

Machine Learning – Term Project

103062528 林玉山

1. 將 Soft-Margin Linear Classifier 原式改寫為 QP Form

$$x = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \\ \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \\ b \end{bmatrix}, Q = \begin{bmatrix} 1 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, C = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ c \\ c \\ \vdots \\ c \\ c \\ 0 \end{bmatrix}$$

$$G = \begin{bmatrix} -r^{(1)}x_1^{(1)} & -r^{(1)}x_2^{(1)} & \dots & -r^{(1)}x_d^{(1)} & -1 & 0 & \dots & 0 & r^{(1)} \\ -r^{(2)}x_1^{(2)} & -r^{(2)}x_2^{(2)} & \dots & -r^{(2)}x_d^{(2)} & 0 & -1 & \dots & 0 & r^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -r^{(n)}x_1^{(n)} & -r^{(n)}x_2^{(n)} & \dots & -r^{(n)}x_d^{(n)} & 0 & 0 & 0 & -1 & r^{(n)} \\ 0 & 0 & 0 & 0 & -1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}, h = \begin{bmatrix} -1 \\ -1 \\ \vdots \\ -1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

以下解釋各向量與矩陣元素：

- 向量 x 中依序為 w 向量(d 維)、 ξ 向量(n 維)與 b 常數
- 矩陣 Q 中，藍色區域為 d 維的 Identity Matrix，其餘為 0
- 向量 c 中，依序為 d 個 0、 n 個 c 與一個 0
- 矩陣 G 中，藍色區域為每一列皆為 $r^{(i)}x^{(i)}$ ($i = 1 \sim n$)，紅色區域各為一個乘上-1 的 n 維 Identity Matrix，綠色區域為向量 r (n 維)，其餘為 0
- 向量 h 中，依序為 n 個 -1 與 n 個 0

2. Coding Implementation

(a) 實作方式

a. Soft-Margin

training 時，使用 CVX Toolbox 代入 Primal Function 得到最佳的 hyperplane，並用於 prediction。

b. SMO

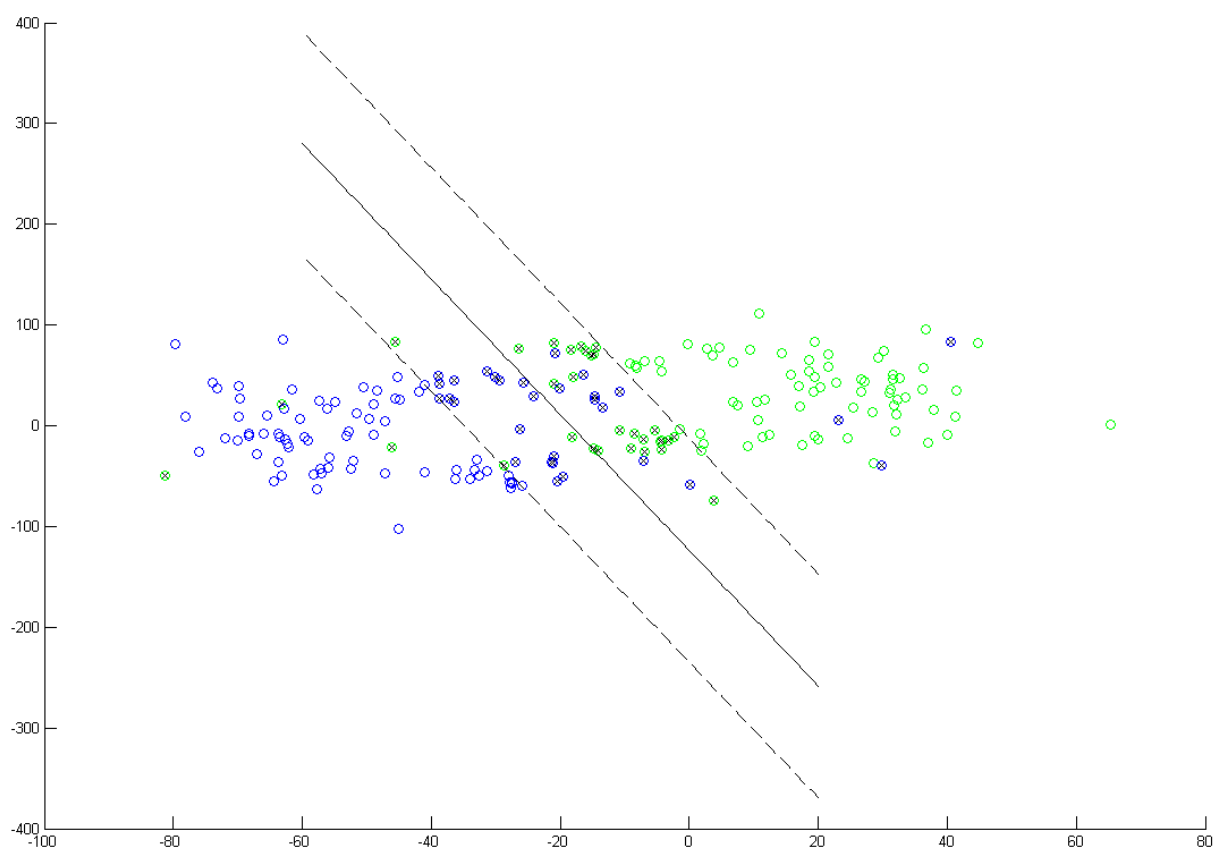
SMO 實作方式是依照講義提供的演算法實作。為了提早 terminate，特別在終止條件上加上一個 tolerance 的變數。只要數值足夠接近，就可以提早結束 SMO 的迴圈。另外，hyper-parameter C 採用固定的數值 10。Kernel Function 則採用 RBF Kernel，hyper-parameter γ 則固定為 0.001。

