

HW-4 : Supporting Vector Machine

組別27

0310515葉尚昀

0310527李韋辰

0310519陳柏諺

原理:

將data用許多boundaries區分開來，而將與其一定的距離定義為margin的範圍，若對應的class的data超越了那個margin則稱之為error，依照給定的penalty參數大小不同，對應了對misclassification的容忍的不同。

C-SVM

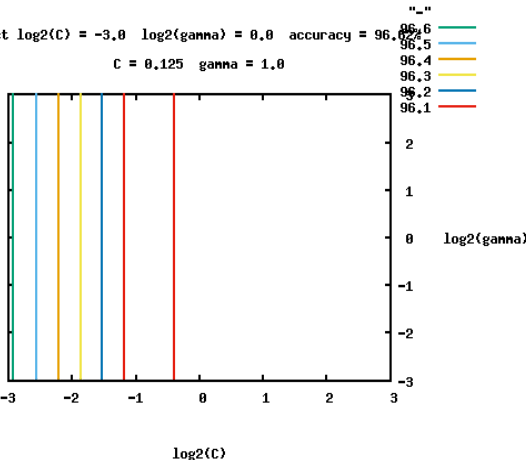
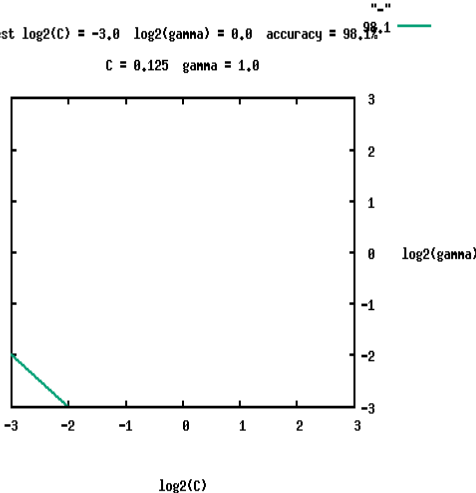
在C-svm裡 C 的大小設定，代表著我們願意讓misclassification 發生的程度。如同在

$$\sum_{n=1}^N E_{\infty}(y(\mathbf{x}_n)t_n - 1) + \lambda \|\mathbf{w}\|^2$$

中，不允許任何misclassification發生，但是否C越高，就代表會切分的越好呢？答案是不一定，會依據每次處理的問題不同而有不同的參數調整，這也是我們在Libsvm中之所以要使用grid的原因，因為它能幫我們找出最好的C的大小和Gamma的參數。其中gamma是在使用rbf和polynomial時的參數。在C-svm的訓練中可以用grid.py來找出最好的參數。

● c-svm 優化參數

對於不同function的參數進行模擬，找到其較優化的參數解。

function	Linear function	Polynomial function with degree 2
最佳參數解	-c cost=0.125	-c cost=0.125 / -g gama=1
結果	<div>Best log2(C) = -3.0 log2(gamma) = 0.0 accuracy = 96.6 C = 0.125 gamma = 1.0</div> 	<div>Best log2(C) = -3.0 log2(gamma) = 0.0 accuracy = 98.1 C = 0.125 gamma = 1.0</div> 

function	Polynomial function with degree 3	Polynomial function with degree 4
最佳參數解	-c cost=0.5 / -g gama=0.03125	-c cost=1 / -g gama=0.03125
結果	<p>Best log2(C) = -1.0 log2(gamma) = -5 accuracy = 97.78% C = 0.5 gamma = 0.03125</p>	<p>Best log2(C) = 0.0 log2(gamma) = -5 accuracy = 96.9% C = 1.0 gamma = 0.03125</p>
function	radial basis function	
最佳參數解	-c cost=0 / -g gama=0.03125	
結果	<p>Best log2(C) = 3 log2(gamma) = -5 accuracy = 98.52% C = 8 gamma = 0.03125</p>	

