SVM_classification

March 10, 2019

In [1]:
```
#support-vector machines
#first example
from sklearn.datasets import load_svmlight_file
X_train, y_train = load_svmlight_file('ijcnn1.bz2')
first_rows = 5000
X_train, y_train = X_train[:first_rows,:], y_train[:first_rows]
```

In [2]:
```
import numpy as np
from sklearn.model_selection import cross_val_score
from sklearn.svm import SVC
hypothesis = SVC(kernel='rbf', random_state=101, gamma='scale')
scores = cross_val_score(hypothesis, X_train, y_train, cv=5, scoring='accuracy')
print('SVC with rbf function -> accuracy in corss validation:\nmean= %f\nstandard devia
      %(np.mean(scores), np.std(scores)))
```

```
SVC with rbf function -> accuracy in corss validation:
mean= 0.903800
standard deviation= 0.000354
```

In [3]:
```
#second example
import pickle
covertype_dataset = pickle.load(open('covertype_dataset.pickle','rb'))
covertype_X = covertype_dataset.data[:50000,:]
covertype_Y = covertype_dataset.target[:50000] -1
```

In [4]:
```
import numpy as np
covertypes = ['Spruce/Fir','Lodgepole Pine', 'Ponderosa Pine', 'Cottonwod/Wollow', 'Asp
print("Original data set: ", covertype_dataset.data.shape)
print("Sample: ", covertype_X.shape)
print("Frequency of target values: ", list(zip(covertypes, np.bincount(covertype_Y))))
```

```
Original data set:  (581012, 54)
Sample:  (50000, 54)
Frequency of target values:  [('Spruce/Fir', 18161), ('Lodgepole Pine', 24335), ('Ponderosa Pi
```

In [5]:
```
from sklearn.model_selection import StratifiedKFold
from sklearn.svm import LinearSVC
```

```
        hypothesis = LinearSVC(dual=False, class_weight = 'balanced')
        cv_strata =  StratifiedKFold(n_splits=5, shuffle=True, random_state=101)
        scores = cross_val_score(hypothesis, covertype_X, covertype_Y, cv=cv_strata, scoring='a
        print('LinearSVC with rbf function -> accuracy in corss validation:\nmean= %f\nstandarc
              %(np.mean(scores), np.std(scores))) # problem seems to be not linear but we use
```

```
LinearSVC with rbf function -> accuracy in corss validation:
mean= 0.670960
standard deviation= 0.007295
```

```
In [6]: # Optimalization
        from sklearn.model_selection import GridSearchCV
        hypothesis = SVC(kernel='rbf', random_state=101, gamma='scale')
        search_dict = {'C':[0.001,0.01,0.1,1,10,100,1000],
                       'gamma':[1,0.1,0.01,0.001,0.0001]}
        search_funct = GridSearchCV(estimator=hypothesis, param_grid=search_dict,scoring='accu
                                    refit=True, cv=5)
        search_funct.fit(X_train, y_train)
        print("Best params: %s" % search_funct.best_params_)
        print("accuracy of cross-validation: mean = %f" %(search_funct.best_score_))
```

```
Best params: {'C': 1000, 'gamma': 0.1}
accuracy of cross-validation: mean = 0.997800
```