M6 Lab: Model Selection with K-Fold Cross Validation

Start Assignment

- Due Sunday by 11:59pm
- Points 20
- Submitting a file upload

In this lab we will learn how to perform K-Fold Cross Validation using python's SciKit Learn library.

Dataset

We will use the a9a dataset in this lab. It is a sample of the US Census Income Dataset formatted as a libsvm file. Use this link to download the a9a file. You don't need to download the a9a.t file.

- **https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#a9a (https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#a9a)**

Please keep in mind that this is a relatively large dataset. The code will take minutes to run and give you results. You can use this alternative smaller dataset if your machine is old and is taking longer time to run the code:

- **https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#a1a (https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#a1a)**

Part I: Understanding the Model Parameters

We will use a Support Vector Machine Classifier in this lab. This classifier has many parameters that can be fine tuned. The full list of parameters is found on this page:

- **https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html (https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html)**

We will focus on this set of parameters:

- C
- kernel
- degree
- gamma
- coef0
- tol
- verbose
- max_iter
- decision_function_shape
- random_state

Using the documentaton page provided above, read about each of these parameters and explain what it does.

Part II: Simple Cross Validation

In this part we will split the dataset into a training set and a testing set. Then we will perform cross validation on the training set to train the model. That is, the training set will be randomly split into k folds (parts). Then the fitting function (which builds the model) will be repeated k times. Each time k-1 folds are used for fitting the model, and the last fold is used to validate the learning process. That is, it is used to check for overfitting and quit the fitting process once overfitting begins to occur.

Here is the documentation page that explains the cross validation function:

- **https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html (https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html)**

Also, this page explains how k-fold cross validation is implemented in the Scikit-learn machine learning library. Notice that the k-folds are used for validation and parameter fine tuning and only one test set is used to evaluate the model's generalization.

- **https://scikit-learn.org/stable/modules/cross_validation.html (https://scikit-learn.org/stable/modules/cross_validation.html)**

The code for this part should perform the following steps:

1. read the libsvm data file
2. define a SVM-Classify object
3. call the cross validation function to fit the SVM classifier using some k cross validation counts.
4. print the mean & standard deviation of the obtained classification accuracies from the multiple fits.

The code should look something like this:

```
from sklearn.datasets import load_svmlight_file
from sklearn import svm
from sklearn.model_selection import cross_val_score

print("Loading Dataset...")
X,y = load_svmlight_file("a9a")
print("Creating classifier object...")
clf = svm.SVC(kernel='linear', C=1, random_state= 42)
print("Training classifier with cross validation, k=5")
scores = cross_val_score(clf, X, y, cv=5)
print("Training Complete!")
acc = scores.mean()
stdiv = scores.std()
print("Cross Validation Mean Accuracy = %0.2f" % acc )
print("Standard Deviation of the Mean Accuracy across all runs = %0.2f" % stdiv)
```

- Run this code. What is the obtained result? **Note: This code will take a few minutes to run. If your machine is old, it might take longer. You can use the a1a dataset instead of a9a if your machine is taking forever to run the code.**
- Modify the code to use an rbf kernel with gamma = 0.1. What is the obtained accuracy?
- Modify the code to use an rbf kernel with gamma = 0.01. What is the obtained accuracy?
- Modify the code to use a polynomial kernel with a degree = 2. What is the obtained accuracy?

Part III: Parameter Fine Tuning

Now that we learned how to perform cross validation, we are ready to experiment with different parameter values, and find out which parameter set gives the best results.

Using a support vector classifier, SVC, we can experiment with different regularization values C. We can also experiment with different kernels: linear vs poly vs rbf, and with different parameter values for each kernel. For example, with rbf, we can experiment with gamma values = 0.1, 0.001, 0.0001, etc. Same with poly kernel and different values for the degree parameter.

This can be accomplished in SciKit Learn using a Parameter Grid combined with the Grid Search Cross Validation method.

The algorithm for performing parameter tuning is:

1. Read the dataset
2. Create a parameter grid
3. Create a classifier object
4. Call the Grid Search Cross Validation method
5. Show obtained results for all parameter sets.

```
from sklearn.datasets import load_svmlight_file
from sklearn import svm
import pandas as pd
from sklearn.model_selection import GridSearchCV
print("Loading Dataset...")
X,y = load_svmlight_file("a9a")
print("Creating Parameter Grid...")
param_grid = [
 {'C': [1, 10], 'kernel': ['linear']},
 {'C': [1, 10], 'gamma': [0.001, 0.01], 'kernel': ['rbf']},
 ]
print("Creating classifier object...")
svc = svm.SVC()
print("Creating a grid search cross validator object...")
clf = GridSearchCV(svc, param_grid)
print("Fitting the models with different parameters...")
clf.fit(X, y)
print("Writing all fitting results...")
df = pd.DataFrame(clf.cv_results_)
df.to_csv("Parameter_Tuning_Results.csv")
```

Run this code & check the output file. Notice that it will take more time to run than the previous part. You can use the sample dataset if your machine is taking a very long time to run that code.

- What is printed in the output file?
- What does every column represent?
- What does every row represent?
- Which run gave you the best outcome?
- Modify the code to include testing with the polynomial kernel with degrees 2 & 3 along with regularization values C 0.01, 0.1, 1, 10. What are the obtained testing errors for these parameter sets?
- Modify the code to go through the results data frame and prints out the ranks of the different parameter sets, along with the obtained mean accuracies & standard deviation. The output for this part should look something like this:
  - Rank 1: {'C': 10, 'kernel': 'linear'}, Mean Test Accuracy=0.828660436137071, Mean StdDev=0.0200928542013671
  - Rank 2: {'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}, Mean Test Accuracy=0.827414330218068, Mean StdDev=0.0234453429560458
  - .
  - .
- Check the pandas.DataFrame.sort_values() function as it can be used in sorting the results before printing them.

Extra Credit: K-Fold Cross Validation with Different Testing Folds (10 points)

Notice the scikit-learn implementation of K-Fold cross validation is different from what was discussed in the lecture slides. In scikit-learn, one portion of the data is always used for testing while the rest is used for training and validation. This might result in an inaccurate estimate for the overall average classification accuracy, specifically when the data points that are hard to classify end up being part of the testing set, which is completely hidden from the training process.

In this extra credit activity, you will implement the standard K-Fold cross validation we discussed.

- Let K=5
- Randomly split the data into 5 equal portions.
- Run the SVM linear model training process 5 times with each time using a different fold for testing, and the remaining 4 folds for training and validation. Calculate the average accuracy and print it.
- Repeat this process for SVM RBF kernel for gamma = 0.01 once and for gamma = 0.001 another time. Use the default value for C in all runs. No need to finetune the C parameter.
- Print the average accuracy over the 5 folds for each of the three models. Which model gives the highest accuracy?

Lab Submission

Insert your answers to Part I, your code as text, screenshot of your code, and screenshots of the obtained results. For the dataframe results stored to file, you can open the file in excel or sublime (or any other text editor) and take a screenshot of the file content.