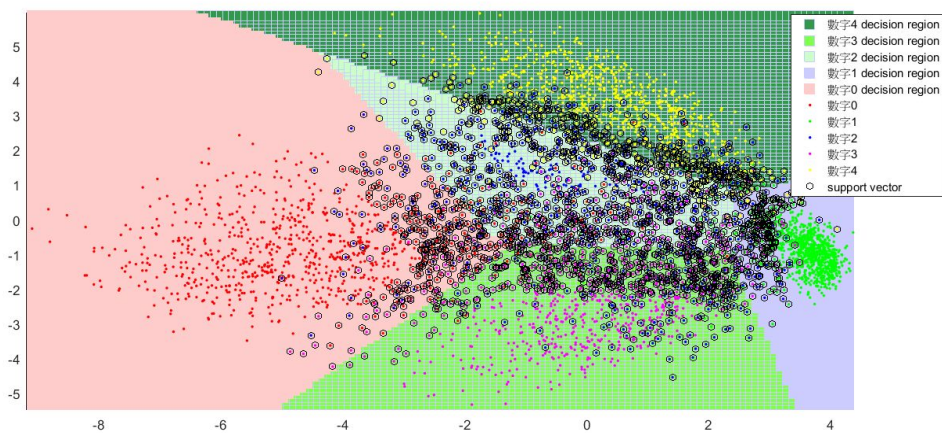
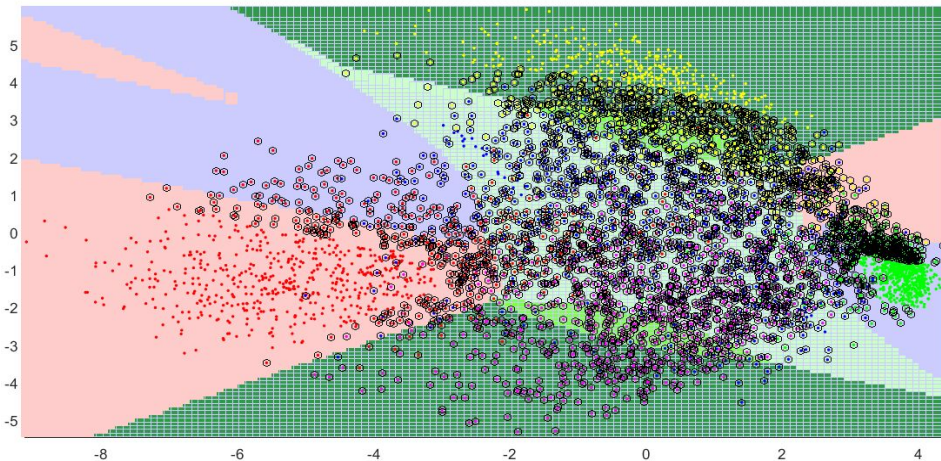
從上方圖得比較可知，使用linear function時，準確率對於cost的影響，由於並無gamma參數，因此準確率的線皆呈垂直狀。而使用polynomial function時隨著項次愈高，準確率的線分裂的程度也愈嚴重，而在所探討的2~4階當中，又以第三階的準確率為最佳。在RBF函數中，可發現c與gamma函數對於準確率影響皆較小，只要在適當的範圍內取直，準確率皆能有穩定的表現。在此五種函數中，發現以RBF函數的效果最好準確率可達到98.52%，其次為二階的多項式函數98.1%，三階的多項式函數97.78%，四階多項式函數96.92%，以最差的線型函數96.62%，以上皆為cross-validation為5做出的最佳結果。

●降維

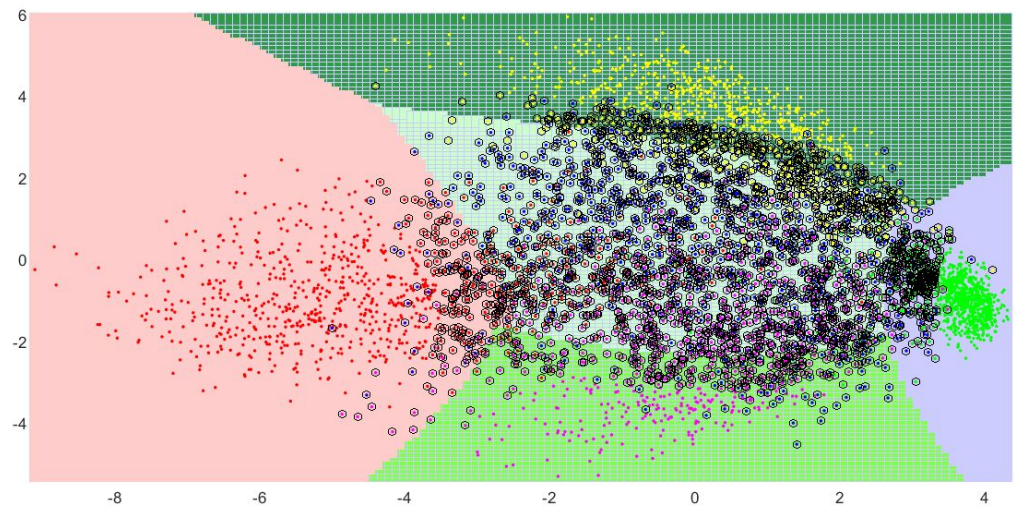
下方為原本的數據X_train 5000*784 matrix經過pca降維成5000*2 matrix，分別代入c-svm的Linear function,Polynomial function 和radial basis function 進行運算和繪圖，所得出來的結果。

- 數字0
- 數字1
- 數字2
- 數字3
- 數字4
- support vector
- 數字4 decision region
- 數字3 decision region
- 數字2 decision region
- 數字1 decision region
- 數字0 decision region

