

笨蛋也可以用的 **libsvm**

by pavement@cmlab.csie.ntu.edu.tw

主要參考：

piaip 的 (lib)SVM 簡易入門

http://ntu.csie.org/~piaip/svm/svm_tutorial.html

A Practical Guide to Support Vector Classification

<http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>

前言：

因為我微積分很爛，libsvm 的原理都聽不懂，可是我還是想用 libsvm，怎麼辦？還好林智仁老師把 libsvm 包得很好用，還寫了很好的 tutorial，piaip 大帥哥似乎寫了一篇中英對照版的 guide，所以我有點不知道要寫什麼。還是把自己執行的過程貼一貼好了。

準備工作：

這篇假設使用的平台是 Windows XP，從命令列執行。先把一些需要的東西裝好，我們總共需要三個東西：libsvm, python, gnuplot。

Libsvm：

到<http://www.csie.ntu.edu.tw/~cjlin/libsvm/> 下載libsvm，然後解壓縮就好了，這邊假設解到C槽。

Python：

到<http://www.python.org/download/> 下載完直接安裝就好了。

Gnuplot：

下載<ftp://ftp.gnuplot.info/pub/gnuplot/gp400win32.zip> 解壓縮到c:\tmp 這樣就準備好了。

使用說明：

到 C:\libsvm-2.82\windows 下面看看，需要的功能大概就這幾個：

Svmtrain

Svmpredict

Svmscale

Svmtoy

先從 svmtrain 說起，這個指令可以將一組 training data 做成一個 model，最簡單的用法就是不加參數，直接下指令：

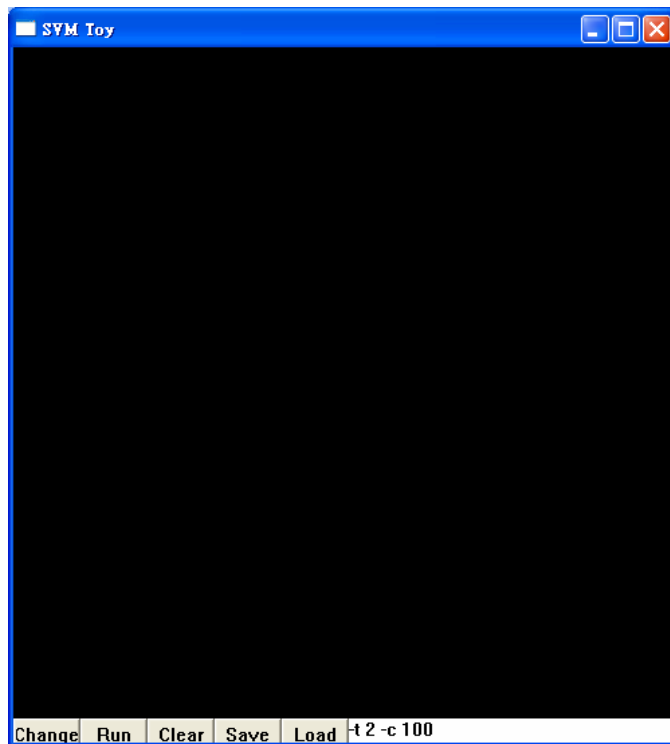
```
C:\libsvm-2.82\windows>svmtrain.exe tdata
```

意思是以 tdata 這個檔案當作 training data，做出一個 model，並輸出成 tdata.model 這個檔案。若是用圖形來觀察也許會比較清楚，這時候就可以用 svmtoy 看看結果，執行

```
C:\libsvm-2.82\windows>svmtoy.exe
```

