Machine Learning HW2

0510894 電機 4D 翁紹恩

一、 Sequential Bayesian Learning

1、 實驗步驟：

要完成一個 sequential learning 須將前一項所算出的 posterior 當作下一項的 prior，這裡設前一項 prior 為$\mathrm{p(w)} = N(\mathrm{w}|m_0, S_0)$，$m_0$為該 prior 分布的 mean，$S_0$則為該 prior 分布的 covariance。

有了 prior 的機率分布之後，因為我們選擇 Conjugate prior distribution 所以 posterior 也會是 Gaussian distribution，由公式(2.116)結果，可以推得 posterior 結果，最後可得課本 p153 所推導出的 Bayesian Linear Regression 公式(3.50)、(3.51)：

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

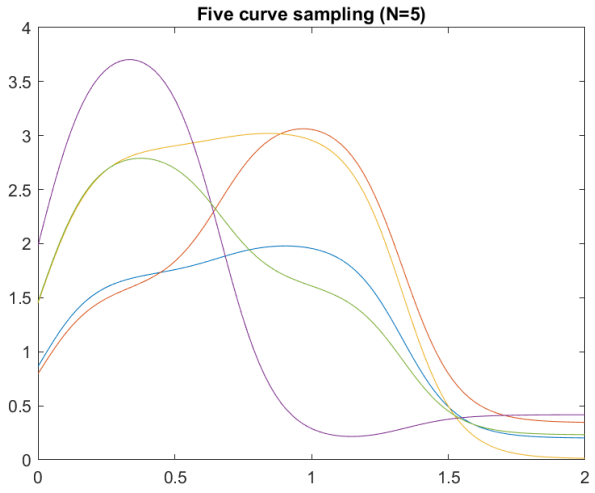
根據題意分別做 5、10、30、80 次後，利用 matlab multinomial random variable 公式 mvnrnd()，即可得出 w 值，並能用此 weight 預測新的數據資料分布。

在這裡$\Phi = \left[\sigma\left(\frac{x-\mu}{s}\right)\right]$,$\mu = \left[0, \frac{2}{3}, \frac{4}{3}\right]$,$s = 0.1$