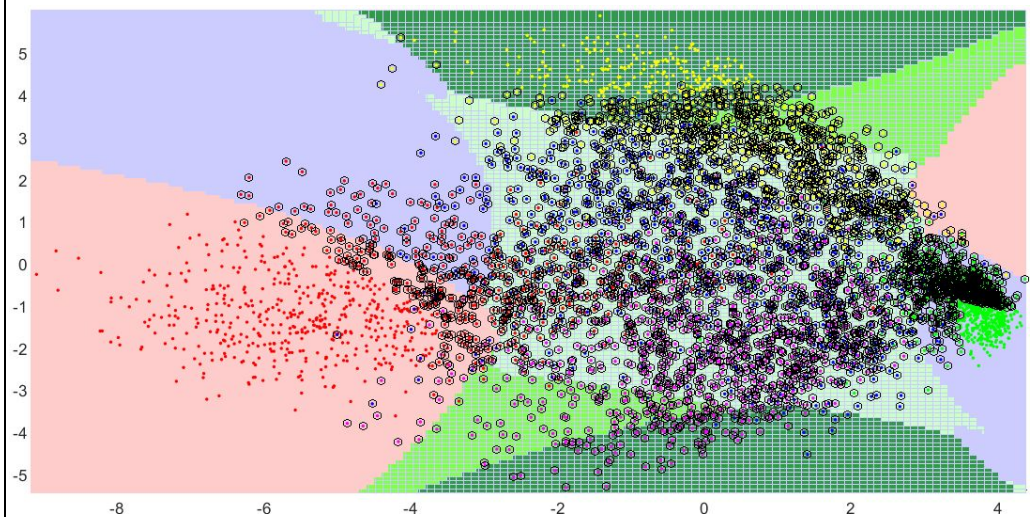
(←左圖為不同數字資料降維後所對應的decision region 和散佈點)

function	結果
Linear function ('s 0 -t 0 -c 0.125 -g 1')	
Polynomial function with degree 2 ('s 0 -t 1 -d 2 -c 0.125 -g 1')	

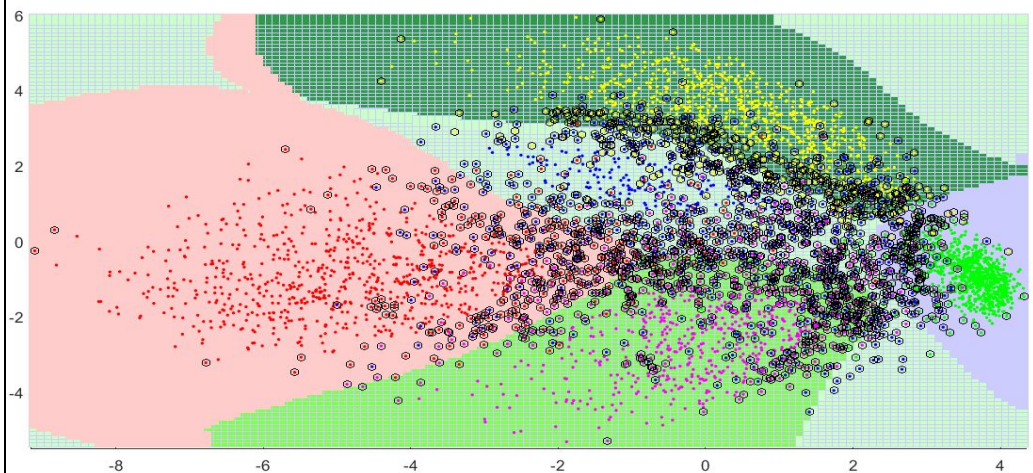
**Polynomial function
with degree 2**
(`'-s 0 -t 1 -d 3 -c 0.5 -g 0.031'`)



**Polynomial function
with degree 2**
(`'-s 0 -t 1 -d 4 -c 1 -g 0.031'`)



radial basis function
(`'-s 0 -t 2 -c 2 -g 0.125'`)

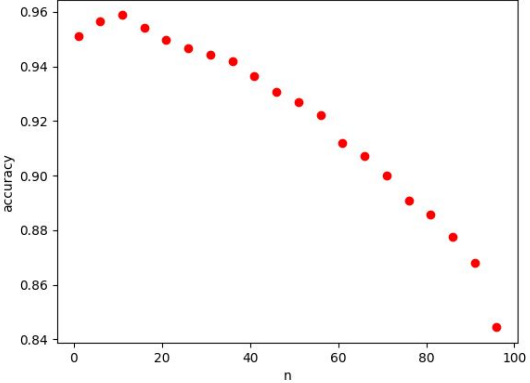
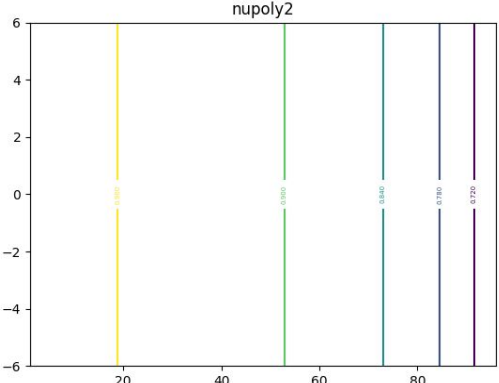
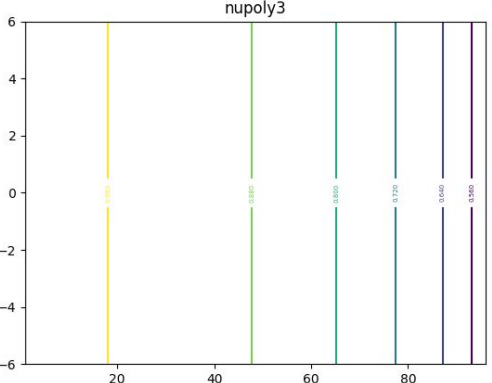
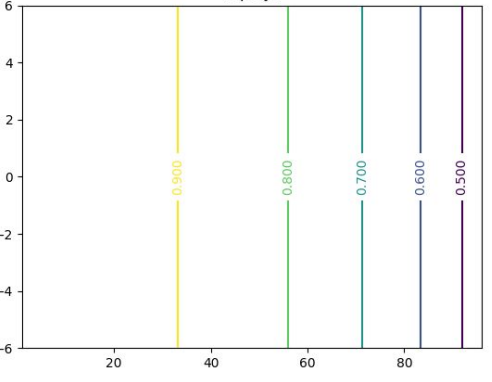


