ICS2203/ARI2203 – NLP Methods and Tools
Assignment – Speech Phoneme Analysis and Classification
Deadline: 30th May 2019

Introduction

The aim of this assignment is to build a simple speech classifier for vowel-based phonemes. You will be utilizing some of the tools discussed in class, as well as researching machine learning classifiers that can be utilized for the task and reporting your results. Like in many classification problems, there are three stages that need to be constructed for a functioning classifier: data collection, model training and model testing.

Besides building a general classifier, you will also be analysing the data to notice and comment on particular patterns that you find interesting. Your report must include a description of this analysis, the conclusions you can make from the data, and supporting evidence of your observations and results.

Dataset

https://drive.google.com/file/d/1iXuN2IqzOKqHa9rl6rf40rMEHE2y7g6W/view?usp=sharing

The data provided is a subset of the ABI-1 Corpus (Accents of the British Isles). In this subset is data from 14 different regional accents of the British Isles. For each accent, there are 10 male and 10 female speakers, and each speaker reads an utterance. Every utterance is equivalent across all speakers. The words for this utterance are chosen specifically to elicit vowels and vowel-based phonemes. The word list is as follows:

Heed, hid, head, had, hard, Hudd, hod, heard, hoard, hood, who'd, hade, hide, hoid, heard, howd, heered, hared, hured, heed, hade

Feature Extraction [50%]

You will need to extract a number of features that will allow you to perform your analysis as well as build a rudimentary classifier. The feature extraction process should be saved into a CSV (comma separated values) file. It is easy to use a spreadsheet for recording your measurements and later converting this into a CSV file when you are done.

The CSV file should contain the following fields:

- Speaker label
- Gender (M/F)
- Word
- Vowel Phoneme (in ARPABET notation, a reference can be found here: https://en.wikipedia.org/wiki/ARPABET)
- A class number e.g. 1, 2, 3 (each particular ARPABET symbol should have the same class number)
- Formant 1 – the frequency of the first formant in Hz
- Formant 2 – the frequency of the second formant in Hz
- Formant 3 – the frequency of the third formant in Hz
- Time marker (time location in wave file at which formant values were picked)

In order to extract formant measurements, you can use Praat. When using the 'View and Edit' tool in Praat, it is possible to instruct Praat to show formants for a file. Furthermore, once formant information is plotted on the spectrogram, you can instruct Praat to calculate formant frequency values for selected portions. Since we are interested in vowel sounds alone, make sure your selection contains only the vowel sound, with as little coarticulation as possible. Praat is able to give a formant average over a selection, however you are required to find only a single formant value at a particular location, ideally the mid-point of the selection that you consider to be the vowel, or vowel-based sound. This time marker should also be noted down in your data collection.

You should perform feature extraction for:

- 5 accents of your choice
- Both genders, 5 speakers per gender
- 3 vowel sounds of your choice (3-class problem)

Classifier [50%]

You next task is to build a simple k-Nearest Neighbour classifier that tries to determine what phoneme of speech is being analysed based on formant values. Each of the vowel-based phonemes will be a class. For each class, you will split your data into a training set and a test set. You shall use 75% of the data for each gender for training, and 25% for testing.

For each of the F1/F2/F3 values of the phonemes in the test set, try to classify the phoneme based on the k-nearest neighbours. It is up to you to choose the value of 'k' judiciously. To give an example, suppose k=5. This means you need to find the 5 closest data points in your training set to the F1/F2/F3 data point from your test set. The test data point is assigned a class label based on the majority vote of the 5 nearest neighbours. For instance, if three neighbours are labelled as 'AA' and the other two neighbours are labelled as 'AY', then the test data point is classified as 'AA'. Present your results in confusion matrix form, comparing the true labels of the test points and the classifier labels. If you do not know what a confusion matrix is, now is the perfect time to learn about one of the most common ways of presenting classifier results. It is relatively easy to generate a confusion matrix in Python. You should also list your F1 score for your classifier. **[10%]**

HINT: It is easy to fit a kNN to a dataset in Python. You should have a look at: https://scikit-learn.org/stable/auto_examples/neighbors/plot_classification.html#sphx-glr-auto-examples-neighbors-plot-classification-py

Run the experiment multiple times e.g. 5 times, each time choosing a different training and test set. **HINT:** If you set up your Python code correctly, this selection can be easily done with something like the following: **[10%]**

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(data_matrix, data_label
s, test_size=0.25, random_state=0)
```

Here data_matrix contains 3 columns, one for each of F1, F2, F3; data_labels contains the phoneme label per row (same number of rows as data_matrix); test_size represents the size

of the test set (25%), and random state is a random seed value. The output is a training dataset, a testing dataset, labels for each point in the training dataset and labels for each point in the test dataset.

Some questions you may want to answer, depending on how well you want to investigate the problem (more marks for a deeper investigation):

1) How does performance change with different values of K? **[10%]**
2) What distance metric did you use? Tried any others? **[10%]**
3) How does performance change when classification is done one data for a single gender alone, or when data from both genders are put together? **[5%]**
4) What are the vowel-based phonemes that produce the most confusion (base this off your confusion matrix) **[5%]**

Deliverables:

Please submit a ZIP file on VLE with the following items:

1. A folder with your Python code
2. The CSV file generated from feature extraction
3. A short PDF report of not more than 4 pages, (excluding title page and plagiarism form) which answers the questions outlined in this assignment, with appropriate subheadings.