**Data Analytics and Big Data**

**Final Project**

**Comparing Classifiers**

**Aarzoo Chopra**

**400421878**

**Jiayan Xu**

**400405265**

In the Fisher’s Linear Discriminant Classifier Training and Support Vector Machine Classifier Training, we used a dataset from the website UCI Machine Learning Repository: banknote authentication Data Set. The dataset we chose is banknote authentication.

The owner of the dataset is Volker Lohweg (University of Applied Sciences, Ostwestfalen-Lippe, volker.lohweg '@' hs-owl.de).

The donor of the dataset is Helene DÃ¶rksen (University of Applied Sciences, Ostwestfalen-Lippe, helene.doerksen '@' hs-owl.de).

The data received by UCI Machine Learning Repository is August 2012.

This dataset contains five real-time attributes:

1. variance of Wavelet Transformed image(continuous).

2. skewness of Wavelet Transformed image (continuous).

3. curtosis of Wavelet Transformed image (continuous).

4. entropy of image (continuous).

5. class (integer).

The total number of instances associated with this classifier training process is 1372, of which 1029 data points (75% of total data points) are used for training and 343 data points (25% of total data points) are used for testing purposes.

Classifier 1: Fisher’s Linear Discriminant

The fisher’s linear discriminant classification method 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. Once the optimal line is created, the algorithm performs the classification in this one-dimensional space.

Here are the following steps we used to implement fisher’s linear discriminant classifier using SK-learn library:

1. Import data from txt file and save it in a 1372 by 5 matrix.
2. Split data rows into two groups: 1029 Training data (75%) and 343 Testing data (25%)
3. Split two data groups into four sub-groups: Training data attributes, Training data class, Testing data attributes, and Testing data class
4. Use SK-learn build-in training method(clf.fit()) to train the classifier.
5. Use SK-learn build-in testing method(clf.predict()) to test the classifier.
6. Use the timer method imported from timeit library to record the computation time for training and testing period.

Classifier 2: Support Vector Machine

1.In the code, we

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.

Here are the following steps we used to implement support vector machine classifier using SK-learn library:

1. Import data from txt file and save it in a 1372 by 5 matrix.
2. Split data rows into two groups: 1029 Training data (75%) and 343 Testing data (25%)
3. Split two data groups into four sub-groups: Training data attributes, Training data class, Testing data attributes, and Testing data class
4. used SK-learn build in training method(svm.SVC(kernel=’linear’)) to create a linear support vector machine.
5. Use SK-learn build-in training method(clf.fit()) to train the classifier.
6. Use SK-learn build-in testing method(clf.predict()) to test the classifier.
7. Use the timer method imported from timeit library to record the computation time for training and testing period.

Text

Description automatically generated