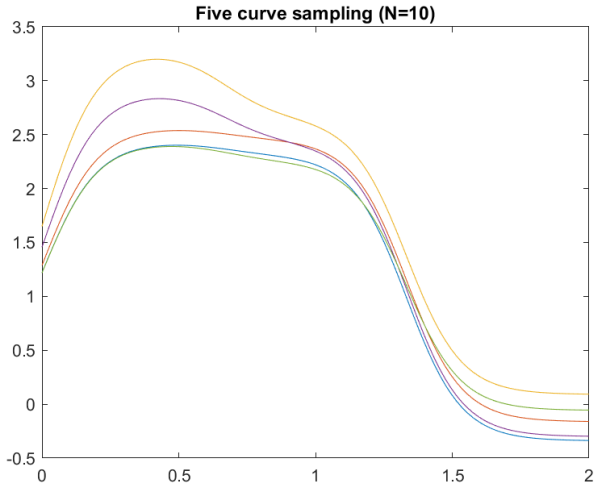
2、 問題與討論：

a、 Generate five curve samples from the parameter posterior distribution

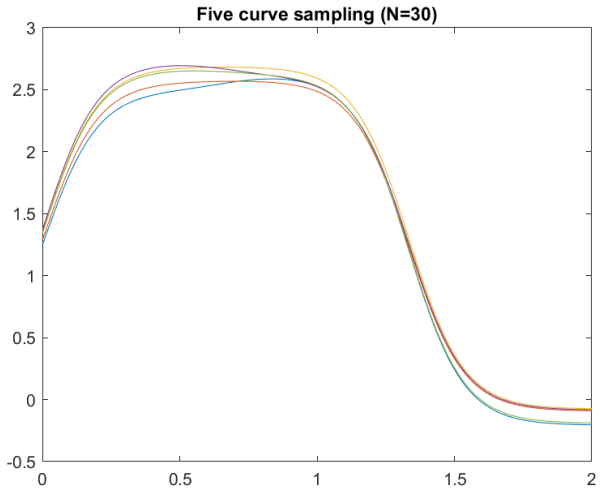
i、 N=5



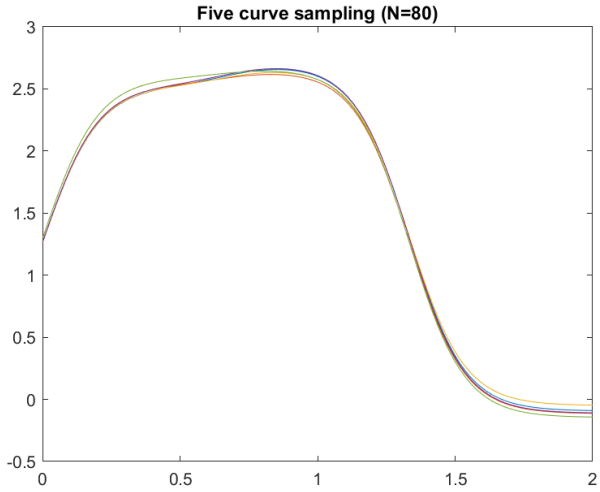
ii、 N=10



iii、 N=30



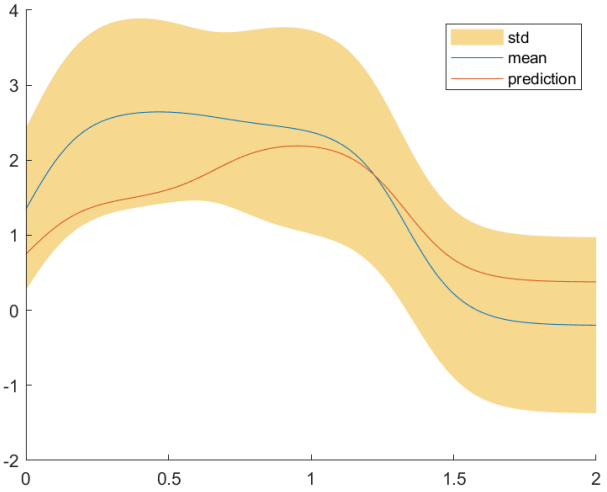
iv、 N=80



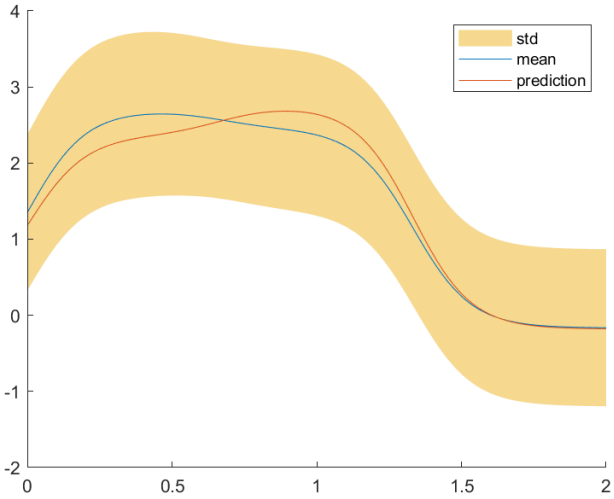
從實驗數據可以發現，當 sequential learning 次數愈多次，所預測出的結果即愈相似。一開始所得出的分布曲線十分分散，但是之後就愈來愈集中，所以依此預測出的結果變異性也較低，因此趨於精確。

b、 Plot the predictive distribution of target value t and show the mean curve and the region of variance with one standard deviation ※此處 std 為由 mean 之後正負一個標準差的範圍

