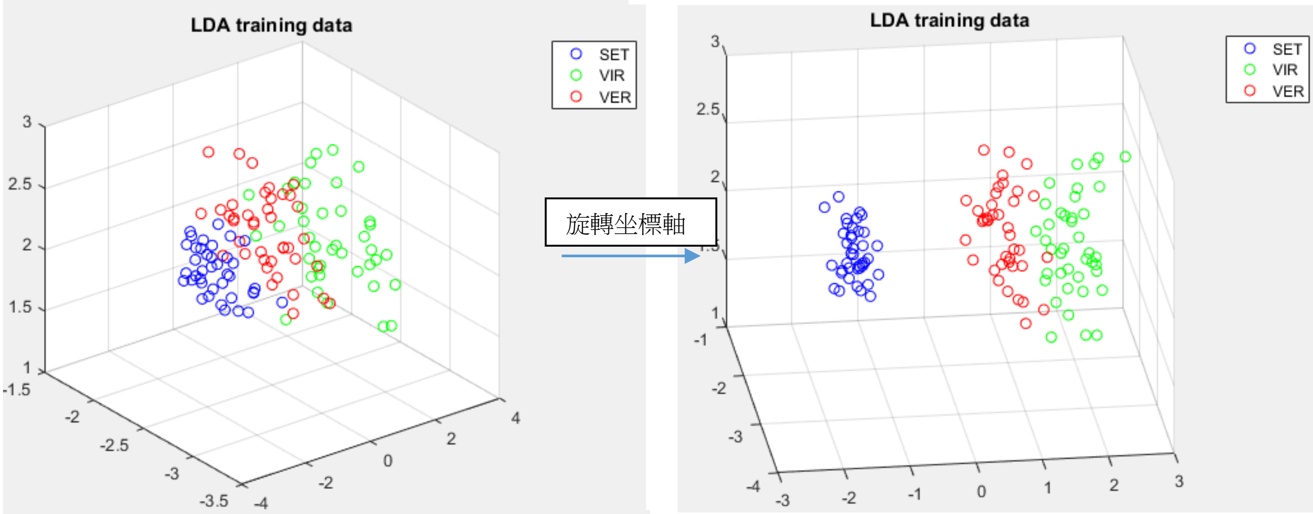
True\Predict	SET	VIR	VER	Total	Accuracy
SET	11	0	0	11	100%
VIR	0	7	2	9	78%
VER	1	3	6	10	60%
Total	12	10	8	30	

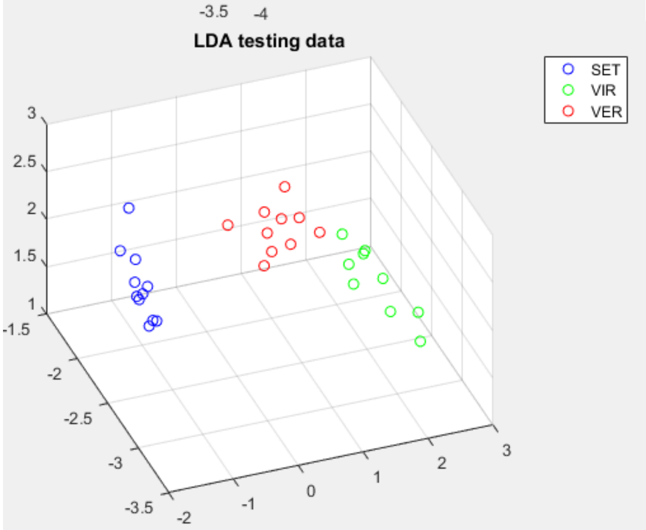
[Plot]



