Project 3: More Classification

Due: April 24, 2019

EE/CS5841: Machine Learning

Project Summary

In this assignment you will apply classification algorithms to data derived from real-world experiments.

1 Part 1: Acquire and Partition Data (5 points)

In this part you will download a data file, load it into Python or Matlab, and partition it into a *training* and *testing* set. The data we are using in this assignment comes from the UCI Machine Learning Dataset Repository which contains many datasets for benchmarking machine learning algorithms. Specifically, we will use the *Sonar* dataset. For more information click here.

1.1 Getting the Data via Python

If you are using Python download the file named "sonar5841.dat" and load into Python using the code snippet below. Make sure to update the directory in the code to match where you saved the data.

```
import numpy as np
data = np.loadtxt('/dir/of/data/sonar5841.dat')
X = data[:,0:-1] # should be 208 by 60
Y = data[:,-1] # should be 208 long
```

1.2 Getting the Data via Matlab

If you are using Matlab download the file named "sonar5841.mat" and load into Matlab using the code snippet below. Make sure to update the directory in the code to match where you saved the data.

```
load('/dir/of/data/sonar5841.mat')
```

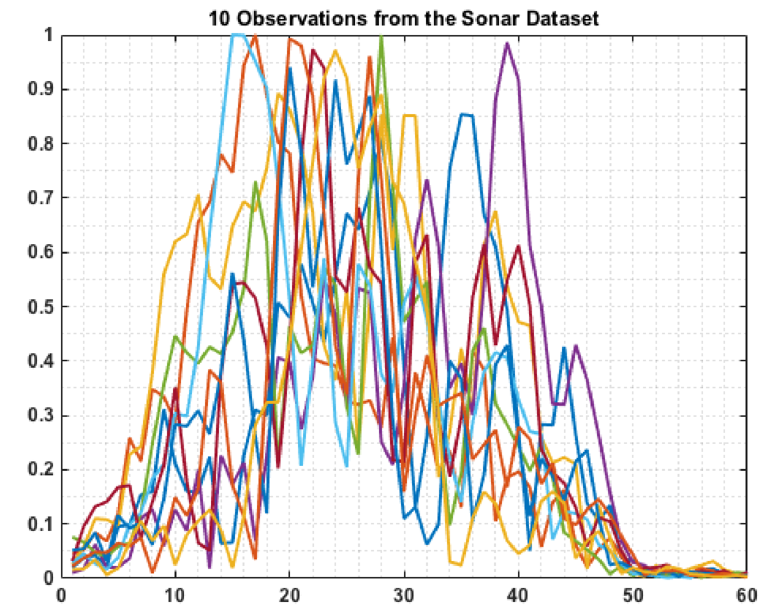

Figure 1: 10 Observations from the Sonar Dataset

1.3 Partitioning the Data

After performing the above actions you should have 2 variables in your workspace: X and Y. X is the data matrix where rows are observations; note that each observation has dimension 60. Y contains the class labels, either -1 or +1. There are 208 observations in this dataset.

Partition the testing data - Sequester 20% of the data for *testing* by randomly selecting 42 rows (without replacement) from the data matrix X and the respective class labels in Y.

Partition the training data - The remaining 80% (166 rows) of X should become the *training* set. **Remember to standardize all the data based on the statistics of the training set.**

1.4 Part 1 Deliverables

Create two plots: one showing the 42 testing samples and the other showing the 166 training samples. An example of 10 observations is shown below in Figure 1. Your plots should look similar but with many more traces in them.

2 Part 2: Apply Classifiers (25 points)

In this part you will apply the classifiers in Table 1 to the data and compare their results.

Logistic Regression
Linear SVM
Kernel SVM
KNN

Table 1: Classifiers to Evaluate

2.1 Logistic Regression (5 points)

Apply logistic regression (LR) to the data: train the LR model using the testing set, and test the LR model on the testing set. You may use built-in functions to implement LR. **Report the testing accuracy of the LR model.**

2.2 Linear SVMs (10 points)

Apply a soft-margin linear SVM to the data: train the SVM using the testing set, and test the SVM on the testing set. Recall the soft-margin SVM has a tunable parameter (C in the SVM class notes). You should use 5-fold cross-validation to find the optimal value of C for these data. **Report the value of C you determined and report the testing accuracy. Create a plot of the cross-validation accuracy vs. the parameter C (this plot should have a maximum at the value you found as optimal for C). You may use built-in SVM functions but you must write your own 5-fold cross-validation code.**

You may choose to use built-in cross-validation code for this problem and the following problem for a penalty of 5 points.

2.3 Kernel SVM (10 points)

Apply a kernel SVM to the data using a Gaussian (RBF) kernel. Recall the RBF kernel's width is a tunable parameter. Use 5-fold cross-validation to find the optimal RBF width. **Report the RBF width you determined and report the testing accuracy. Create a plot of the cross-validation accuracy vs. the width parameter γ (this plot should have a maximum at the value you found as optimal for γ).You may use built-in kernel SVM functions but you must write your own 5-fold cross-validation code.**

You may choose to use built-in cross-validation code for this problem and the previous problem for a penalty of 5 points.

2.4 Part 2 Deliverables

- A table of the test accuracies from the classifiers you implemented above

  Also note in the table the best values you found for C and γ

  Indicate which classifier is best for these data?

- A plot of the cross-validation accuracy vs. the parameter C for the linear SVM

  This plot should have a maximum at the value of C you determined to be optimal

- A plot of the cross-validation accuracy vs. the parameter γ for the kernel SVM

  This plot should have a maximum at the value of γ you determined to be optimal

3 Submissions

Your submission should include a PDF with all deliverables clearly labeled. You must also submit all your code. This assignment may be completed using either Matlab or Python. Jupyter notebooks are acceptable, however, please print to PDF for your submission.