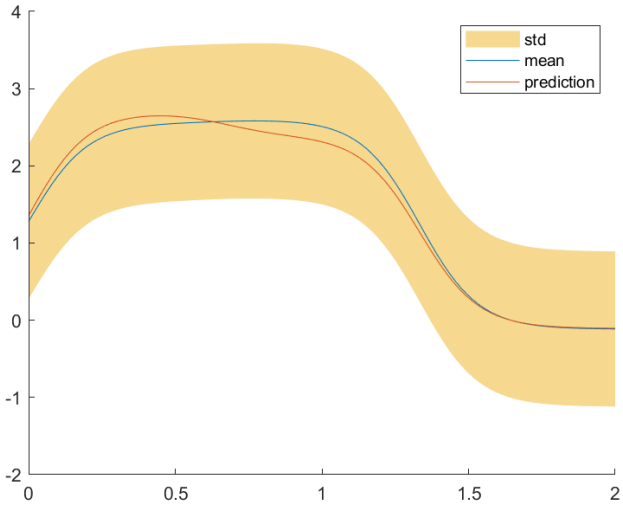
i、 N=5



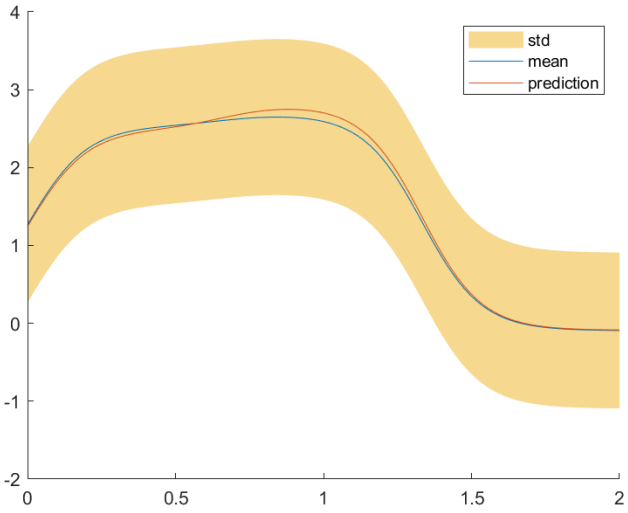
ii、 N=10



iii、 N=30



iv、 N=80

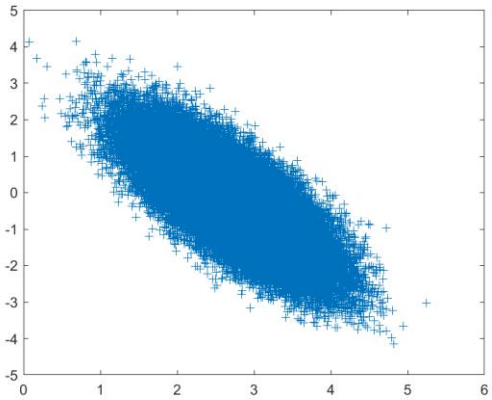


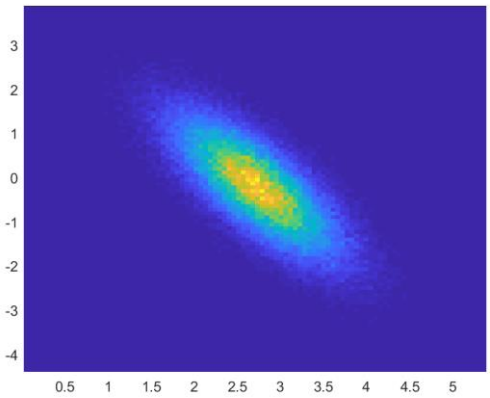
當 N=5 時，可以發現標準差的範圍明顯較其他三張圖大，因此預測結果範圍也會較不精確。而其他三張圖的標準差差異不大，但還是可以隱約看出從 N=10 到 N=80 標準差有逐漸縮小的趨勢。

從預測結果和平均值的角度來看，很明顯的，預測值在 N 大的時候趨近平均值，N 小的時候隨機亂數採樣之預測值相當不準確。由此可知，要達到好的預測結果，至少需要一定大小且盡量多一點的遞迴次數。

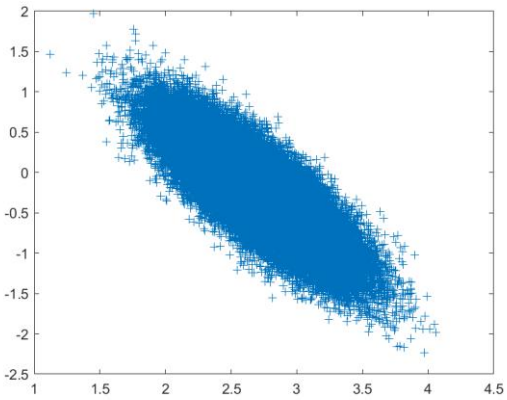
c、 Arbitrarily select two weights and plot the corresponding prior distributions(此提取第一、二維)