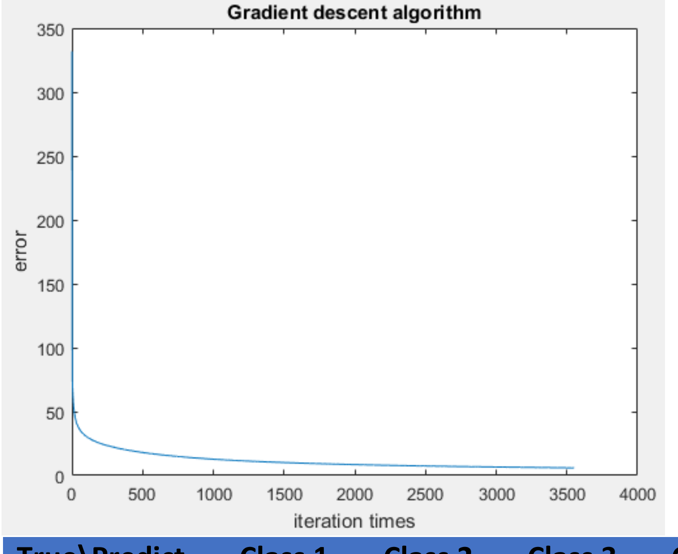
可以看到幾乎都可以把三種class分開，特別是SET(藍色的點)，VIR跟VER也只有少部分有重疊，可以得到即使我們把dim從4維降到3維，經過適當的轉換(PCA、LDA皆是以eigenevector為座標軸)後，分類結果還是不錯的







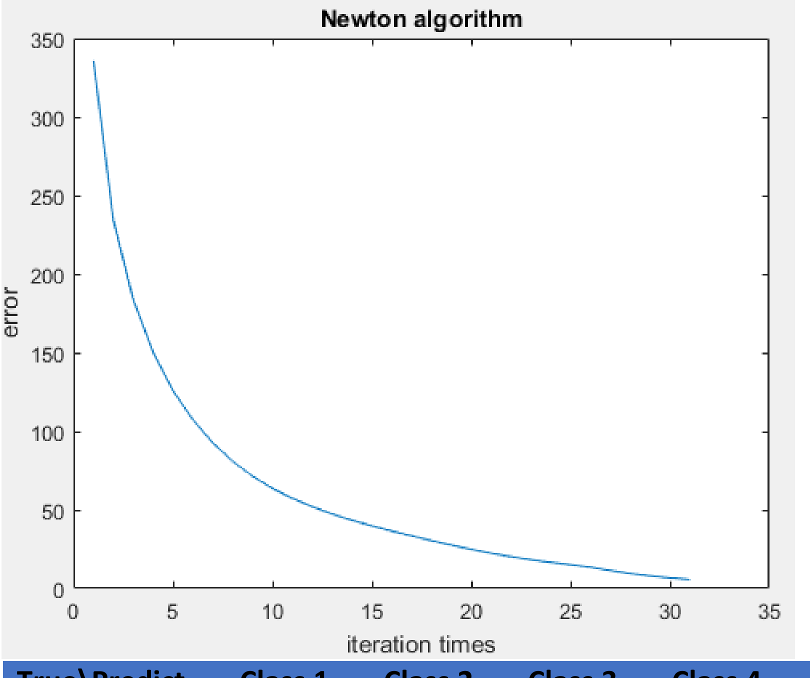
2.(a)(HW2_2.m)
Mathodology
Gradient descent
因為沒有要把x經過φ的feature space 轉換，所以代課本的公式的時候就直接用x當作φ(x)
先init一個w，然後固定learning rate，在照課本的公式去迭代，直到cross-entropy error<6 就停止訓練w
Newton Method
跟上題一樣，只是learning rate變成一個Matrix($X^T * H^{-1} * X$)，也是照課本的公式
但因為一開始出現NAN，所以我error前面乘上一個λ= exp(-2)，就沒有這個問題
Result
[Gradient descent]



True\Predict	Class 1	Class 2	Class 3	Class 4	Class 5	Total	Accuarcy
Class 1	10	0	0	0	0	10	100%
Class 2	0	10	0	0	0	10	100%
Class 3	0	0	10	0	0	10	100%
Class 4	3	0	0	5	2	10	50%
Class 5	0	0	0	0	10	10	100%
Total	13	10	10	5	12	50	

Misclassification rate = 1- 45/50 = 10%
Iteration times = 3551

[Newton algorithm]



True\Predict	Class 1	Class 2	Class 3	Class 4	Class 5	Total	Accuarcy
Class 1	10	0	0	0	0	10	100%
Class 2	0	10	0	0	0	10	100%
Class 3	0	0	10	0	0	10	100%
Class 4	3	0	0	5	2	10	50%
Class 5	0	0	0	0	10	10	100%
Total	13	10	10	5	12	50	

Misclassification rate = 1-45/50 = 10%
Iteration times = 31

可以發現在牛頓法的iteration times比一般的Gradient descent快很多(3551次 ---> 31次)!!

PS: 若將λ變小(exp(-3))，iteration times會變多，是因為step size的縮小所以要迭代較多次