討論:

➤pca降維影響:因為經過pca降維後，只取部分的特徵向量，造成data feature降低和主成分降低，所以在進行svm後，錯誤率相較於以"原本的784維度"高，同時也要儲存較多的support vector;因此在做圖時較不精確。

ν -SVM:

在nu-svm的訓練過程中，因為沒有現有的軟體能夠幫我們找到最適合的參數，於是我們自己寫出了三個程式分別為 [nulinear.py](#) [nupoly.py](#) [nurbf.py](#) 來幫我們找出最好的參數(詳細內容請見壓縮檔裡的tools資料夾)，以下為其輸出結果和作圖。

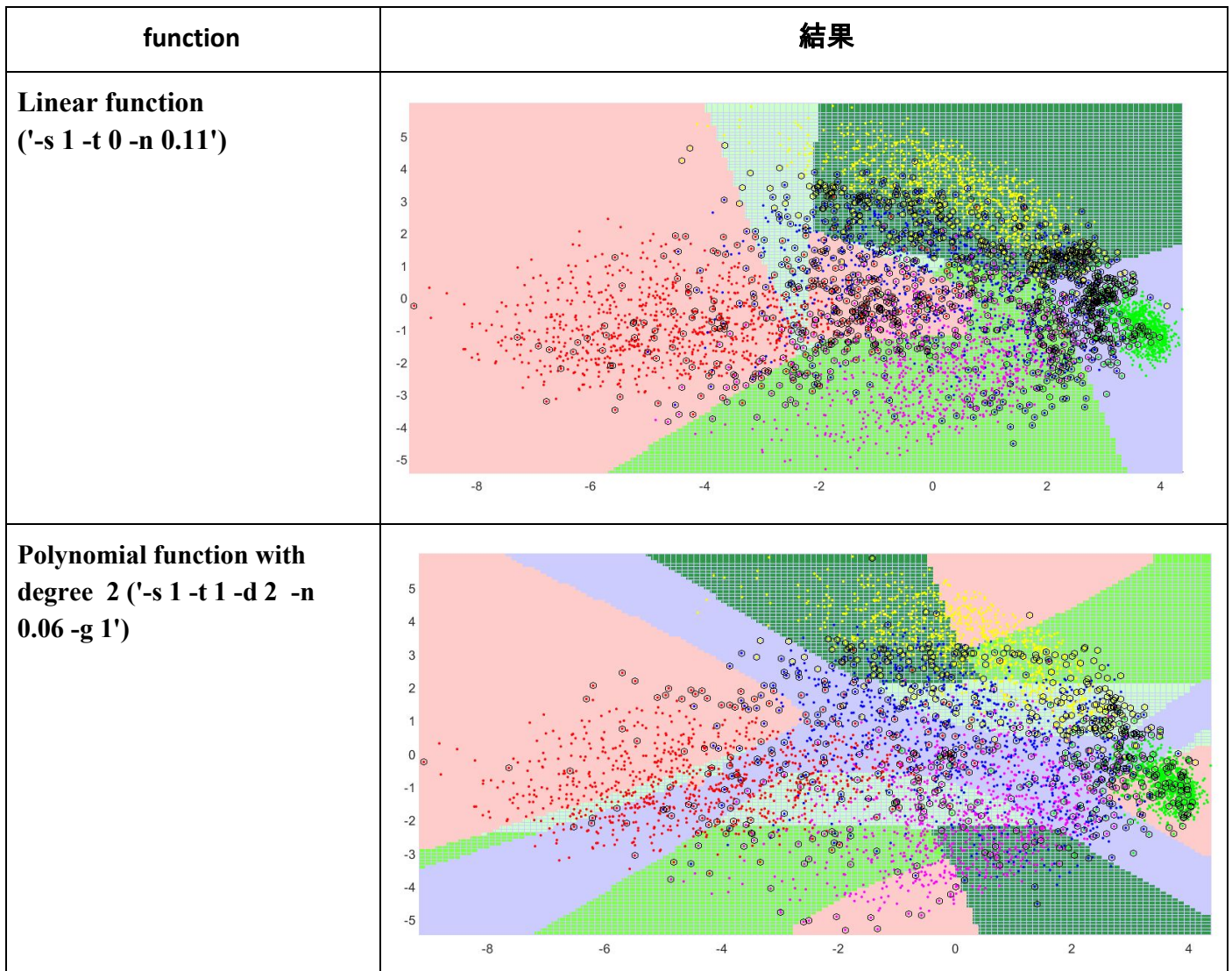
Linear function	Polynomial d=2
	
relation between nu and accuracy. Maximum is around 95.88% at (-s 1 -t 0 -n 0.11)	relation between nu, gamma and accuracy. Maximum is around 97.76% at (-s 1 -t 1 -d 2 -n 0.06 -g 1)
Polynomial d=3	Polynomial d=4
	
