# Psychoinformatics - Week 9 (Exercises)

by 許維仁 (r10h41014@ntu.edu.tw)

In [ ]:
```
import numpy as np
from sklearn import *
from sklearn import model_selection
from matplotlib.pyplot import *
%matplotlib inline
```

## 1 檢查 machine learning pipeline (8 points)

### 1.1 請打亂原本的Y觀察正確率是否和chance level (0.33)有差異? 若有, why? (4 points)

In [ ]:
```
# 本題在研究打亂X和打亂Y有差別嗎?
from sklearn import datasets, metrics, neighbors


iris = datasets.load_iris()
X=iris.data
Y=iris.target
Y2=np.random.permutation(Y)
clf=neighbors.KNeighborsClassifier(1)
clf.fit(X,Y2)
accuracy=np.mean(clf.predict(X)==Y2)
print('Accuracy of K = 1, X predict random Y',accuracy)


clf2=neighbors.KNeighborsClassifier(90)
clf2.fit(X,Y2)
accuracy2=np.mean(clf3.predict(X)==Y2)
print('Accuracy of K = 90, X predict random Y',accuracy2)
```

```
Accuracy of K = 1, X predict random Y 0.9933333333333333
Accuracy of K = 90, X predict random Y 0.37333333333333335
```

從本題我們可以發現，由於我們是在predict已經看過的答案(考古題)，當K=1時，我們對於任一一個X都可以從KNN的Model裡面找到相對應的考古答案，所以正確率會幾乎全對，而不是像隨機亂猜的0.33，除非我們將K的數值拉的很大(90)，它就會參考X最接近的90個X預測的Y，幾乎可以說是亂猜了。 打亂X與打亂Y是一樣的結果，除非打亂X的時候我們不是用permutation的方式打亂，而生成新的random_X，像這種情況，兩者就會不一樣，因為此時的random_X已經和原本的X不一樣了。

### 1.2 請用母數或無母數統計檢定以下accuracies中的結果是否和chance level (0.5)有差異? 若有, why? (4 points)

In [ ]:
```
Y=np.remainder(range(200),2)
print(Y) #Y的0和1個數一樣多
```

```
[0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1]
```

In [ ]:
```
# 跑一百次測試:
from sklearn import svm


clf=svm.SVC()
accuracies=[]
for i in range(100):
 X=np.random.rand(200,2) # X取亂數
 kf=model_selection.KFold(len(Y),shuffle=True) # Leave-one-out cross-validation
 sc=model_selection.cross_val_score(clf,X,Y,cv=kf)
 accuracies.append(sc.mean())
```

In [ ]:
```
# Please do your statistical tests here:
from scipy.stats import ttest_1samp
ttest_1samp(accuracies,0.5)
```

Out[ ]:
```
Ttest_1sampResult(statistic=-2.212010537625113, pvalue=0.02926504355653675)
```

In [ ]:
```
print(sum(accuracies)/100)
```

```
0.48200000000000015
```

In [ ]:
```
#非Leave one out (balanced training dataset)
accuracies=[]
for i in range(100):
 X=np.random.rand(200,2) # X取亂數
 kf=model_selection.KFold(2,shuffle=False) #False, so the training set is always balanced
```

```
    sc=model_selection.cross_val_score(clf,X,Y,cv=kf)
    accuracies.append(sc.mean())
    # Please do your statistical tests here:
from scipy.stats import ttest_1samp
print('Fold: 2 (balanced training data)')
ttest_1samp(accuracies,0.5)
```

```
Fold: 2 (balanced training data)
```

Out[ ]:

```
Ttest_1sampResult(statistic=-0.484834701057972, pvalue=0.6288646535231612)
```

Please write your discussion here, if any.

我們可以看到P-value < 0.05 ，也就是說accuracy是顯著的不等於0.5，從下方我們算出accuracy的平均約為0.4949，會有這樣低於0.5的accuracy是因為當我們在training的時候使用LOOCV的方式使得training data的Y會呈現 "49個1、50個0" 或是 "50個1、49個0" 兩種imbalance狀況， 而由於這些data全是random的，所以在兩種狀況中，預測結果為Y數量比較多的值將會在training時得到比較低的error，但恰好testing data中的Y是數量少的那一個，使得Model猜錯的機會增加，因而造成正確率低於0.5。

另一方面可以看到，如果我CV使用Fold=2(可使training data balanced)，我們可以看到p-value 大於0.05，所以我們不拒絕accuracy=0.5的假說。

### Please submit your notebook in PDF to NTU Cool by next Friday (11/10).