待 training 結束後，將會記錄下所有的 support vectors，以及相關的 labels。在 prediction 時，會被用來與目標 data 計算內積。最後，為了要加速存取 kernel matrix 的時間。特別實作了一個 handle class，KernelMatrix，用來暫存 kernel function 的結果。

(b) 畫出 Decision Boundary

a. Soft-Margin Classifier

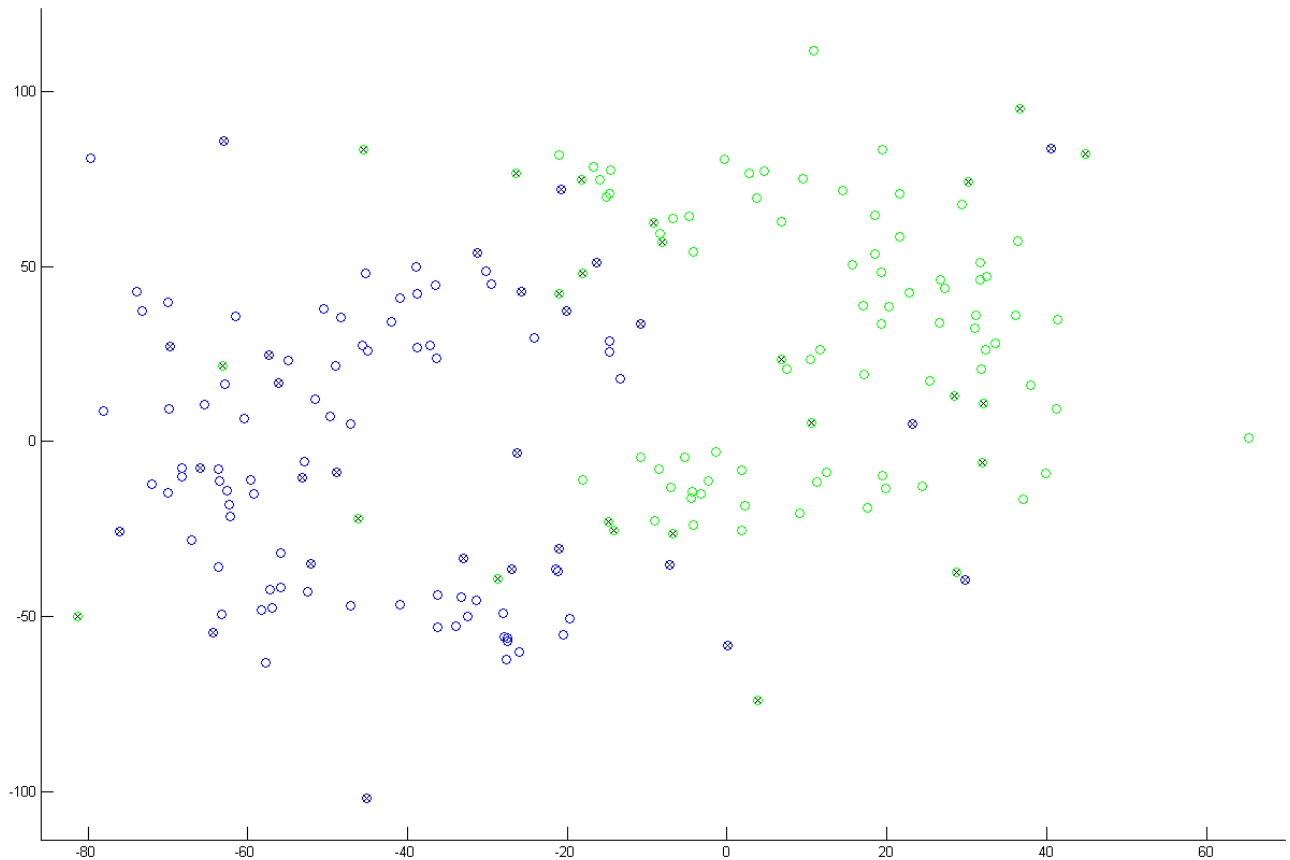
Label 為正的點為綠色，負的則為藍色。打叉的是 Support Vectors。黑線是 decision boundary，虛線是 margin 的邊界。