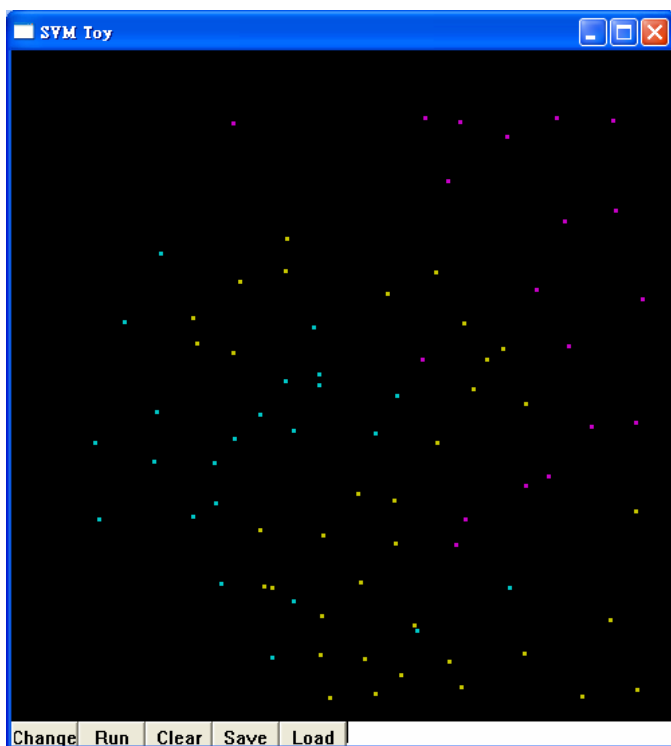
應該會跳一個小視窗出來

圖一：



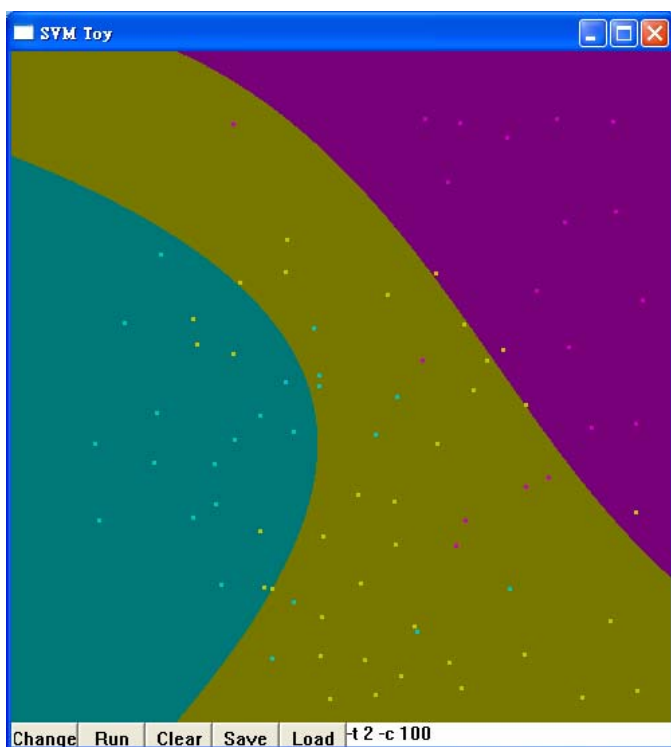
用滑鼠左鍵灑點，”Change”換顏色，可以做出下面的圖二。

圖二：



按下"Run"就會跑出以下的圖。

圖三：



可以發現原先三種顏色的點被分區了，這邊所有的點就是我們的 **training data**，而 **model** 記錄的就是點的分區狀況。把 **training data** 存起來，再用文字編輯器打開看看，格式長的像這樣：

