Machine Learning – Term Project

103062528 林玉山

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

以下解釋各向量與矩陣元素：

1. 向量x中依序為w向量(d維)、 向量(n維)與b常數
2. 矩陣Q中，藍色區域為d維的Identity Matrix，其餘為0
3. 向量c中，依序為d個0、n個c與一個0
4. 矩陣G中，藍色區域為每一列皆為 ()，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
5. 向量h中，依序為n個 -1與n個0
6. **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**

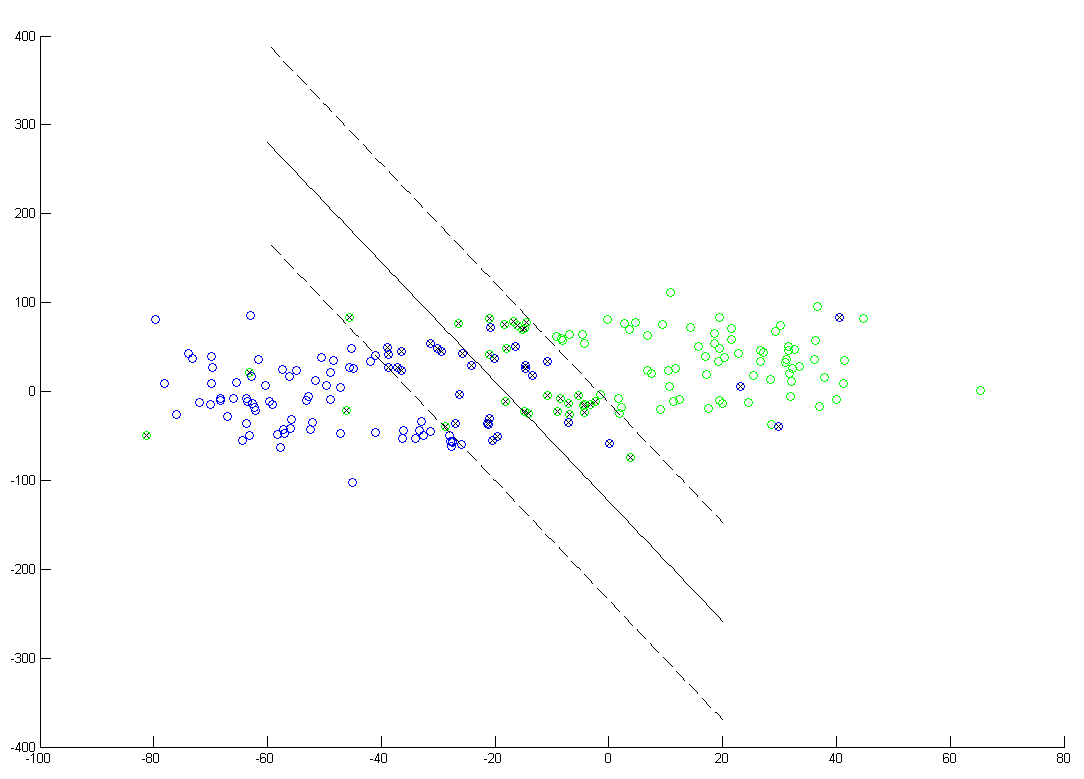
SMO實作方式是依照講義提供的演算法實作。為了提早terminate，特別在終止條件上加上一個tolerance的變數。只要數值足夠接近，就可以提早結束SMO的迴圈。另外，hyper-parameter C採用固定的數值10。Kernel Function則採用RBF Kernel，hyper-parameter 則固定為0.001。