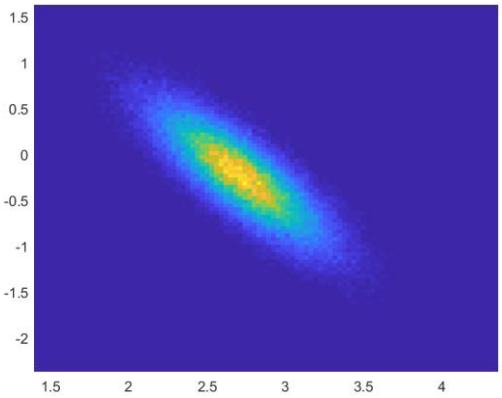
i、 N=5



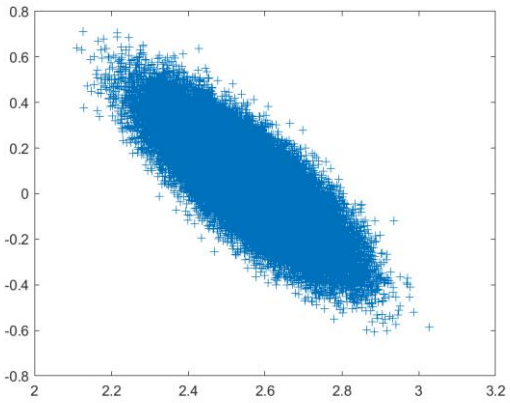


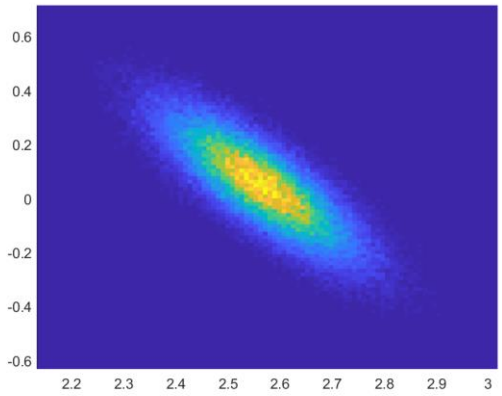
ii、 N=10



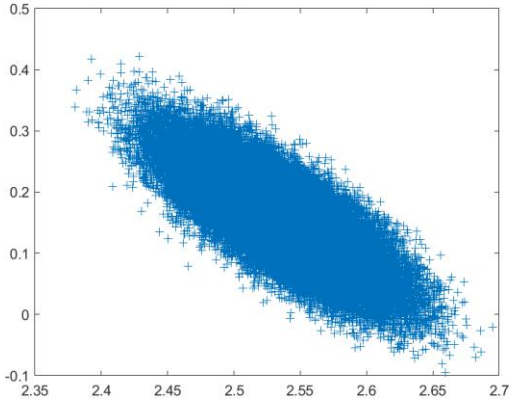


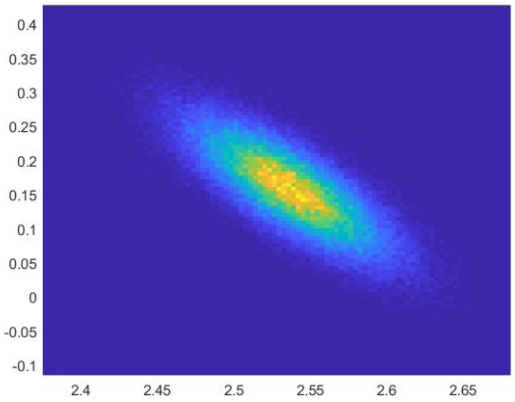
iii、 N=30





iv、 N=80





從上述實驗圖可以發現，當遞迴次數愈多時分布愈來愈集中，雖然圖看起來差不多，但相對應坐標軸的範圍其實是愈來愈小的，也可以從此圖右看出，因為是 normal distribution，所以中間的密度最高、顏色愈黃，往旁邊密度逐漸變疏、顏色愈藍。此可對應一.2.b.結果，sequential learning 次數愈多次，標準差愈小，而平均則差異不大。

二、 Logistic Regression

1、 實驗步驟：

a、 未使用 PCA 方法

$$a_k = {w_k}^T \varphi$$

$$y_k = \frac{\exp(a_k)}{\sum_{j=1}^{5} \exp(a_k)}$$

$$E(w) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_{nk}$$

$$\nabla E(w) = -\sum_{n=1}^{N} (y_n - t_n)\varphi_n$$

$$H = 0.001$$

$$w^{new} = w^{old} - H\nabla E(w)$$

重複此步驟直至 error 收斂。

b、 Principle Component Analysis PCA

PCA 的目的是要找到投影向量使得投影後的資料變異量最大以最佳化問題，也就是說盡可能使資料在降維之後仍還可以區分原始資料並獲得最大訊息量。因此需要先計算出資料的 Covariance matrix 後再去求解，而此過程即為解 Covariance matrix 的 eigenvector 與 eigenvalue，而最重要的特徵訊息就會存在於最大的 eigenvalue 相對應的 eigenvector 中，因此作法如下：

$$\mathrm{S} = (\frac{1}{N-1})\mathrm{X}^T X$$

$$[\mathrm{V}, \mathrm{D}] = \mathrm{eig}(\mathrm{S})$$

其中 S 為 Covariance matrix，eig()為求解 eigenvalue 和 eigenvector 方法，得出 V, D 後由於我們需要求出求幾個最大的主成分，因此在將之排序：

$$[\mathrm{d}, \mathrm{ind}] = \mathrm{sort}(\mathrm{diag}(\mathrm{D}), 'descend')$$
$$\mathrm{Ds} = \mathrm{D}(\mathrm{ind}, \mathrm{ind})$$
$$\mathrm{Vs} = \mathrm{V}(:, \mathrm{ind})$$

最後只要取出前幾項即可得到我們需要的結果，如二.2.c 附圖。

c、 使用 PCA+ Newton-Raphson algorithm 方法

先使用二.1.b 方法降維後，將得到的特徵當作 filter 重新對資料作處理以縮小原始資料訊息量，但仍可以適當保留主要判斷特徵，接著使用二.1.a 方法，其中 H 不再設定為固定數值 0.001，而設為：

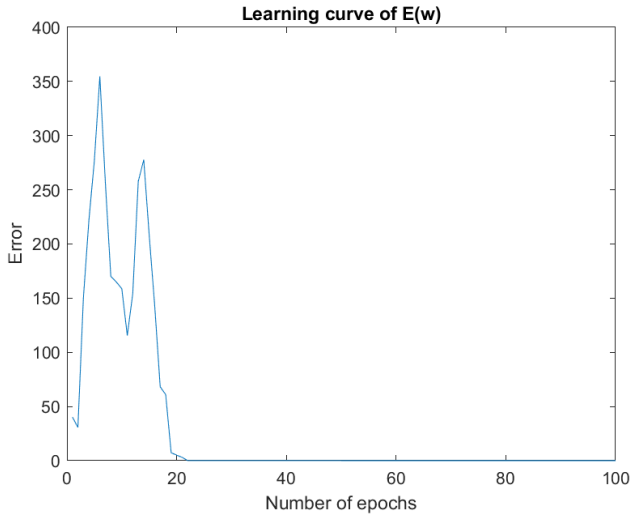
$$H = \sum_{n=1}^{N} y_n(1-y_n)\varphi_n\varphi_n{}^T$$

