Support Vector Machines

Ishita Jaju
16BCE1059

1 SVM for my dataset

```
from sklearn import svm
clf = svm.SVC(gamma='auto')
clf.fit(X_train, y_train)
```

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

```
y_pred=clf.predict(X_test)
```

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

```
0.7358490566037735
```

The accuracy for classification comes out to be 73%.

2 Pipeline of SVM parameters

2.1 Varying C

C is the penalty parameter of the error term. The results don't really change on varying C.

```
C = [0.0000001,1000000]
for c in C:
    clf_ = svm.SVC(C=c,gamma='auto')
    clf_.fit(X_train, y_train)
    y_pred=clf_.predict(X_test)
    print("For C =",c,", accuracy =",accuracy_score(y_test,y_pred))
```

```
For C = 1e-07 , accuracy = 0.7358490566037735
For C = 1000000 , accuracy = 0.7358490566037735
```

2.2 On varying gamma

Gamma is the kernel coefficient in the case of the kernel being 'rbf', 'poly', or 'sigmoid'. Its default value is given as $\frac{1}{n}$ where n is the no. of features.

```
gamma = [0.001,0.1,1,100,3000]
for g in gamma:
    clf = svm.SVC(gamma=g)
    clf.fit(X_train, y_train)
    y_pred=clf.predict(X_test)
    print("For gamma =",g,", accuracy =",accuracy_score(y_test,y_pred))
```

```
For gamma = 0.001 , accuracy = 0.7358490566037735
For gamma = 0.1 , accuracy = 0.7358490566037735
For gamma = 1 , accuracy = 0.7358490566037735
For gamma = 100 , accuracy = 0.7358490566037735
For gamma = 3000 , accuracy = 0.7358490566037735
```

2.3 On varying kernel

The kernel is the type of kernel function to be used in the algorithm.

```
kernel = ['linear','poly','sigmoid']
for k in kernel:
    clf = svm.SVC(kernel=k,gamma='auto')
    clf.fit(X_train, y_train)
    y_pred=clf.predict(X_test)
    print("For kernel =",k,", accuracy =",accuracy_score(y_test,y_pred))
```

```
For kernel = linear , accuracy = 0.9811320754716981
For kernel = poly , accuracy = 0.9811320754716981
For kernel = sigmoid , accuracy = 0.7358490566037735
```

3 SVM for Regression

Using the automobiles dataset, which predicts the price of a car based on features like make, fuel type, number of doors, engine type, number of cylinders

etc.
After some preprocessing,

```
y_pred = clf.predict(X_test)
```

```
clf.predict(X_test)
clf.score(X_test,y_test)
```

```
-0.15946670587632417
```

The best possible score is 1.0 and our score comes out to be negative, because the model here is arbitrarily worse.