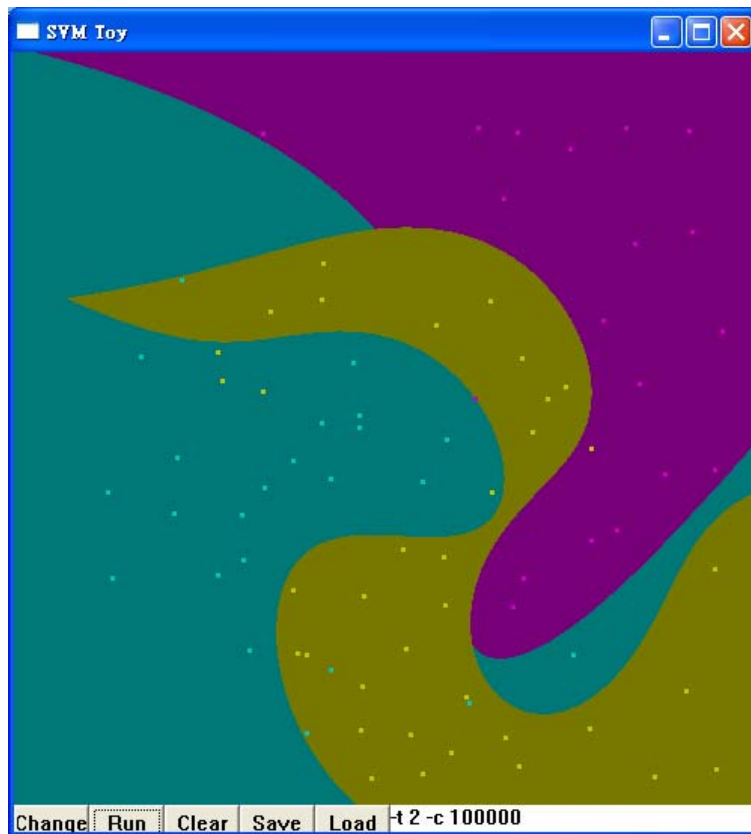
```
1 1:0.386000 2:0.902000
1 1:0.602000 2:0.862000
1 1:0.310000 2:0.792000
1 1:0.302000 2:0.672000
2 1:0.568000 2:0.668000
2 1:0.764000 2:0.524000
2 1:0.514000 2:0.658000
.....
2 1:0.890000 2:0.846000
2 1:0.928000 2:0.684000
2 1:0.650000 2:0.908000
2 1:0.540000 2:0.956000
3 1:0.610000 2:0.458000
3 1:0.780000 2:0.354000
3 1:0.828000 2:0.438000
3 1:0.938000 2:0.368000
3 1:0.898000 2:0.236000
3 1:0.648000 2:0.192000
3 1:0.736000 2:0.126000
```

挑其中一行來看

```
3 1: 0.736000 2: 0.126000
```

冒號的前後分別代表 **feature** 的編號及數值，即，第一個 **feature** 的值為 0.736，第二個 **feature** 的值為 0.126。開頭的 3 代表這個點屬於第三個分類。從圖上看來，兩個 **feature** 分別為 X 軸及 Y 軸，而分類代表顏色。換句話說，座標在(0.736, 0.126)的點是黃色。做出來的 **model** 可以當作圖上的顏色分區。有了分區後，隨便指定一個座標，就可以得到對應的顏色了。所以一個好的 **model**，應該要切得很乾淨，黃色區域最好不要出現紫色的點。從這裡看來圖三切的有點差，改一下參數就可以切的好很多。例如參數改成 `"-t 2 -c 100000"`，切出圖四：

圖四：



所以要切的好，參數就要下的好。還好林智仁老師有寫好的 tool 幫我們試參數，不用自己手動試啦。首先，到 `C:\libsvm-2.82\tools` 底下找 `grid.py`，然後 copy 到 `C:\libsvm-2.82\windows`，接著執行：

```
C:\libsvm-2.82\windows>python grid.py tdata
```

就會看到一堆數據和圖在亂飆，不用怕，那是在暴力試參數。整個跑完之後，去最後一行找參數：

```
512.0 8.0 89.6104
```

前兩個分別是 `c` 跟 `g`，後面那個不要管他就好了。用這組參數 `train` 一次看看：

```
C:\libsvm-2.82\windows>svmtrain -c 512.0 -g 8.0 tdata
```

可以得到 `tdata.model`，再用這個 `model` 跟原來的 `training data` 比比看切得乾不乾淨，使用 `svmpredict` 這個指令：

```
C:\libsvm-2.82\windows>svmpredict.exe tdata tdata.model tdata.out
```

```
Accuracy = 97.4026% (75/77) (classification)
```

```
Mean squared error = 0.025974 (regression)
```