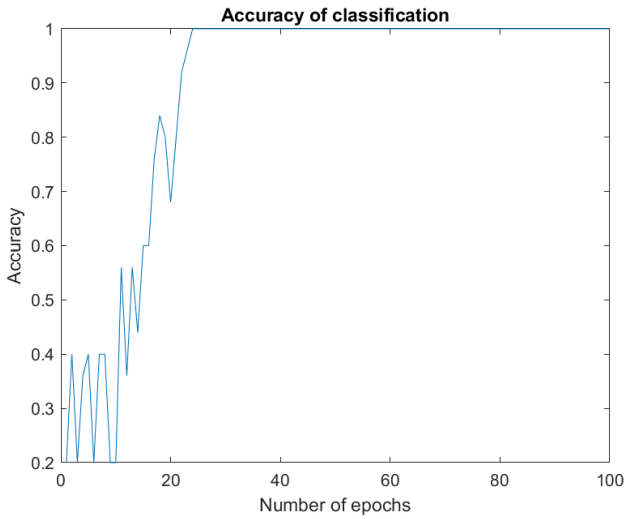
$$w^{new} = w^{old} - H^{-1}\nabla E(w)$$

使用 PCA 降維之後再做的好處是當訓練數據集很大時，有效地縮小資訊量可以增快計算過程、降低計算成本，並用 Newton-Raphson algorithm 做 learning rate 調整後，便可以依照學習狀況適當做調配，較不容易發生因為 learning rate 太大而錯過了真正可以收斂的最小值或是因為太小而收斂速度慢的問題，如二.2.d 附圖

2、 問題與討論：

a、 Learning curve of E(w) and the accuracy of classification versus the number of epochs





可以看到設定固定 learning rate 的 learning curve 在收斂前變化十分大，直到剛好找到適合收斂數值後即快速收斂。

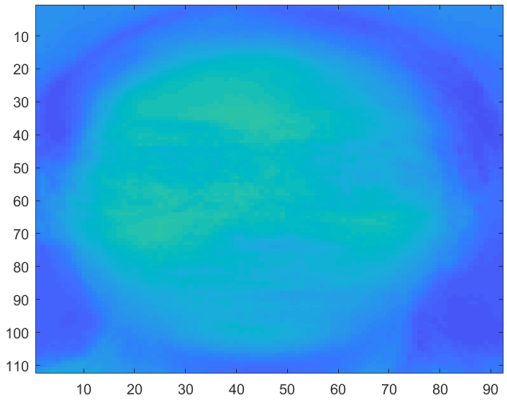
b、 The classification result of test data

此測試資料未經過打亂，所以放置順序為從第 1~5 人，可以發現收斂之後的資料預測十分精準。

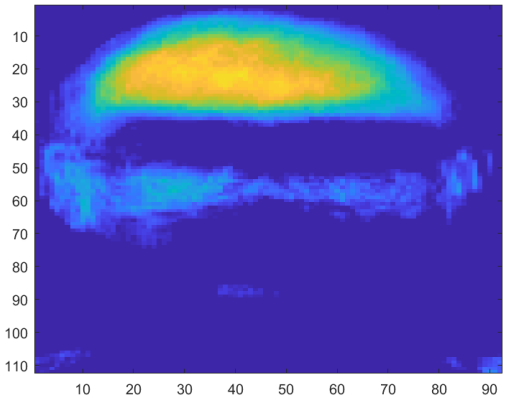
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
類別 1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
類別 2	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
類別 3	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
類別 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
類別 5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

c、 Use the principal component analysis (PCA) and plot five eigenvectors corresponding to top five eigenvalues

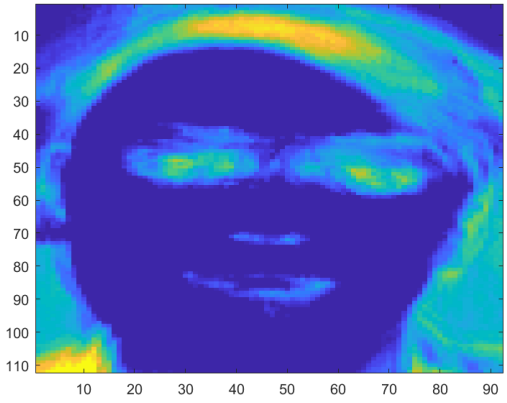
The largest eigenvector (eigenface)
(可以隱約看出是一張人臉的輪廓)