relation between nu, gamma and accuracy. Maximum is around 97.76% at (-s 1 -t 1 -d 3 -n 0.06 -g 1)	relation between nu, gamma and accuracy. Maximum is around 96.88% at (-s 1 -t 1 -d 4 -n 0.06 -g 1)

Radius basis function: Best accuracy is around 98.44% when (-s 1 -t 2 -n 0.01 -g 0.015625).

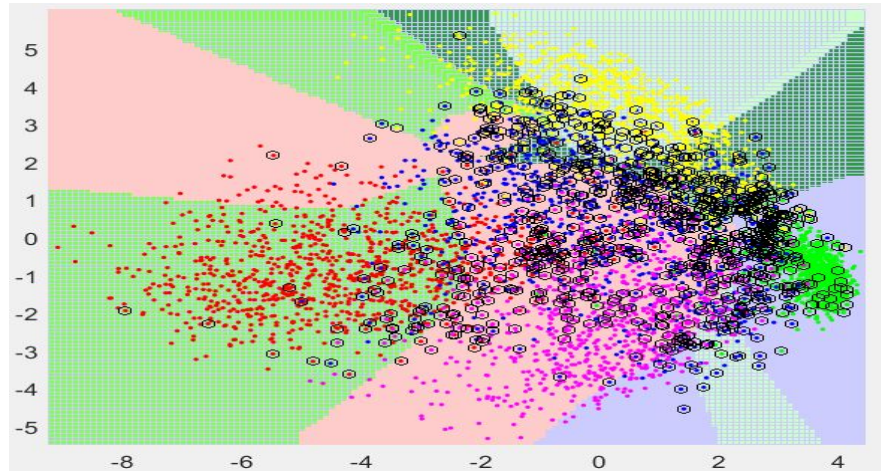
從以上的圖可以發現，gamma參數對nu-svm沒有什麼影響，我認為可能是因為gamma跑的範圍跑到了其飽和區所以到達了平衡，亦或者是nu的效用抑制了gamma所能造成的影響，然而gamma參數在rbf訓練中就有產生一定的影響，其原因為在此gamma影響的是標準差(來不及跑圖出來，但在nurbf.xlsx中有一定的資訊，詳細意義請參閱readme.txt)

- 數字4 decision region
- 數字3 decision region
- 數字2 decision region
- 數字1 decision region
- 數字0 decision region
- 數字0
- 數字1
- 數字2
- 數字3
- 數字4
- support vector

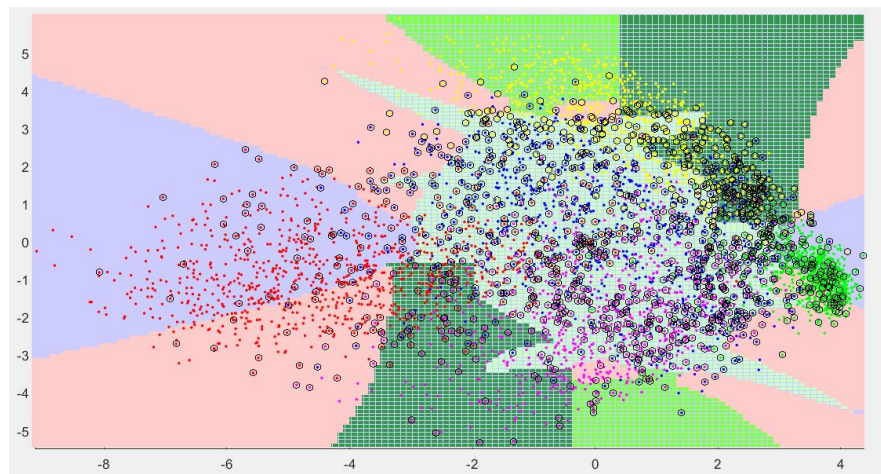
(←左圖為不同數字資料降維後所對應的decision region 和散佈點)



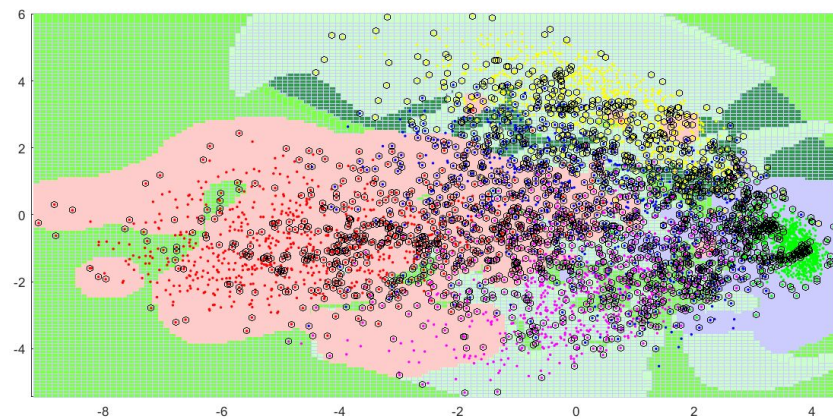
**Polynomial function with
degree 3**
(-s 1 -t 1 -d 3 -n 0.06 -g 1)



