Homework 2

108B (5403) Machine Learning　王傳鈞　0416047

本次作業所使用到的程式碼，都已經上傳到[GitHub](https://github.com/a2468834/Machine_Learning_Grad/tree/master/HW2/MATLAB%20code)當中，若有需要參考其詳細內容，歡迎點擊連結直接前往網頁瀏覽。例如：「Q1\_plot.m」的MATLAB source code代表是第一題所使用到的繪圖程式碼，「Q2\_a.m」的MATLAB code則代表第二題的(a)小題所使用的程式碼，其餘命名方式請依此類推。

第一題

根據題目給定的數據，我們可以依據posterior probability由高到低做排序，並且依次計算出某筆數據之下 # of true positive instances (TP)、# of false positive instances (FP)、 # of true negative instances (TN) 、# of false negative instances (FN)、false positive rate (FPR)、true positive rate (TPR)等各項數值(詳細請見表格一)，最後繪製成FPR -TPR平面圖即可得到ROC curve(圖一)。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Rank | Label | TP | FP | TN | FN | FPR | TPR |
| 1 | P | 1 | 0 | 10 | 9 | 0.0 | 0.1 |
| 2 | P | 2 | 0 | 10 | 8 | 0.0 | 0.2 |
| 3 | P | 3 | 0 | 10 | 7 | 0.0 | 0.3 |
| 4 | P | 4 | 0 | 10 | 6 | 0.0 | 0.4 |
| 5 | N | 4 | 1 | 9 | 6 | 0.0 | 0.4 |
| 6 | P | 5 | 1 | 9 | 5 | 0.1 | 0.5 |
| 7 | P | 6 | 1 | 9 | 4 | 0.1 | 0.6 |
| 8 | N | 6 | 2 | 8 | 4 | 0.1 | 0.6 |
| 9 | P | 7 | 2 | 8 | 3 | 0.2 | 0.7 |
| 10 | N | 7 | 3 | 7 | 3 | 0.2 | 0.7 |
| 11 | P | 8 | 3 | 7 | 2 | 0.3 | 0.8 |
| 12 | N | 8 | 4 | 6 | 2 | 0.3 | 0.8 |
| 13 | P | 9 | 4 | 6 | 1 | 0.4 | 0.9 |
| 14 | N | 9 | 5 | 5 | 1 | 0.4 | 0.9 |
| 15 | N | 9 | 6 | 4 | 1 | 0.5 | 0.9 |
| 16 | N | 9 | 7 | 3 | 1 | 0.6 | 0.9 |
| 17 | N | 9 | 8 | 2 | 1 | 0.7 | 0.9 |
| 18 | P | 10 | 8 | 2 | 0 | 0.8 | 1.0 |
| 19 | N | 10 | 9 | 1 | 0 | 0.9 | 1.0 |
| 20 | N  Table 1 Ranking Method to Draw ROC curve and Calculate Area Under Curve | 10 | 10 | 0 | 0 | 1.0 | 1.0 |