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**

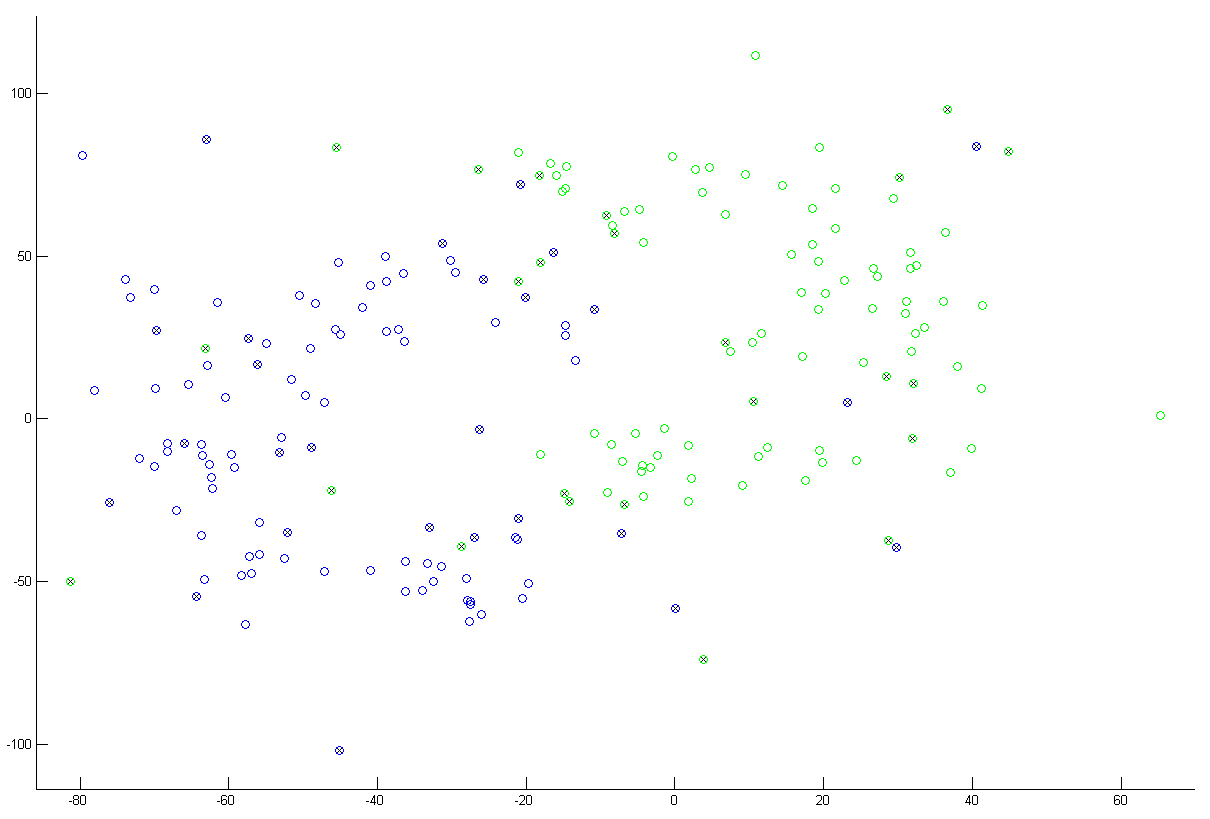
**a. Soft-Margin Classifier**

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**

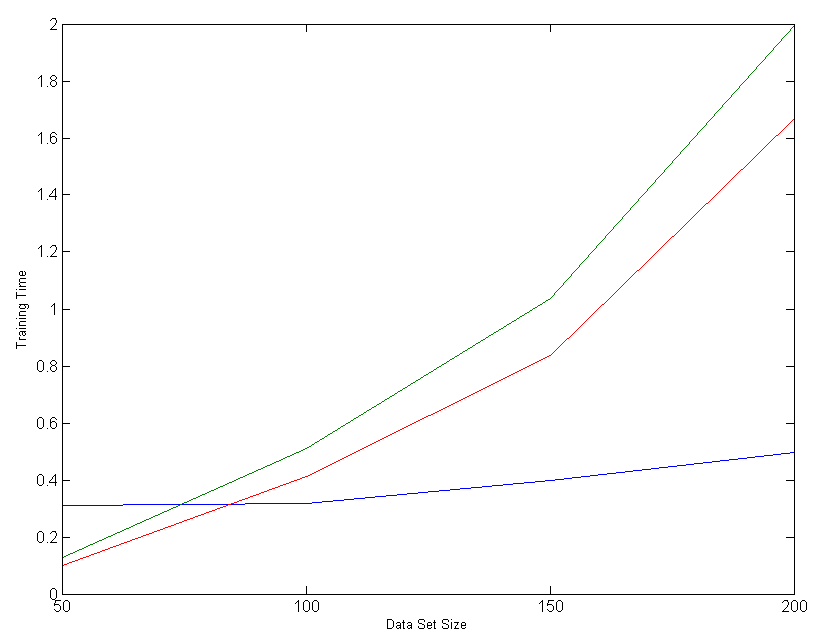
根據所提供的200個Data，得到的結果如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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。