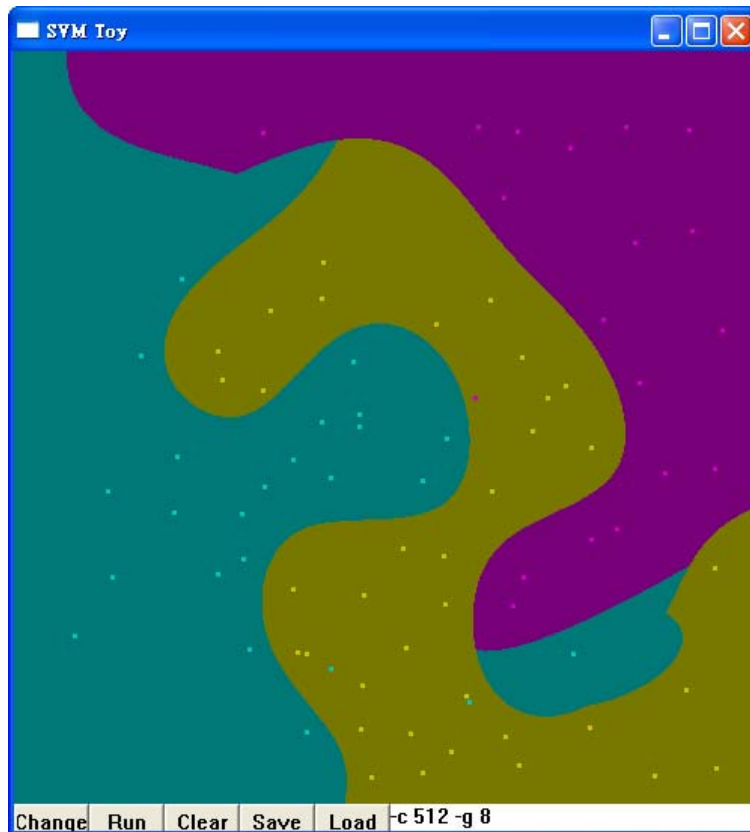
```
Squared correlation coefficient = 0.95108 (regression)
```

可以看到 accuracy 已經很好了，比起不下參數作出來的結果：

Accuracy = 64.9351% (50/77) (classification)

從圖上來看到也有很明顯的改善：

圖五：



Svmscale:

svmscale 是用來調整 feature 的大小範圍，以免有某一項 feature 值太大，在算距離時主導了結果。通常我們將取值的範圍訂在 0~1 或 -1~1，注意，training data 與 test data 都必須作相同程度的 scaling。

用法：

```
C:\libsvm-2.82\windows>svmscale.exe tdata > tdata.scale
```

可以將做完 scaling 的新資料寫到 tdata.scale 裡，預設是 scale 到 -1~1。

同時 scale test data 以及 training data 的方法：

```
C:\libsvm-2.82\windows>svmscale.exe -s scale trainingdata > trainingdata.scale
```

```
C:\libsvm-2.82\windows>svmscale.exe -r scale testdata > testdata.scale
```

林智仁老師有一套建議的 classify 流程：

1. 把資料轉成 libsvm 看的懂得格式
2. Scaling
3. 選用效能較好的 RBF kernel (預設值就是 RBF，所以不用管這一行。)
4. 用 cross validation 選擇較好的參數 (就是 grid.py 作的事)
5. 用剛剛找到的參數來 train model
6. Test

以 a1a 這個 dataset 為例，a1a 為 training data，a1a.t 為 test data：

```
C:\libsvm-2.82\windows>svmscale -s scale a1a > a1a.scale
C:\libsvm-2.82\windows>svmscale -r scale a1a.t > a1a.t.scale
C:\libsvm-2.82\windows>python grid.py a1a.scale
8.0 0.0078125 83.4891
C:\libsvm-2.82\windows>svmtrain -c 8.0 -g 0.0078125 a1a.scale
C:\libsvm-2.82\windows>svmpredict a1a.t.scale a1a.scale.model a1a.t.out
Accuracy = 83.9869% (25999/30956) (classification)
Mean squared error = 0.640522 (regression)
Squared correlation coefficient = 0.301888 (regression)
```

上頁這一連串的命令其實，可以換成一個 script：

```
C:\libsvm-2.82\windows>python easy.py a1a a1a.t
```

所以呢，真的要 classify 的話，只要一行 code 就搞定了。那我前面寫這麼多幹嘛？其實我寫到一半才發現這個好東西，發現之後就不知道寫什麼了，那就寫到這了。果然笨蛋也可以用 libsvm 吧。