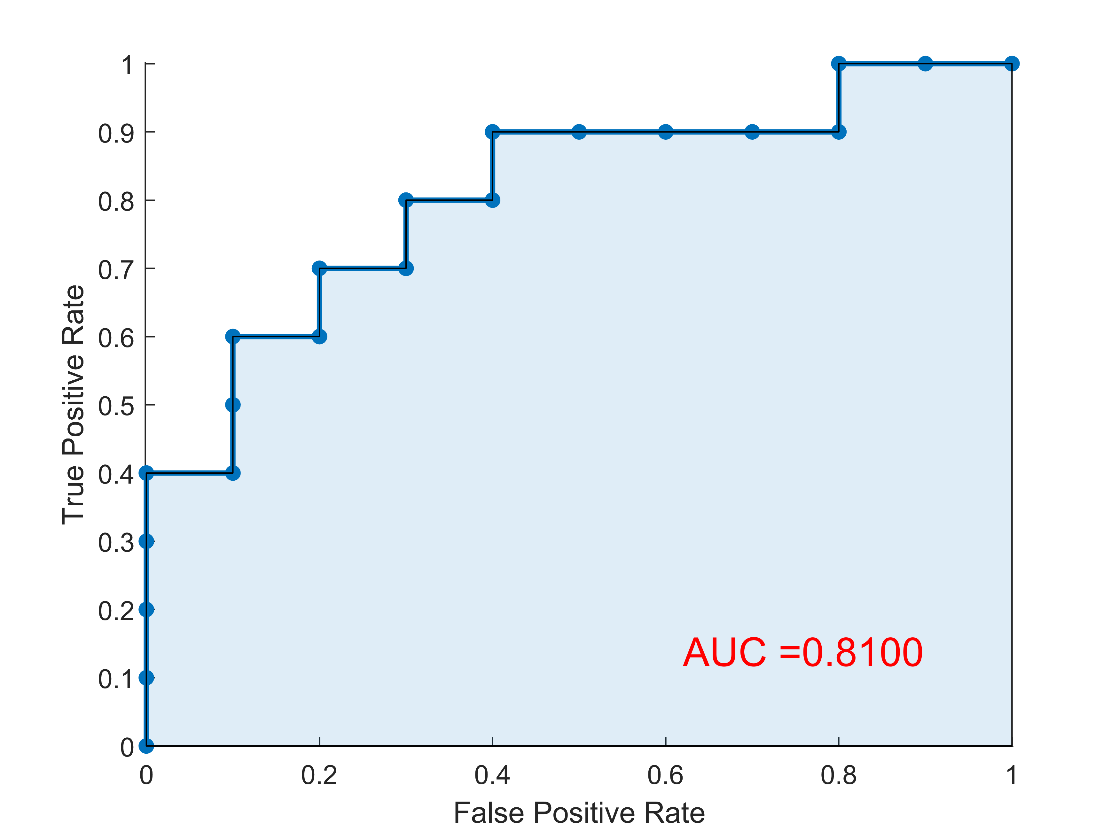


Figure 1 ROC curve to Question 1 and Its Area Under Curve

第二題

利用肉眼計算，我們可以很容易得知： 的最小值發生在 且 。

1. 利用我撰寫的MATLAB code「Q2\_a.m」，總共經過9731次疊代收斂在：

、。

以下表格二列出首三輪與尾三輪的詳細 疊代數值與變化量：

Table 2 Detailed Info. about First Three and Last Three Iterations (Gradient Descent)

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  |
|  |  |  | - |
|  |  |  |  |
|  |  |  |  |
| ．．． | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. 利用我撰寫的MATLAB code「Q2\_b.m」，總共經過1次疊代收斂在：

、**。**

以下表格三列出全部的詳細 疊代數值與變化量：

Table 3 Detailed Information about All Iterations (Newton’s Method)

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  |
|  |  |  |  |
|  |  |  |  |

經過以上兩種不同的方法，我們可以發現牛頓法收斂的速度驚人地快速，幾乎是早就知道答案般地，直接從起始點跳到最終結果；相反地，梯度下降法因為受限於每一次最多只能行走「梯度」長度的下降量，因此要很久才收斂到理想水準。但是，此例並無法顯示牛頓法的一個缺點：遇到特殊函數，若起始點選的離最終答案並不接近，則有可能永遠無法收斂。

第三題

1. 代表訓練集資料曾經被Perceptron分類到錯誤類別的次數，因此就是所有的訓練集資料被Perceptron分類錯誤的總次數。
2. 根據Novikoff定理，Perceptron algorithm可以在有限的步驟內停下來、得到答案，具備Knuth (1968, 1973)的演算法五個性質其一「finiteness」，因此updating rule是有效率的。

