CSE569

PROJECT 2

Experimenting with SVM

Nitheese Thirumoorthy

ASU ID:1225417336

Importing Libraries

```
import numpy as np
import pandas as pd
import scipy.io
from libsvm.svmutil import *
import os
```

Loading the data

```
train=scipy.io.loadmat("trainData.mat")
X1_train=train['X1']
X2_train=train['X2']
X3_train=train['X3']
Y_train=train['Y']

test=scipy.io.loadmat("testData.mat")
X1_test=test['X1']
X2_test=test['X2']
X3_test=test['X3']
Y_test=test['Y']
```

STEP 0

```
#training the SVM
Y_train=[i[0] for i in Y_train]
Y_train=np.asarray(Y_train)
Y_test=[i[0] for i in Y_test]
Y_test=np.asarray(Y_test)
```

part1

```
#Training and testing with X1_train,X1_test
prob=svm_problem(Y_train,X1_train)
param=svm_parameter('-c 10 -t 0')
m=svm_train(prob,param)
label_11,acc_11,pval_11=svm_predict(Y_test,X1_test,m)
```

```
Accuracy = 10.7807% (203/1883) (classification)
```

```
#Training and testing with X2
prob=svm_problem(Y_train,X2_train)
param=svm_parameter('-c 10 -t 0')
m=svm_train(prob,param)
label_12,acc_12,pval_12=svm_predict(Y_test,X2_test,m)
```

```
Accuracy = 16.6755% (314/1883) (classification)
```

```
#Training and testing with X3
prob=svm_problem(Y_train,X3_train)
param=svm_parameter('-c 10 -t 0')
m=svm_train(prob,param)
label_13,acc_13,pval_13=svm_predict(Y_test,X3_test,m)
```

```
Accuracy = 8.92193% (168/1883) (classification)
```

```
print("Accuracies obtained in Step-0 Part-1 :")
print("For X1_test: ",str(acc_11[0])+"%")
print("For X2_test: ",str(acc_12[0])+"%")
print("For X3_test: ",str(acc_13[0])+"%")
```

```
Accuracies obtained in Step-0 Part-1 :
For X1_test:  10.780669144981413%
For X2_test:  16.675517790759425%
For X3_test:  8.921933085501859%
```

part 2

```
# Calculating posterior probability accoring to X1
prob=svm_problem(Y_train,X1_train)
param=svm_parameter('-t 0 -c 10 -b 1 -q')
m=svm_train(prob,param)

label_21,acc_21,pval_21=svm_predict(Y_test,X1_test,m,'-b 1')
```

```
Accuracy = 27.1907% (512/1883) (classification)
```

```
# Calculating posterior probability accoring to X2
prob=svm_problem(Y_train,X2_train)
param=svm_parameter('-t 0 -c 10 -b 1 -q')
m=svm_train(prob,param)
```

```
label_22,acc_22,pval_22=svm_predict(Y_test,X2_test,m,'-b 1')
```

```
Accuracy = 28.0404% (528/1883) (classification)
```

```
# Calculating posterior probability accoring to X3
prob=svm_problem(Y_train,X3_train)
param=svm_parameter('-t 0 -c 10 -b 1 -q')
m=svm_train(prob,param)

label_23,acc_23,pval_23=svm_predict(Y_test,X3_test,m,'-b 1')
```

```
Accuracy = 28.6245% (539/1883) (classification)
```

```
print("Accuracies obtained in Step-0 Part-2 :")
print("For X1_test: ",str(acc_21[0])+"%")
print("For X2_test: ",str(acc_22[0])+"%")
print("For X3_test: ",str(acc_23[0])+"%")
```

```
Accuracies obtained in Step-0 Part-2 :
For X1_test:  27.190653212958043%
For X2_test:  28.040361125862983%
For X3_test:  28.624535315985128%
```

STEP 1

```
#Feature Combination by fusion of Classifiers.
combined_probability=[]

#Combining the 3 classifiers by probability fusion.
for i in range(len(pval_23)):
    temp=[]
    for j in range(50):
        p=(pval_21[i][j]+pval_22[i][j]+pval_23[i][j])/3
        temp.append(p)
    combined_probability.append(temp)

#Assigning it to the label with the highest posterior probability and
counting the number of correct predictions.
correct=0
for i in range(len(Y_test)):
    Class=1+
combined_probability[i].index(max(combined_probability[i]))
    if Class==Y_test[i]:
        correct+=1

print("Accuracy obtained in Step 1 by fusion of classifiers is ",
(correct/len(Y_test))*100,"%")
```

```
Accuracy obtained in Step 1 by fusion of classifiers is
44.503451938396175 %
```

STEP 2

```
#Concatenating all the three features and then performing train and
test using SVM.

X_train=[]
#Concatinating X1_train,X2_train,X3_train
for i in range(len(X1_train)):
    x=np.concatenate((X1_train[i],X2_train[i],X3_train[i]),axis=None)
    X_train.append(x)


X_test=[]
#Concatinating X1_test,X2_test,X3_test
for i in range(len(X1_test)):
    x=np.concatenate((X1_test[i],X2_test[i],X3_test[i]),axis=None)
    X_test.append(x)


#Training on the new feature-combined dataset(X_train).
prob=svm_problem(Y_train,X_train)
param=svm_parameter('-c 10 -t 0')
m=svm_train(prob,param)


#Testing on the new feature-combined dataset(X_test).
label_2,acc_2,pval_2=svm_predict(Y_test,X_test,m)

print("Accuracies obtained in Part-2 :")
print("For X_test: ",str(acc_2[0])+"%")
```

```
Accuracy = 37.0685% (698/1883) (classification)
Accuracies obtained in Part-2 :
For X_test:  37.06850770047796%
```