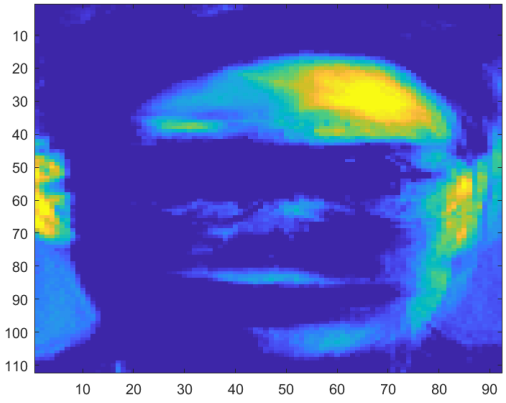
The second large eigenvector
(集中在人的額頭以及鼻子耳朵連線區域)



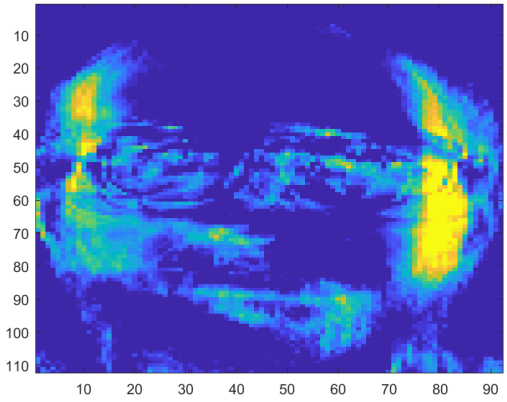
The third large eigenvector
(主要是眼耳鼻口頭髮等五官)



The fourth large eigenvector
(含額頭、臉頰等)



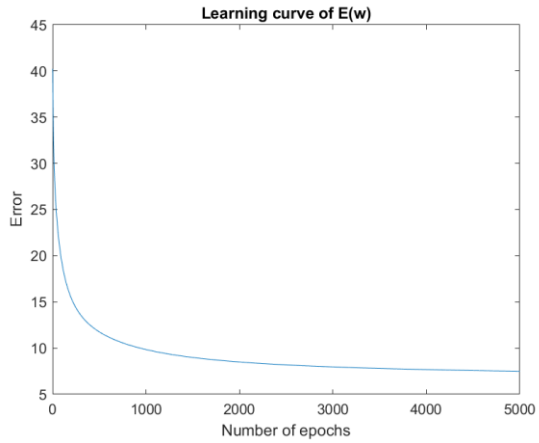
The fifth large eigenvector
(開始無法清楚描述出特定人臉特徵)

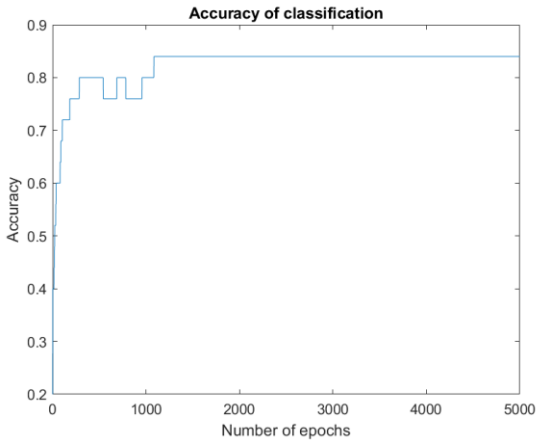


d、 Repeat 1 and 2 by applying Netwon-Raphson algorithm

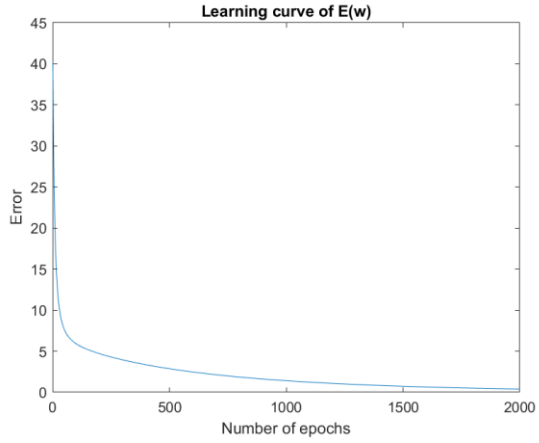
分別使用前 2、5、10 大的 eigenvalue 相對應的 eigenvetor，對圖片先進行降維後訓練。