第四題

1. 請參考(b)小題內容。
2. 利用我撰寫的MATLAB code「Q4\_b.m」，可以得到：

hypothesis ，其中

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

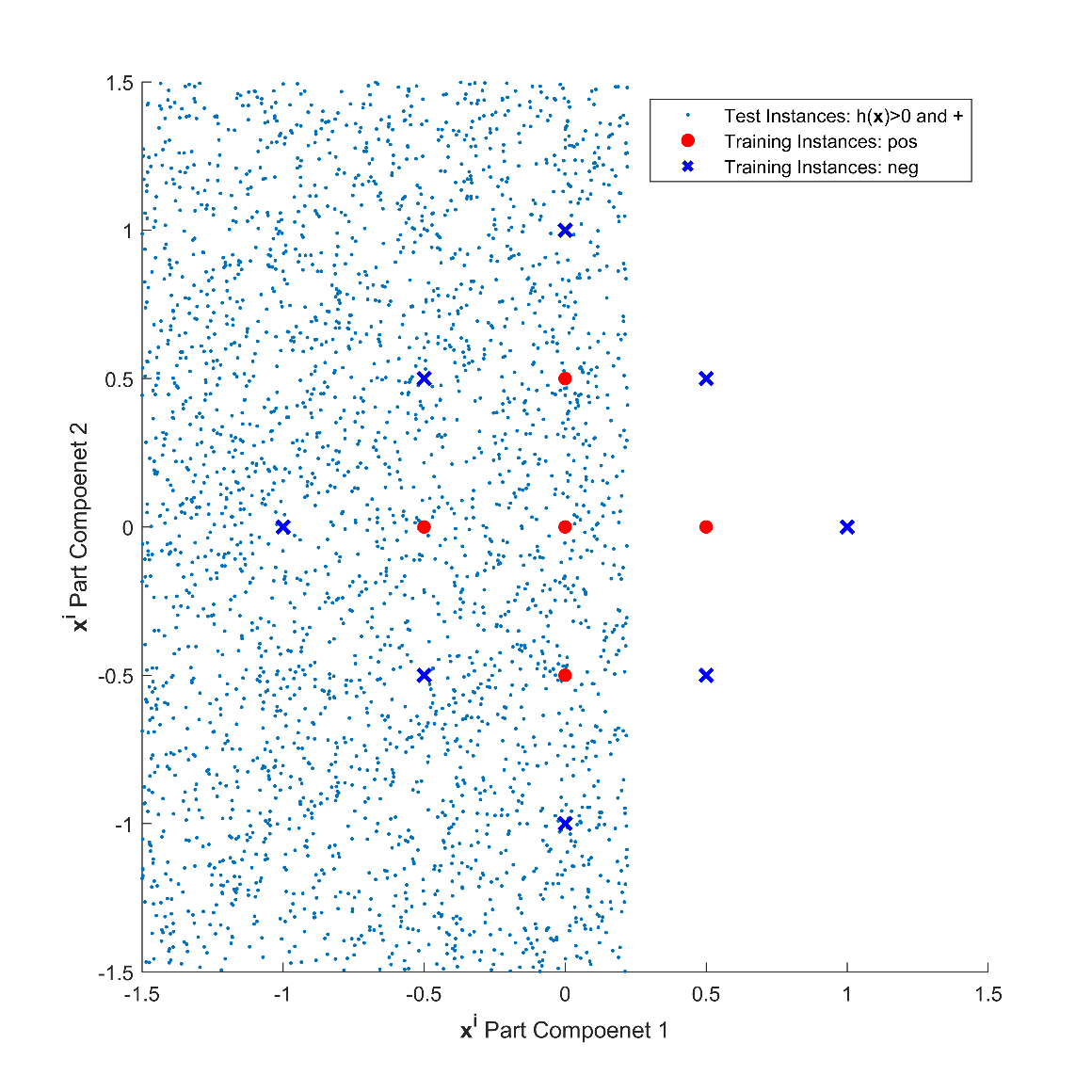
接著，我們用隨機產生的亂數當作測試集資料，並作圖可得以下圖二：

Figure 2 Scatter Plot of Question 4 (b)

上圖當中，較大顆的紅色、藍色點為訓練集資料；較小的淺藍色點為測試集資料且被Perceptron algorithm (dual form)認為positive且實際上也是positive之資料點。

我們可以發現：因為訓練集不是linearly separable，所以Perceptron找出切開平面的水平線只好犧牲一個positive instance被歸類成negative。

