

機器學習作業指示 - Homework #3

目標：

利用鳶尾花資料(Iris data set)來訓練Support Vector Machine (SVM) · 並分別比較線性與非線性SVM所訓練的超平面(hyperplane)有何差異。

Part1: Linear SVM (Initialization: $C=1$)

- Step1:** 將Iris data set的變色鳶尾(Versicolor, label=2)以及維吉尼亞鳶尾(Virginica, label=3)分別設為positive class與negative class · 並採用花瓣長度(3rd feature)與花瓣寬度(4th feature)作為特徵。
- Step2:** 將positive class與negative class的前25筆data設為training data · 剩餘的50筆設為test data。
- Step3:** 實現SVM分類器 (請勿直接使用開源的SVM套件)
- 補充：** 在求解對偶問題(Dual problem)時 · 可以利用MATLAB內建函式(quadprog : <https://www.mathworks.com/help/optim/ug/quadprog.html>) 或是 python 的開源程式(qpsolvers : <https://pypi.org/project/qpsolvers/>) · 並請自行將訓練資料矩陣化並參閱前兩者的documentation調整輸入的形式。
- Step4:** 將training data送至SVM中訓練 · 求解Lagrange multiplier - α · 再透過Kuhn-Tucker(KT) condition分析並求解**bias**。(紀錄**底線標記**之結果 · 數值請列到小數第四位)。
- Step5:** 將test data輸入至**Step4**所訓練的SVM decision function · 對50筆test data分別進行分類決策 · 並記錄test data的分類率。
- Step6:** 將penalty weight C 分別改為10以及100並重複**Step4 – Step5**。

Part2: RBF kernel-based SVM (Initialization: $C=10$, $\sigma=5$)

- Step1 – Step5:** 同Part1中**Step1 – Step5**流程 · 但採用RBF kernel。
- Step6:** 將sigma分別改為1、0.5、0.1以及0.05並重複**Step4 – Step5**。

Part3: Polynomial kernel-based SVM (Initialization: $C=10$, $p=1$)

- Step1 – Step5:** 同Part1中**Step1 – Step5**流程 · 但採用Polynomial kernel。
- Step6:** 將 p 分別改為 2、3、4、5 並重複 **Step4 – Step5**。

Part4 : Discussion and results presenting

請試著討論以下問題：

1. Linear SVM與kernel-based SVM所訓練的hyperplane有何差異？
2. 隨著 kernel parameter 的改變，RBF kernel 與 polynomial kernel 所訓練的 hyperplane可能有什麼變化？其與分類率的變化有何關聯？請嘗試解釋之。
3. 設定kernel parameter時，是否有方法避免hyperplane過度擬合(overfitting)的現象發生？若有請詳細討論。

將上列實驗之結果與討論事項整理後做詳細的討論，並以書面報告呈現。

作業繳交注意事項(遲交一周該次作業打8折，遲繳超過一周視同該次作業0分)：

1. 作業報告請以書面(Word or Markdown)呈現，並將程式碼一併壓縮在一個壓縮檔中。
2. 壓縮檔名請符合下列格式：「學號_姓名_HW3」(如：309511001_王小明_HW3)，否則作業成績打8折。
3. 請將壓縮檔上傳至E3數位教學平台。

Deadline : 2024/11/11 11:59 P.M. (“Three” weeks from now)

Alpha值參考：

Matlab：

機器學習HW#3											
利用Matlab之quadprog()之參考解答；僅列出"前5筆"之詳細 α 值，與"所有" α 值之總和。											
Linear SVM				RBF Kernel SVM (C = 10)				Polynomial Kernel SVM (C = 10)			
C=1		C=100		sigma = 5		sigma = 0.1		poly = 1		poly = 4	
alpha	Total	alpha	Total	alpha	Total	alpha	Total	alpha	Total	alpha	Total
1.0000	14.4000	0.0000	577.7778	10.0000	193.7710	0.2801	36.0576	0.0000	76.0000	0.0000	49.1489
0.0667		0.0000		8.9618		0.3292		0.0000		0.0000	
1.0000		44.5999		10.0000		1.2411		9.0002		2.0497	
0.0000		0.0000		0.0000		0.3631		0.0000		0.0000	
1.0000		0.0001		10.0000		0.0704		0.0000		0.0000	
C=10				sigma = 1		sigma = 0.05		poly = 2		poly = 5	
alpha	Total			alpha	Total	alpha	Total	alpha	Total	alpha	Total
0.0000	76.0000			0.0000	76.2552	0.5377	42.7545	0.0000	54.2594	0.0000	47.9100
0.0000				0.0000		0.3213		0.0000		0.0000	
9.0002				8.8908		0.6269		3.8295		3.4936	
0.0000				0.0000		0.5453		0.0000		0.0000	
0.0000				0.0000		0.9401		0.0000		0.0000	
				sigma = 0.5				poly = 3			
alpha	Total			alpha	Total			alpha	Total		
0.0000				0.0000	62.7532			0.0000	53.0684		
0.0000				0.0000				0.0000			
5.9852				0.0000				3.8736			
0.0000				0.0000				0.0000			
0.0000				0.0000				0.0000			

Python：

機器學習HW#3											
利用Python之開源套件qpsolver()之參考解答；僅列出"前5筆"之 α 值，與"所有" α 值之總和。											
Linear SVM				RBF Kernel SVM (C = 10)				Polynomial Kernel SVM (C = 10)			
C=1		C=100		sigma = 5		sigma = 0.1		poly = 1		poly = 4	
alpha	Total	alpha	Total	alpha	Total	alpha	Total	alpha	Total	alpha	Total
1.0000	14.4000	0.0000	577.7778	10.0000	193.7710	0.2801	36.0576	0.0000	76.0000	0.0000	49.1489
0.0667		0.0003		8.9618		0.3292		0.0000		0.0000	
1.0000		44.4438		10.0000		1.2411		9.0000		3.2052	
0.0000		0.0000		0.0000		0.3631		0.0000		0.0000	
1.0000		0.0004		10.0000		0.0704		0.0000		0.0000	
C=10				sigma = 1		sigma = 0.05		poly = 2		poly = 5	
alpha	Total			alpha	Total	alpha	Total	alpha	Total	alpha	Total
0.0000	76.0000			0.0000	76.2552	0.5377	42.7545	0.0000	54.2594	0.0000	47.9100
0.0000				0.0000		0.3213		0.0000		0.0000	
9.0000				8.8908		0.6269		3.8295		6.7459	
0.0000				0.0000		0.5453		0.0000		0.0000	
0.0000				0.0000		0.9401		0.0000		0.0000	
				sigma = 0.5				poly = 3			
alpha	Total			alpha	Total			alpha	Total		
0.0000				0.0000	62.7532			0.0000	53.0684		
0.0000				0.0000				0.0000			
5.9852				0.0000				3.8735			
0.0000				0.0000				0.0000			
0.0000				0.0000				0.0000			