b. SMO Classifier

因為 Kernel Function 是使用 RBF，所以無法在 2D 平面上畫出來，因此只有標出 Support

Vectors (打叉的點)。藍色與綠色分別為 label 為正與為負的點。



(c) 效率比較

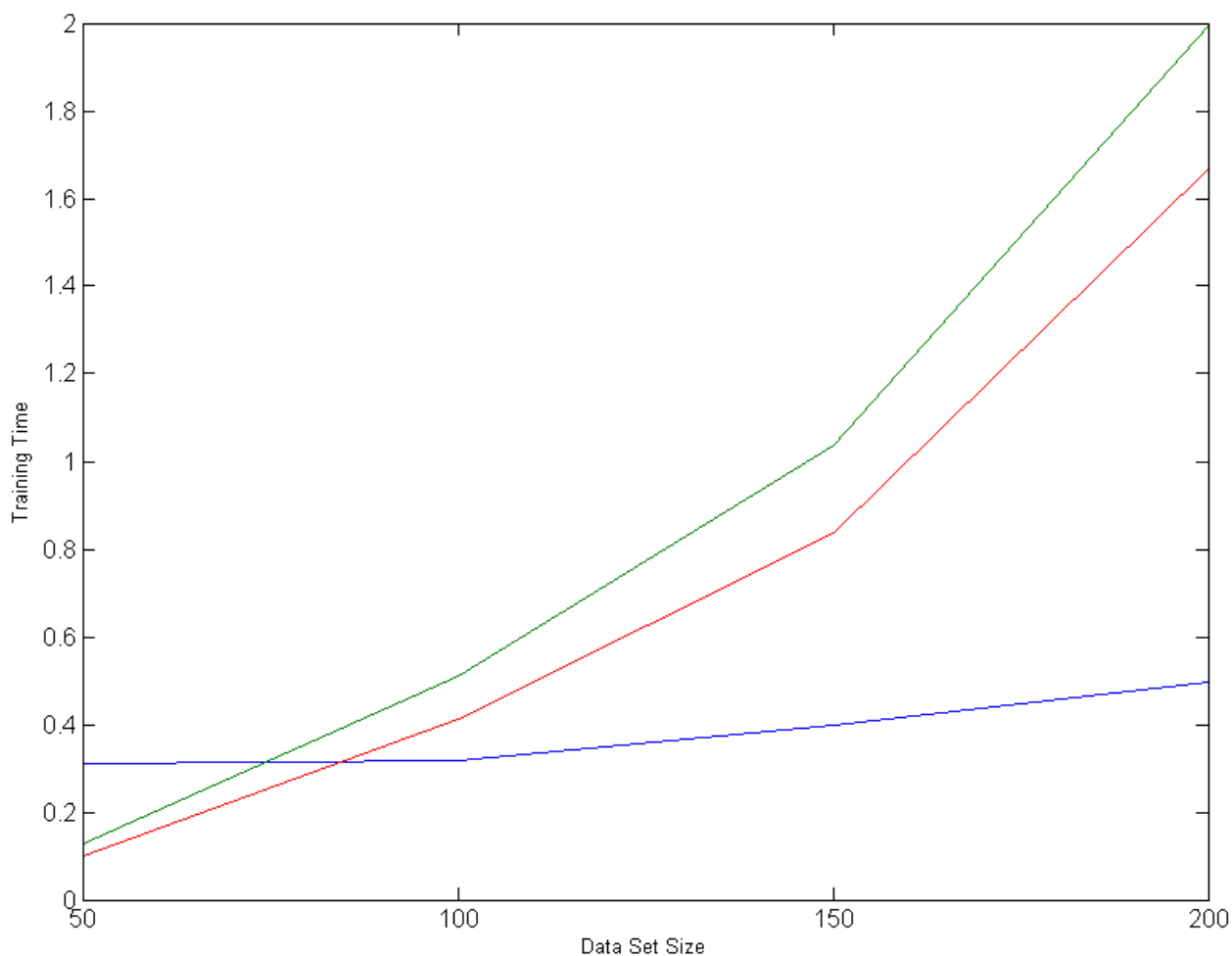
根據所提供的 200 個 Data，得到的結果如下：

Classifier	Num. of Iteration	Error Rate	Num. of SV.	Ex. Time (sec.)
Soft-Margin	24	9%	60	0.135
SMO	90	4%	50	1.869

(標示紅色者代表較佳，SV. 為 Support Vector)

從這個結果來看，使用 Soft-Margin 會得到較佳的速度。關於這點，很明顯是因為 Soft-Margin 並沒有做 lifting 到高維的動作，因此不需要計算 kernel function。另外，SMO 的 Error Rate 則比較低，這可能是因為使用了 RBF Kernel，將 data lifting 到較高維度的關係，使得在做分類的時候比較容易分開。

(d) Training 時間比較



上圖藍線為 Soft-Margin，綠線為 SMO, without caching，紅線為 SMO, with caching

Soft-Margin Classifier 使用的是 CVX Toolbox，可以看的出來，data size 似乎還沒大到會使 CVX 受到影響，因此變化不大。SMO 則多少會受到 data size 的影響。另外，有沒有 kernel caching 確實也會使得 training 時間縮短。根據實際實驗顯示，大概可以省下 30% 左右的時間。