1. 仿造(a)、(b)小題的步驟使用「Q4\_c.m」，我們得到以下結果：

hypothesis ，其中

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

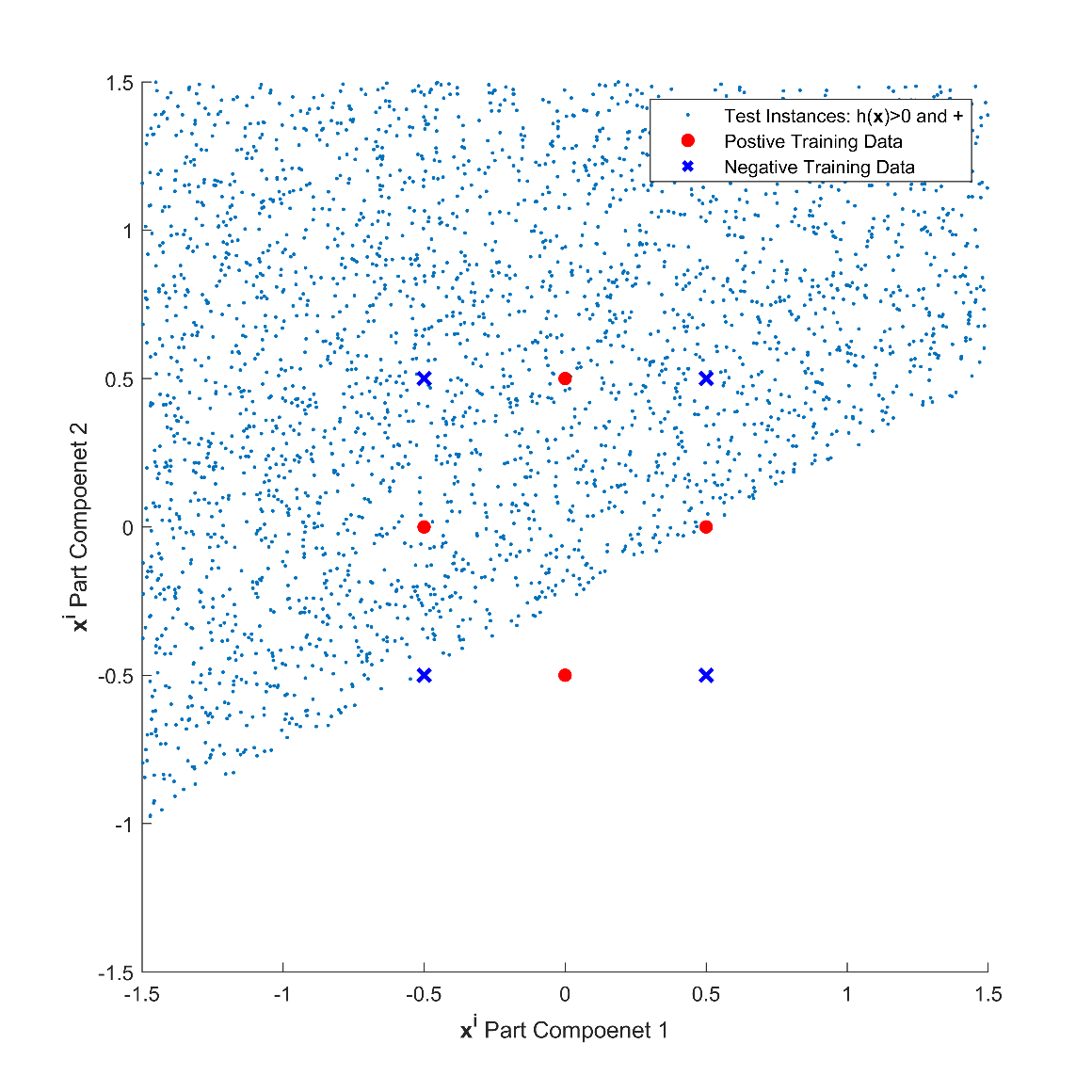


Figure 3 Scatter Plot of Question 4 (c)

