CSE 572: DATA MINING

Assignment 4
Spring 2018

GROUP 2

Dwarsala Venkata Manoj Reddy	1213185010
Dipika Ramaswamy	1213237517
Kiran Teja Settipalli	1213019858
Rohit Polisetti	1212933824
Srinivas Puranam	1211168267

Table of Contents

1. Data Preparation:

For this assignment, we worked on data collected by all 37 users. The new feature matrix obtained in assignment 2 is used for training and testing the three machines. Data collected from 10 users has been used for training the machines. The data from the remaining users has been used for testing the machines. Since we have 10 different gestures, we created 10 training data matrices and 10 test data matrices.

Data for Training:
For each gesture, data from new feature matrix of 10 users has been taken and given a class label '1'. Then, we picked data points belonging to the same 10 users from the other 9 gestures and assigned them a class label '0'. Since the number of no-class data points will be very high compared to yes-class, we picked only few data points from the remaining 9 gestures to avoid **Class Imbalance problem**. Now, this entire data will be used to train the machines for that particular gesture. The similar steps are repeated for all the other gestures.

Data for Testing:
For each gesture, we have taken data of the remaining 27 users from new feature matrix and given them a class label '1'. Then, we picked data points of these same set of users from the other 9 gestures and assigned them a class label '0'. Again, we picked only few data points from the remaining 9 gestures. Now, this entire data will be used to test the machines for that particular gesture. The similar steps are repeated for all the other gestures.

Since test data is combined together from the 27 users, we reported the overall accuracy metrics instead of user-wise metrics. The three accuracy metrics - Precision, Recall and F1-Score have been computed for each of the 10 Decision Trees, 10 Support Vector Machines and 10 Neural Networks. The values are reported in the corresponding sections below.

2. Decision Tree:

Decision Tree is a supervised learning model primarily used for classification and regression analysis. This model forms a tree structure based on the training data supplied. While the tree is incrementally being developed, the data is broken down into smaller and smaller subsets. Given a set of training examples where each data-point is assigned to one of the two classes, decision tree builds a model such that it can classify the data points. When the test data is passed to this decision tree model, it tries to categorize the data into one of the two classes.

The following steps have been performed in this phase to classify data using Decision Tree.

- In the first step of this phase, the data collected from 10 users has been used for training and the data collected from remaining 27 users has been used for testing.
- All the data from each user has been accumulated into a single file for each category or class.

- Then, in the matlab code, Decision Tree model is built based on the training data and its class labels using the inbuilt matlab function fitctree().
- Once the decision tree model has been built, we use the matlab inbuilt function predict() which takes the testing data and the built model as input.
- The predict() function then predicts and outputs the labels for the testdata.
- These labels are then used to determine 3 types of accuracy metrics, Precision, Recall and F1 Score by constructing the confusion matrix.

The accuracy measures are calculated using the following formulae,

Precision:
This is used to measure the fraction of relevant data from a group of newly predicted labels. The values of precision for each class are calculated and displayed in the command window of matlab.

Precision = (True-positive) / (True-positive + False-positive)

Recall:
This is used to measure the fraction of relevant data that have been correctly classified by the Decision Tree. The Recall values are displayed in the command window of matlab.

Recall = (True-positive) / (True-positive + False-negative)

F1 Score:
In binary classification, this is used to test the accuracy of Decision Tree using the calculated values of Recall and Precision. The F1-score values for each class have been calculated and displayed in the command window. The F1-score is calculated using the following formula:

F1-Score = Harmonic Mean(Precision and Recall) = 2*Precision*Recall/(Precision+Recall)

Accuracy Metrics for each class using Decision Tree:

Class	Precision	Recall	F1-Score
About	0.5471	0.2929	0.3815
And	0.8061	0.7979	0.8020
Can	0.8148	0.6666	0.7333
Cop	0.3500	0.2828	0.3128
Deaf	0.4705	0.1616	0.2406
Decide	0.7903	0.4949	0.6086
Father	0.5619	0.6868	0.6181
Find	0.7551	0.7474	0.7512
Go Out	0.4642	0.3939	0.4262
Hearing	0.5308	0.4343	0.4777

3. Support Vector Machine:

SVM is a supervised learning models mainly used for classification and regression analysis. Given a set of training examples where each data-point is assigned to one of the two classes, SVM builds a model based on the training data. Then, test data is passed to the SVM model which tries to categorize the data into one of the two classes as accurately as possible.