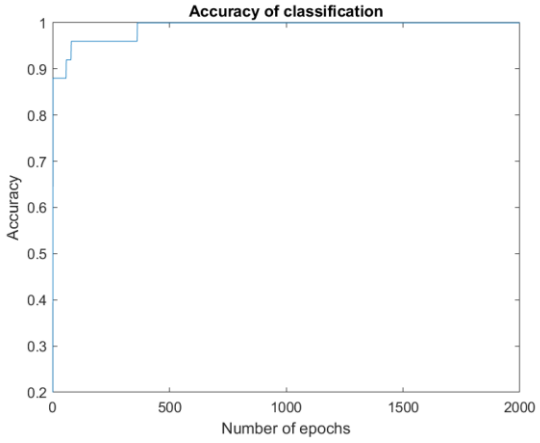
Dimension 2



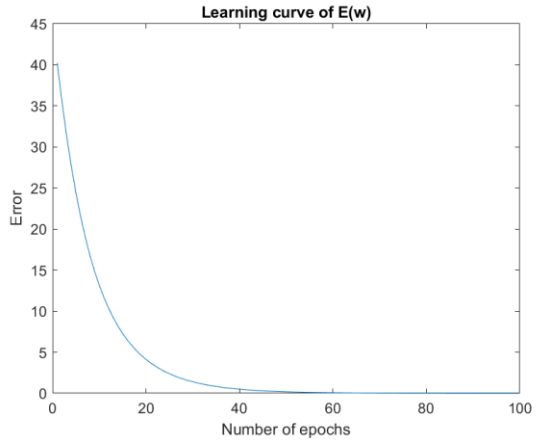


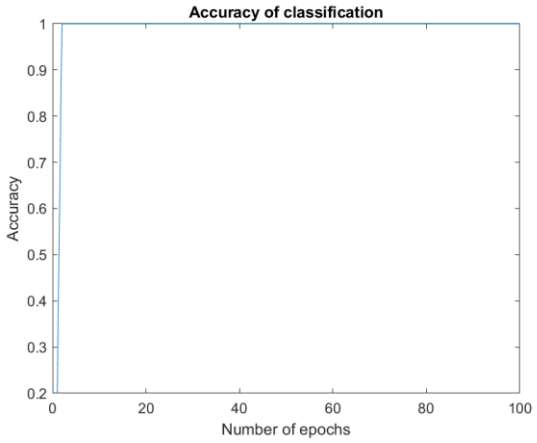
Dimension 5





Dimension 10





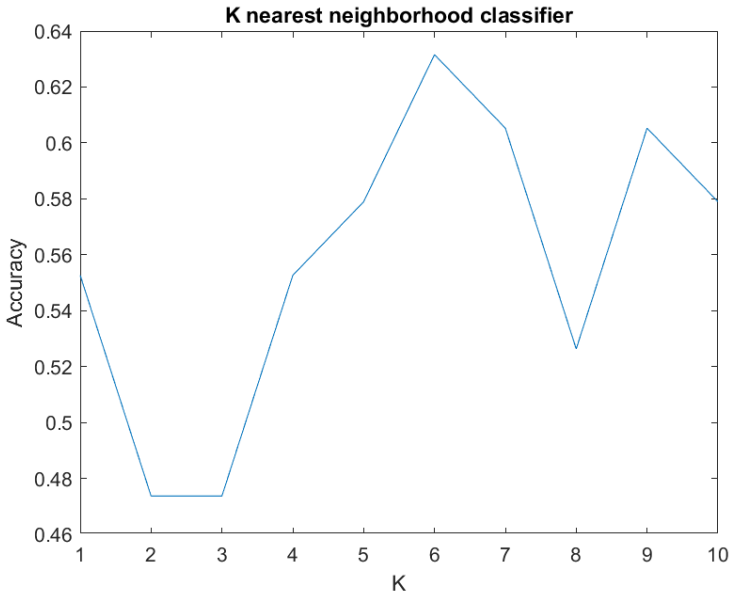
可以發現 learning curve 相對於二.2.a 的曲線變得十分平滑，但在 Dimension 2 和 5 時都需要遞迴訓練非常多次後才收斂，然而當取到 Dimension 10 後，因為有足夠特徵資訊因此和二.2.a 一樣，很快就收斂了。此方法在維度低時遞迴次數要較多次，但是訓練相對穩定，且資料量訊息量也較低，並且經過適當選取資料特徵維度後，便可以快速降低遞迴次數又達到穩定、低訊息量優點，且精確度提升十分快速。

e、 Discussion on the results of Netwon-Raphson and gradient descent algorithms

從上述實驗數據及推論，可以得出在訓練時對數據做主成分分析後再適當的降維可以幫助數據的訓練速度，儘管 PCA 本身需要運算一段時間，但是對於真正大量的訓料資料而言會節省許多時間也不會造成 GPU、CPU 負荷不了爆掉；至於使用 gradient descent 可以十分有效的得到正確結果，但若是在搭配上有效的設定 learning rate：Netwon-Raphson，便可以使演算法十分穩定，達到事半功倍的效果。

