CS669/DS403 Assignment 2

October 30, 2022

Note.

i. This is an **individual** assignment.

ii. Submit a report with figures and discussions. The report should be named as `PRA2_rollnum.pdf`. Roll number is in lowercase. If you use a different name, your submission will not be considered.

iii. The report should include a Google Colab page with your code and outputs.

iv. The assignment is due on **14 November 2022, 8 PM.**

In this assignment, you will develop a language identification (LID) system using Gaussian mixture models (GMM.) 39-dimensional ($d = 39$) Mel frequency cepstral coefficients (MFCCs) are used as the feature representation. These MFCC features are of varying length (depending on the duration of the speech utterance from which they were derived.) This is achieved by using a voice activity detector algorithm (which itself is based on a GMM.) The code for this is provided.

Dataset

You are going to use the IIT Mandi LID dataset which has audio data from Prasar Bharati (PB) and from YouTube (YT). There are 12 languages to be classified. For each class, use the data in PB_train to train the model. There are two test sets for each class: PB_test and YT_test. Report classification accuracy for each set separately. Each example from the test sets has to be classified.

The data (1.4 GB) and feature extraction code is available here: `https://drive.google.com/drive/folders/1JXx3Tw2mJFUkjrGAqSkvV4e1g-QmFUIC?usp=sharing`

Steps in data processing

1. Run the feature extraction code (`main.py`) to generate csv files containing a set of 39-dim MFCCs. Each `wav` file will result in a CSV having a variable number of rows.

2. Use the data from the CSV files to build your classifier.

3. Repeat the same for test files before performing classification.

4. You can listen to the `wav` files to get an idea of the data.

LID systems to be built

1. **System 1.** This system uses a GMM to model each class conditional density. This is done using the EM algorithm.

2. **System 2.** This is a UBM-GMM system. Pool the data of all classes to form a large GMM, called the universal background model (UBM.) From the UBM, class-specific GMMs are built using MAP adaptation. Only the means are to be adapted, and other parameters (Σ_k, π_k) are used as such from the UBM.

   Given the training data for class c as $X_c = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$, adapt the UBM to get MAP estimates of the mean vectors as :

   $$\hat{\boldsymbol{\mu}}_k = \alpha_k \tilde{\mathbf{x}}_k + (1 - \alpha_k)\boldsymbol{\mu}_k$$

   where

   $$\alpha_k = \frac{N_k}{N_k + r}.$$

   Here, $\tilde{\mathbf{x}}_k$ is the partial estimate of the mean vector using X_c, as in the E-step of the EM algorithm, N_k is the effective number of examples from the kth component using X_c, $\boldsymbol{\mu}_k$ is the mean from the UBM, and r is a *relevance factor*, which can be taken as 0.7.

Details to include in the report

Give results in terms of accuracy and include the confusion matrix where applicable.

1. Which system (1 or 2) performs better and why?
2. How does performance vary with the number of mixtures in the GMM? Give a meaningful plot.
3. Is it better to use a full covariance matrix or a diagonal covariance matrix in the GMM?
4. Compare the performance on PB_test and on YT_test. Why is there a difference?
5. Which languages are confusable and why?