Data Analytics and Big Data

Final Project

Comparing Classifiers

Aarzoo Chopra
400421878

Jiayan Xu
400405265

INTRODUCTION

This project looks at two different approaches to classification: Fisher's Linear Discriminant Analysis and Linear Support Vector Machine.

Each method is used to classify and test the same data set, and the results are compared. The dataset chosen is the Banknote Authentication Data Set from the UCI Machine Learning Repository.

sklearn is used to implement the classifiers.

DATA SET

The Banknote Authentication Data Set is a multivariate dataset. It comprises of the data extracted from images taken to authenticate banknotes.

It has 1372 instances, each with 5 attributes, namely:

- Variance of Wavelet Transformed Image
- Skewness of Wavelet Transformed Image
- Curtosis of Wavelet Transformed Image
- Entropy of Image
- Class

All the attributes are real numbers, and there are 2 classes: genuine banknotes and forged banknotes.

From this dataset, 75% of the data points (1029) are used for training the classifiers, while the remaining 25% (343) are used for testing it.

FISHER'S LINEAR DISCRIMINANT ANALYSIS (LDA)

The Fisher's Linear Discriminant Analysis projects high-dimensional data points onto a line. This line has the characteristics that the distance between the means of two classes is maximized, while the variance within each class is minimized. Once the optimal line is created, the algorithm performs the classification in this one-dimensional space.

LINEAR SUPPORT VECTOR MACHINE (SVM)

The Support Vector Machine algorithm looks for the data points closest to both groups. These points are called support vectors. Then, it computes the distance between each data point, and

finds the optimal line or hyperplane that separates two classes of data points by their maximized margin.

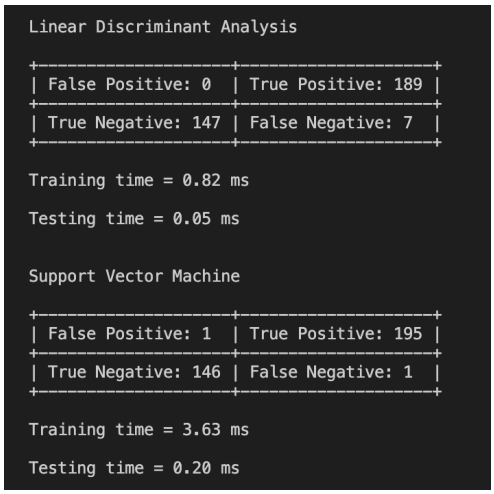
PROCEDURE

The following steps are used to implement the classifiers:

- Data is read from the text file into a numpy array.
- As the original data is organized according to class, the rows are shuffled to make it random.
- Data is split into test data (75%) and training data (25%).
- The columns are split to separate the other attributes from the class for both test and training data.
- LDA and SVM classifiers are implemented using sklearn. After initializing both the classifiers separately, the training data is used to train the classifiers (clf.fit()), and the test data is used to test them (clf.predict()).
- A timer from the timeit module is used to get the computational times for training and testing for each classifier.
- The output of testing is used to build a confusion matrix.
- The computational times and the confusion matrices are displayed (using PrettyTable).
- There is a function to validate the results in the confusion matrix. It gives the total number of elements in class A and B (test data). These can be used to make sure all the elements are accounted for in the confusion matrix. The function call for the check is currently commented out.

RESULTS

As we can see from the output, the training and testing times for LDA are less than those for SVM for this dataset. LDA take about a quarter of the time that SVM takes. On running the code multiple times, even though the actual times had slight variations, this outcome remained the same.



From the confusion matrices, we can see that the number of true positives is higher for SVM than LDA. This also stayed consistent over multiple attempts.

Since there are very few errors in both the cases, we can conclude that this dataset is well-suited to a linear classifier, and either of the two, LDA or SVM, would be a good fit. If computational speed is a priority, then LDA would be a better choice. However, if accuracy is more important, then SVM would be preferable.

REFERENCES

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.