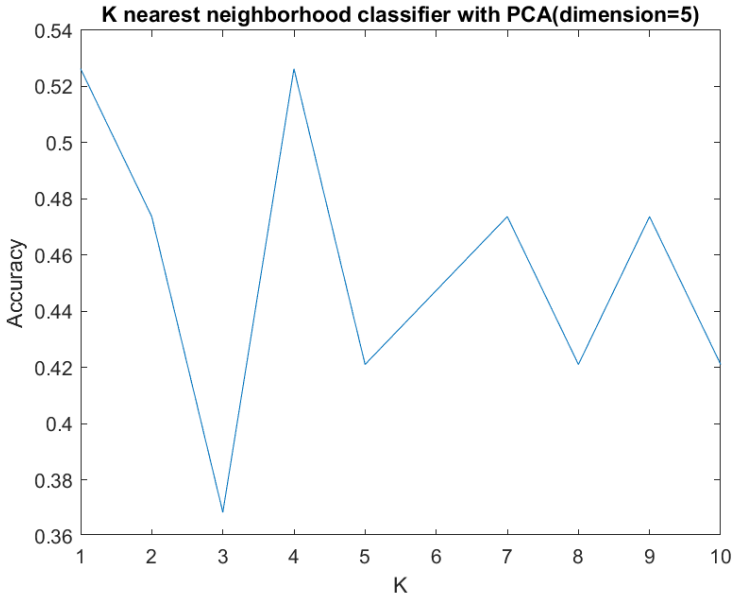
三、 Nonparametric Methods

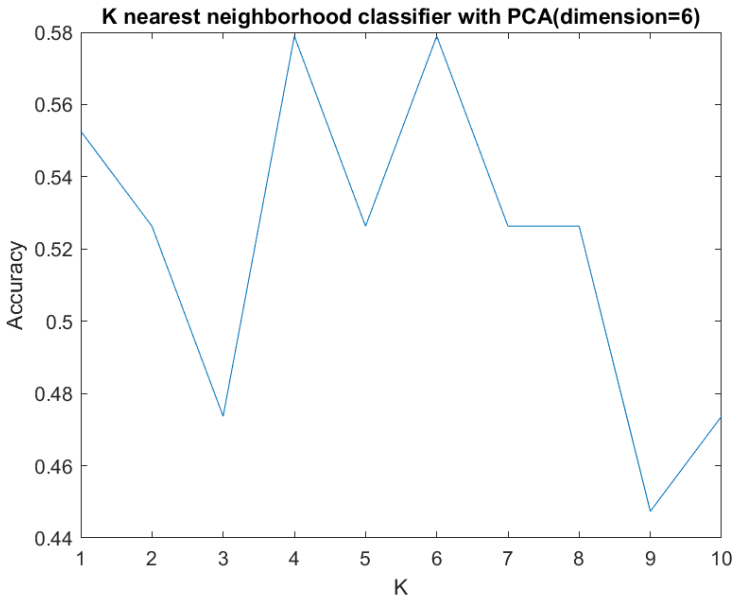
1、 The figure of accuracy

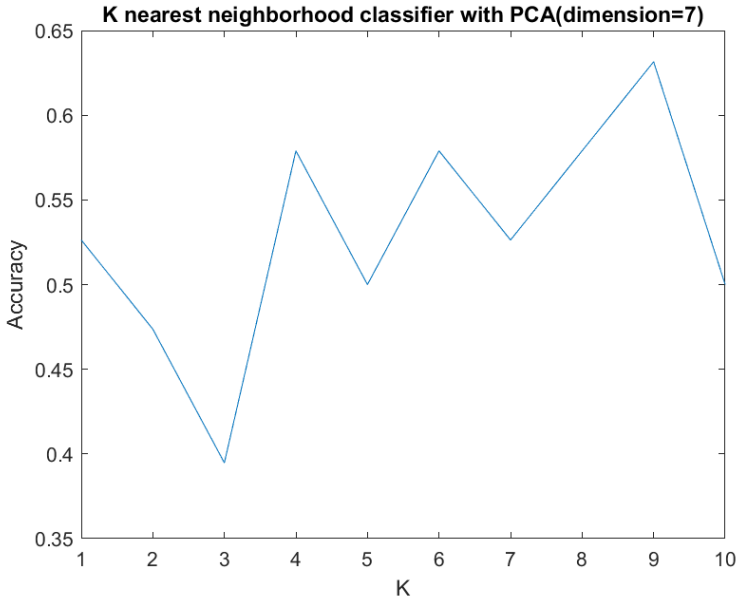


由上圖發現用 K nearest neighborhood 的方法，做出來的準確度並不高，主要是因為此方法對於資料局部結構十分敏感，像是有一個數值較特別的點在要測試的點附近，可能就會對該筆資料產生極大的影響。不過還是可以發現當 K 愈大時，整體精準度有上升的趨勢，但也由於較敏感的關係，數值點也不是取愈多表現結果愈好。

2、 Implement the principal component analysis (PCA) to plot the figure of accuracy







如二.1.b 先使用 PCA 降維之後，取的維度低精準度整體就稍偏低，當 K=7 的時候和使用全部維度資料做測試的精準度結果差不多，不過同上題，模型表現依然不佳，因為使用 K nearest neighborhood 對資料較敏感的緣故。