**Polynomial function with
degree 4**
(-s 1 -t 1 -d 4 -n 0.06 -g 1)



radial basis function
(-s 1 -t 2 -n 0.01 -g 1/64)



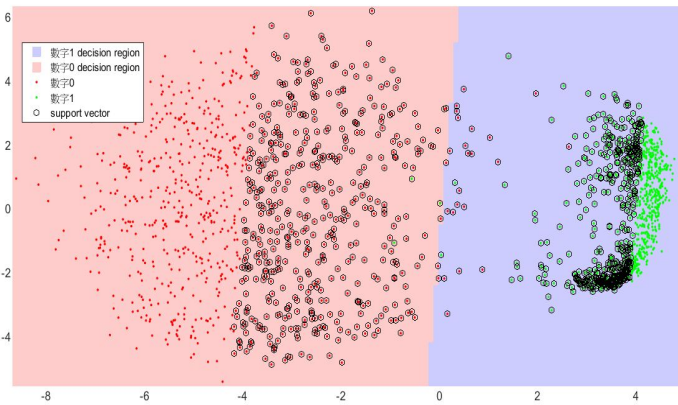
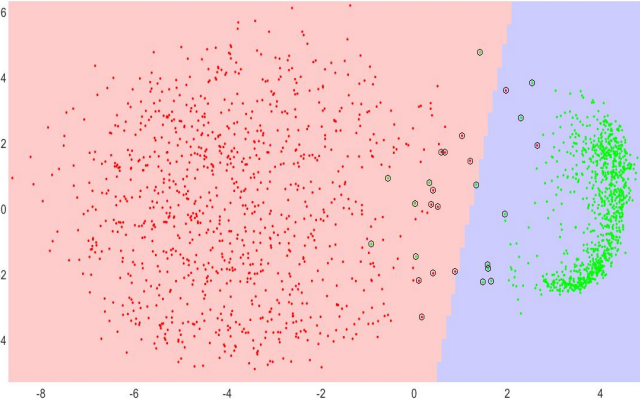
分析與討論：

● C-svm 與 Nu-svm 的不同：

C 跟 Nu 兩項皆為對 Misclassification的penalty，不同點在於 $0 < C$ 而 $0 < \nu < 1$ 。 ν 的意義為對Margin Error的上限和有多少比例的supporting vector的下限。然而C比Nu容易優化，這也是為何網路上找不太到 ν 的optimized program的原因吧。下方圖表為以 c-svm 和 Nu-svm的Linear function 來進行探討，只對數字0和數字1降維後的資料(matrix:2000*2)來train c-svm 和 Nu-svm 的Linear function，藉此觀察隊model 和 suport vector 的影響，同時利用數字0和數字1的testing data 去判別準確率。

