Weekly Report **Abhijeet Ghodgaonkar**
4379 Lab Date: 11/07/18

1 Problem Statement

To completely perform SVM and Random Forest on Open Datasets Indian Pines , Pavia University and Kennedy Space Center.

2 Tasks

- Implementing GridSearchCV() (and k-fold cross validation) to find the optimal hyperparameters in SVM and Random Forest Classifier.
- Observe the nature of Discriminant functions in SVM
- Perform k-fold cross validation selectively on the data
- Dump/Pickle the optimal Model and indices of the class-wise segregated dataset into binary files , so that it need not be retrained again for a different dataset.

3 Solutions

Using SVM and Random Forest, the first task was to see the accuracy metrics by training the classifier functions by train test splitting the data.
The main objectives here are to overcome the random splitting of data which is not suited for the hyperspectral datasets , and to see the ideal train test split ratio in order to minimize the variance in accuracy.
The next step to fine tune the training is to use k-fold cross validation and gridsearch to search for the optimal parameters.

4 Code Implementation

Polynomial Kernel SVM

In [9]:
```
poly_p = svm.SVC(C=1 ,kernel='poly' ,cache_size=9000,degree=3, gamma=1)
poly_p.fit(ix_train,iy_train)
poly_p.score(ix_train,iy_train)
ipred1=poly_p.predict(ix_test)
ipred1
```

Out[9]:
```
array([ 1,  1,  1, ..., 16, 16, 16], dtype=uint8)
```

In [10]:
```
accuracy_score(iy_test, ipred1)
```

Out[10]:
```
0.8190327613104524
```

In [11]:
```
save_classifier = open ("IndianPines_LinearSVM.pickle","wb")
pickle.dump(linear_I,save_classifier)
save_classifier.close()
```

In [12]:
```
save_classifier = open ("IndianPines_PolySVM.pickle","wb")
pickle.dump(poly_p,save_classifier)
save_classifier.close()
```

Using Grid Search , varying C from 1 to 10 , and kernel linear and RBF

In [14]:
```
from sklearn.model_selection import GridSearchCV
parameters = {'kernel':('linear', 'rbf'), 'C':[1, 10]}
svc = svm.SVC()
clf = GridSearchCV(svc, parameters)
```

In [16]:
```
1
```

Out[16]:
```
SVC(C=1, cache_size=9000, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma=1, kernel='poly',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

In [4]:
```
save_classifier = open ("poly_k.pickle","wb")
pickle.dump(linear_I,save_classifier)
save_classifier.close()
```

5 Analysis & Further Work

5.1 Analysis

- The accuracy with train-test split is less as compared to K-fold CV. The optimal value of k is 10.
- Doing GridSearchCV on SVM is easier to do than on Random Forest, due to the complicated parameters of the function 'sklearn.ensemble.RandomForestClassifier'.

5.2 Further Work

- To fine tune parameters for Random Forest Classifier
- To use sklearn.pipeline to ensure that the model gets trained and its parameters gets updated in a pipeline , without having to feed inputs into outputs, and outputs into inputs.

Final Evaluation

The classical methods are good only when the hyperparameters are fine-tuned, and the data is trained properly.
This method will not be good for original remote sensing data as the boundaries or rules cannot be learned properly from the data, and neural network models have to be used to do classification.