1. 請參考(e)小題內容。
2. 利用我撰寫的MATLAB code「Q4\_e.m」，可以得到：

hypothesis ，其中

、

接著，我們用隨機產生的亂數當作測試集資料，並作圖可得以下圖四：

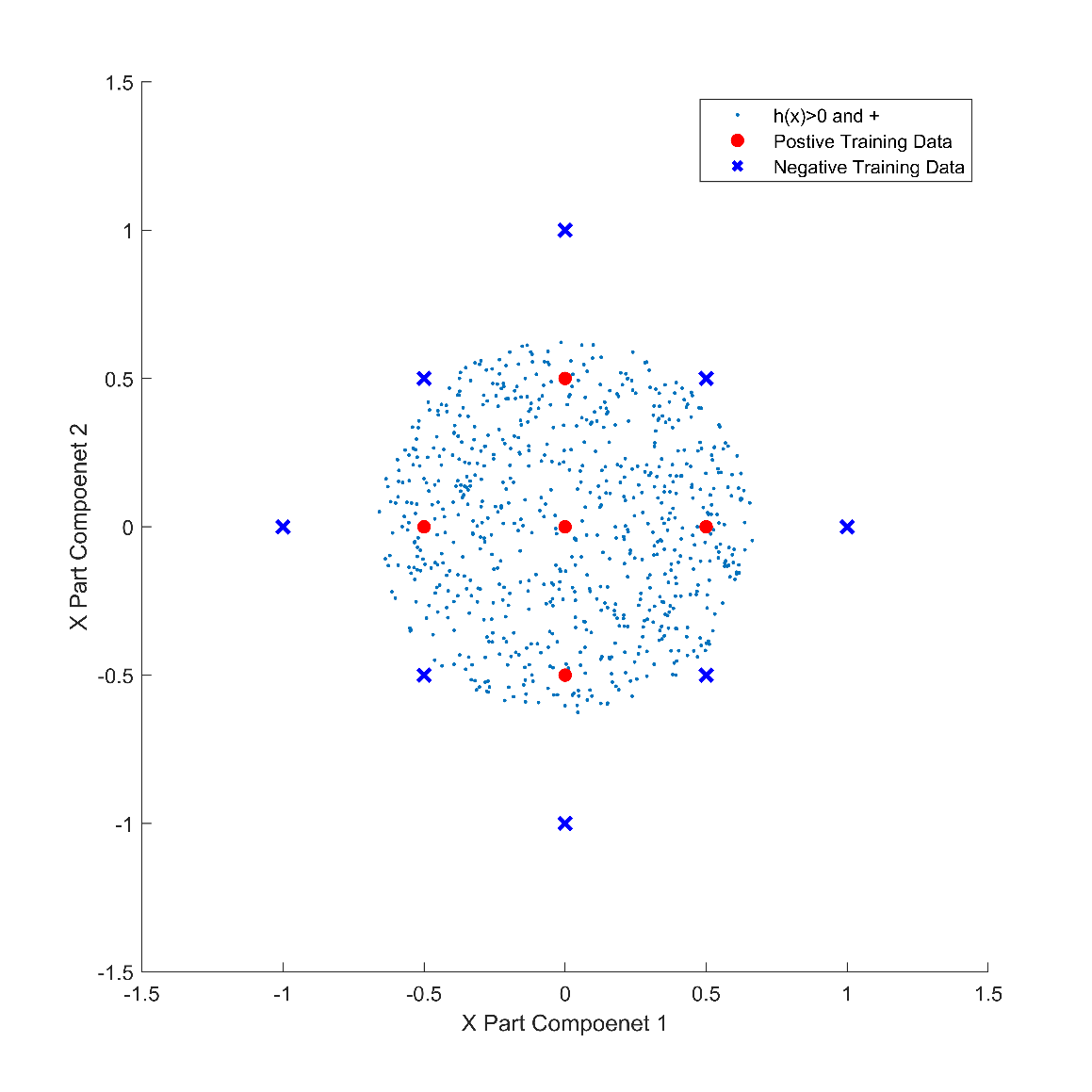


Figure 4 Scatter Plot of Question 4 (e)

上圖當中，較大顆的紅色、藍色點為訓練集資料；較小的淺藍色點為測試集資料且被Perceptron algorithm (primal form)認為positive且實際上也是positive之資料點。

我們可以發現：應用了映射到高維度的技巧，原本不是linearly separable的訓練集，也能被Perceptron找出完美的分割面，另測試集資料以合理的方式來分類。