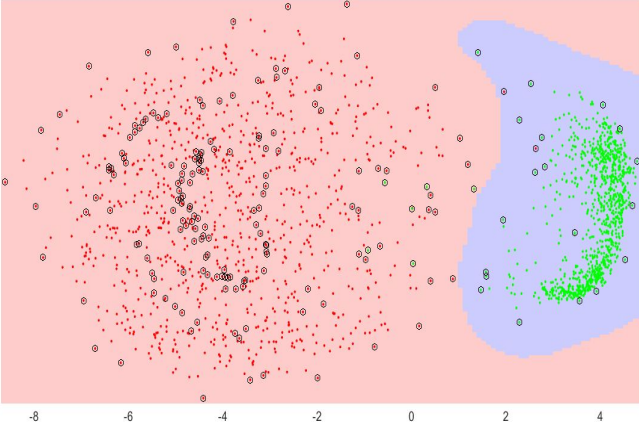
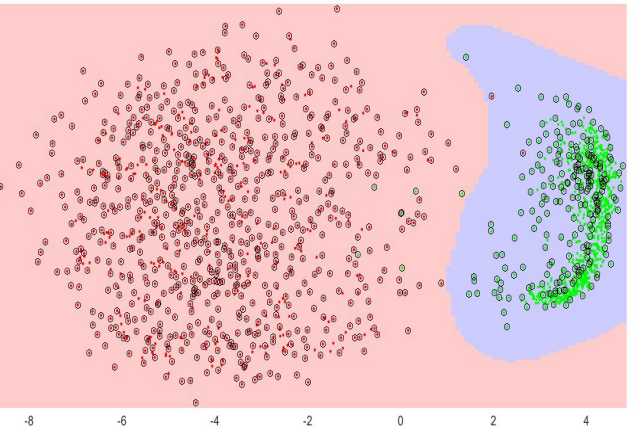
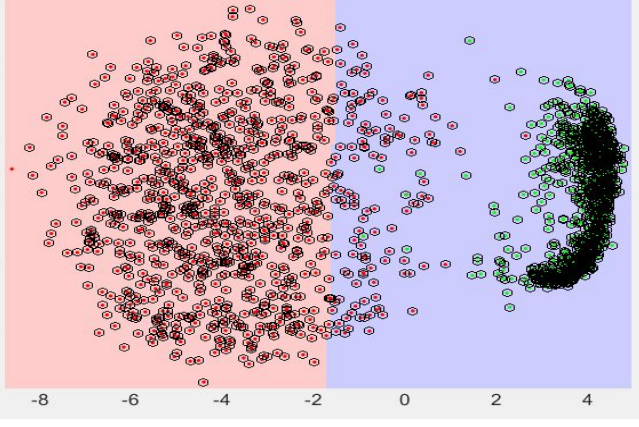
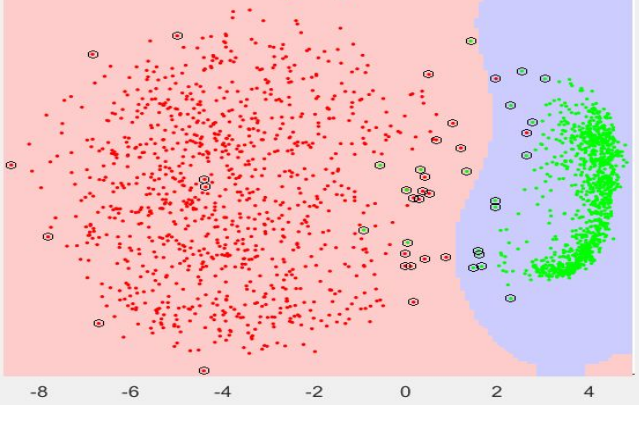


←左圖為數字0,1資料降維後所對應的decision region 和散佈點

Nu-svm(-s 1 -t 0 -c 1)	C-svm(-s 0 -t 0 -c 1)
	
support vector= 1001 Accuracy = 99.7% (997/1000) (classification)	support vector= 27 Accuracy = 99.6% (996/1000) (classification)
討論： >c-svm 和 nu-svm差別 c-svc中的c的範圍為1到無窮大 nu-svm的nu的範圍為0到1，Margin Error的上限和有多少比例的supporting vector的下限 所以如上圖，左方的圖為nu-svm 中默認的nu為0.5,因此在train data =2000筆資料時，所以nu-svm 的support vector =1001(約等於2000*0.5)	

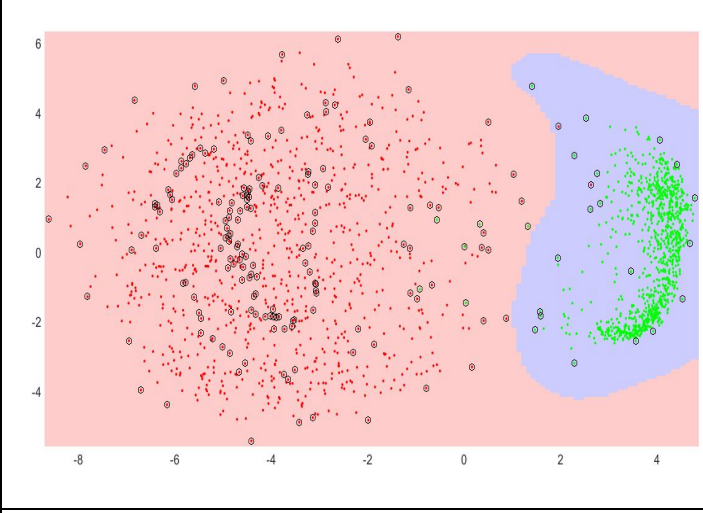
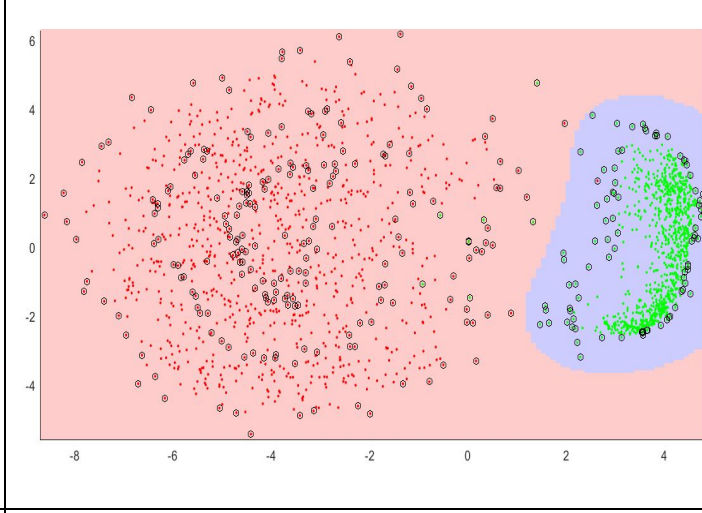
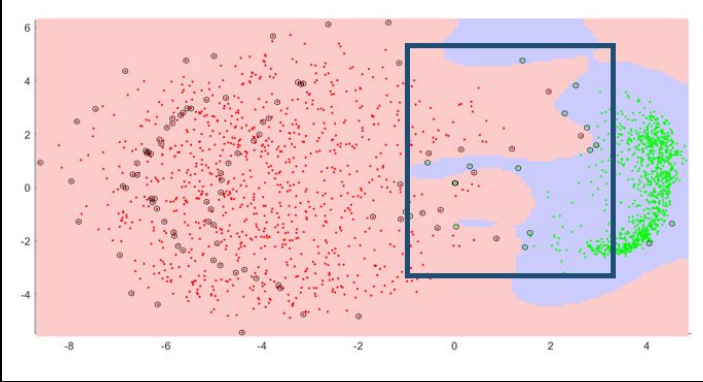
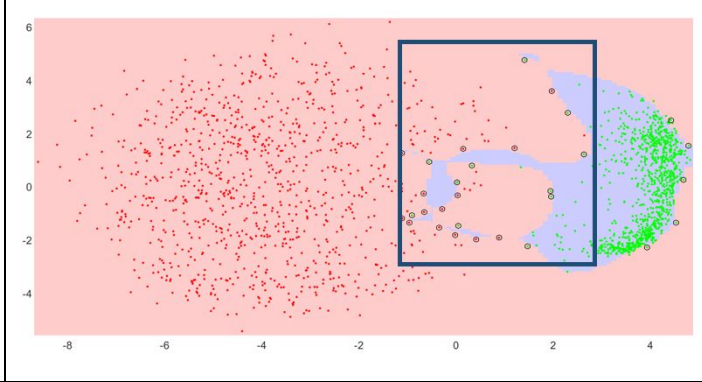
● Gamma參數的意義：