The following steps have been performed in this phase to classify data using SVM.

- In the first step of this phase, the data collected from 10 users has been used for training and the data collected from remaining 27 users has been used for testing.
- All the data from each user has been accumulated into a single file for each category or class.
- In the matlab code, SVM model is built based on the training data and its labels using the matlab inbuilt function fitcsvm().
- Once, the SVM model has been built, we use the matlab inbuilt function predict() which takes the testing data and the built model as input.
- The predict() then predicts and outputs the labels for the testdata.
- These labels are then used to determine 3 types of accuracy metrics, Precision, Recall and F1 Score, by calculating true-positive, false-positive and false-negative.

Similar to decision trees, the accuracy measures are calculated using the following formulae,

Precision:
This is used to measure the fraction of relevant data from a group of newly predicted labels. The values of precision for each class are stored in the PrecisionVals[] matrix of matlab.

Precision = (True-positive) / (True-positive + False-positive)

Recall:
This is used to measure the fraction of relevant data that have been correctly classified by the SVM. The Recall values are stored in the RecallVals[] matrix of matlab.

Recall = (True-positive) / (True-positive + False-negative)

F1 Score:
In binary classification, this is used to test the accuracy of SVM using the calculated values of Recall and Precision. The F1-score values for each class have been calculated and stored in F1scores[] matrix of matlab. The F1-score is calculated using the following formula.

F1-Score = Harmonic Mean(Precision and Recall) = 2*Recall*Precision/(Recall+Precision)

Accuracy Metrics for each class using SVM:

Class	Precision	Recall	F1-Score
About	0.5253	0.8300	0.6434
And	0.6750	0.8100	0.7364
Can	0.4667	0.6300	0.5362
Cop	0.4219	0.5400	0.4737
Deaf	0.6333	0.7600	0.6909
Decide	0.7051	0.5500	0.6180
Father	0.6016	0.7400	0.6637
Find	0.7217	0.8300	0.7721
Go Out	0.2604	0.2500	0.2551
Hearing	0.6514	0.7100	0.6794

4. Neural Network:

Neural networks classifiers are modeled similar to the network of neurons in the human brain. They are made up of an input layer, an output layer and one or more hidden layers consisting of one or more hidden nodes. They learn from the dataset by predicting class labels for records, which are then used to calculate the error between the actual class and the predicted class values. This error is propagated backward through all the layers to adjust the weights of the edges to nodes in the hidden layer.

There are many types of neural networks such as feedforward, convolutional neural networks and recurrent neural networks. Here, we have used a feedforward neural network for classification with the following characteristics:

- Number of hidden layers - 10
- Activation function - Tangent sigmoid function for hidden layers and Linear function for output layer.
- Number of nodes in each hidden layer - 10
- Performance function - Mean Square Error

The steps involved in classification are:

- The training and test datasets have been prepared as mentioned in Data Preparation section.
- The neural network is configured with the newff() function from the Neural Network Toolkit in Matlab with the given number of layers and nodes.
- The training data along with class labels is used to train the neural network model using the train() command.
- The test data is fed into the network to get test classes predicted by the model.

- These predicted classes are now compared with the known test data classes to calculate different performance metrics like precision, recall and F1 score by constructing a confusion matrix.

Accuracy Metrics for each class using Neural Network:

Class	Precision	Recall	F1-Score
About	0.6900	0.5476	0.6106
And	0.7600	0.8000	0.7794
Can	0.8800	0.8148	0.8461
Cop	0.5600	0.4341	0.4890
Deaf	0.9500	0.5688	0.7116
Decide	0.9500	0.5974	0.7335
Father	0.5800	0.8787	0.6987
Find	0.8800	0.7904	0.8097
Go Out	0.3500	0.3398	0.3448
Hearing	0.6500	0.5963	0.6220

5. Conclusion:

The training data has been obtained from 10 users and it is used to train the different types of machines. Based on test data obtained from the other 27 users, the accuracy metrics - Precision, Recall and F1-Score have been computed. This time, the accuracy metrics are significantly lower than what we observed in the previous assignment. The reason could be that the machines have been trained using data from 10 users and tested using data from 27 different users. Since the test data is from new users, the accuracies are lower than before.