Gamma在 radius basis function中象徵著標準差的大小，將會影響向量對應的高斯標準差的範圍，gamma越大則 σ 越小，代表其分布只會在supporting vector的附近，會造成分類效果很差。下方圖表為以 c-svm 的 radius basis function 來進行探討，只對數字0和數字1降維後的資料 (matrix:2000*2)來train c-svm 的radius basis function，藉由調整Gamma 對於model 和 suport vector 的影響。

<p style="text-align: center;">Gamma=1</p> 	<p style="text-align: center;">Gamma=10</p> 
<p>support vector= 183 Accuracy = 99.6% (996/1000) (classification)</p>	<p>support vector= 904 Accuracy = 99.4% (994/1000) (classification)</p>
<p style="text-align: center;">Gamma=0.00001</p> 	<p style="text-align: center;">Gamma=0.1</p> 
<p>support vector= 1998 Accuracy = 92.4% (924/1000) (classification)</p>	<p>support vector= 45 Accuracy = 99.6% (996/1000) (classification)</p>
<p>討論： >如同gamma參數的定義所說，gamma越大則σ越小，代表其分布只會在supporting vector的附近，所以分布的範圍變小，因此所需要記錄更多的support vector，如同上方的結果gamma=10,1和0.1，當gamma增加時，所需的support vector從45增加到904，而準確率也相對的較低;反之，當gamma過小則σ過大，分布的關係互相影響程度過大，導致模型十分不準確，如上表gamma=0.00001</p>	

● **c cost 對model 的影響:**

以 c-svm 的radius basis function 來進行探討，只對數字0和數字1降維後的資料(matrix:2000*2)來train c-svm 的radius basis function，藉由 調整cost 對於model 和 suport vector 的影響。

<p style="text-align: center;">cost =1(原初始值)</p> 	<p style="text-align: center;">cost =0.1</p> 
<p>support vector= 183 Accuracy = 99.6% (996/1000) (classification)</p>	<p>support vector= 294 Accuracy = 99.5% (995/1000) (classification)</p>
<p style="text-align: center;">cost =100</p> 	<p style="text-align: center;">cost =10000000</p> 
<p>support vector= 103 Accuracy = 99.4% (994/1000) (classification)</p>	<p>support vector= 31 Accuracy = 98.6% (986/1000) (classification)</p>
<p>討論:</p> <p>➢ C (cost)是對於資料誤差所給予的懲罰權重，右方公式中的c便是cost $\min_{w,b,\xi} \frac{1}{2}w^T w + C \sum_{i=1}^l \xi_i$</p> <p>➢ c過小時，表示給與的懲罰權重減少，導致只要資料有誤時造成的影響減少，同時也影響和控制 support vector，因此model 所記錄的support vector 數量增加;如同上方的結果中cost=1和cost=0.1，當 cost從1減少到0.1時，所需的support vector從183增加到294，而準確率也相對的較低。</p> <p>➢ c過大時，表示給與的懲罰權重過大，導致只要資料有誤時造成的影響增大，因此產生overfitting 的問題，從上方表格中cost=100和cost=10000000的圖中的藍色方框可以明顯發現decision region沿著 traing data 產生，而此可得知為overfitting的問題，也因此造成所需的support vector從183減少到31，而準確率也相對的較低。</p>	