12CS369 Applied Machine Learning SEE: Day 1

Date: 26th May 2015
Time: 10:45 am to 5:45 pm IST

Batch # 3, Day 2, Afternoon & Evening Session: Information Extraction from Audio

The objective of this problem is to extract certain specific information from audio clips. As a part of the current session and the next session, you will have to solve three problems.

Problem 1:

You are expected to build an intelligent system that is capable of identifying the number of distinct participants in an audio clip. Students are encouraged to build the system using Discrete or Continuous HMMs.

We will now discuss one out of the many approaches that can be taken to solve the problem at hand.

The high-level steps involved in developing such a system are as follows:

a) Limit the maximum number of participants that the system is capable of identifying to a number 'n' (Let n=3).
b) The system will consist of 'n' separate HMMs, each HMM capable of identifying a certain number of distinct participants. In simpler words, you will have one HMM to identify audio clips that have 1 distinct participant, another HMM to identify audio clips that have 2 distinct participants and so on.
c) Given an input audio clip, the clip has to be passed via each HMM and the probability of each HMM emitting the input audio clip is calculated. The HMM corresponding to the maximum probability will be equal to the number of distinct participants.

The high-level steps involved in building each individual HMM are as follows:

a) Train the individual HMM with audio clips that the HMM is supposed to identify. In simpler terms, train the HMM that needs to be capable of identifying 1 distinct participant with a number of audio clips that have only 1 distinct person speaking and so on.
b) The training audio clips, which are WAV files need to be converted to a form wherein they can be processed by a HMM. The steps involved in the conversion process are as follows:
    i. Use the dataset generation module provided to convert the WAV files into a form that can be processed by the HMMs.
    ii. This module can convert the WAV files into a series of numbers, formally known as vector quantized codes or into a series of vectors, formally known as MFCC vectors.
    iii. Instructions as to how to use the module are given in the README file that can be found in the module folder.

You can either implement the design described above or come up with your own architecture. Irrespective of which design you implement, the problem can be broken down into smaller tasks as follows:

- Dataset Preparation
- Implementation of the design
- Testing the system

Dataset Preparation:

1. You are given the dataset prepared by students of Batch 1.
2. The folder 'Training Dataset' consists of the audio clips for training the individual HMMs
3. The folder 'Testing Dataset' consists of the audio clips for testing the system.

URLs to download the required modules/datasets:

- Master Audio Clips for Dataset Preparation –
  https://goo.gl/G68oPx

- Dataset Generation –
  https://github.com/arjunsinghvi/SEE-exam/tree/master/Dataset_Generation
  Note: The '*kmeans_train.wav*' file is used to train the k-means classifier and can be a concatenation of all the clips prepared or can be an audio clip of around 20-25 minutes which consists of audio from all the four audio clips provided.

- Initial Model File Creator –
  https://github.com/arjunsinghvi/SEE-exam/tree/master/Initial_Model_Creator
  (Module to create the JSON file required by the Discrete HMM module)

Report to the invigilator the results obtained for the testing clips. (Play the clips and demonstrate)

Deliverables:

- Source code of all your py modules
- Pickled file of the intelligent system (or pickled files of all individual HMMs)
- Files indicating the accuracies obtained (Name of the file – Results.txt)
- File answering the following questions(Name of the file – Design.txt):
  - What is the number of symbols or mixtures per GMM chosen by you? Why?
  - What is the number of hidden states chosen by you? Why?

Problem 2:

You are expected to build an intelligent system that is capable of identifying how much time each participant spoke in an uninterrupted way, given an audio clip. You are encouraged to build the system using Discrete or Continuous HMMs.

The high-level steps involved in developing such a system are as follows:

a) Build a sub-system that is capable of generating a time-line indicating whether a single person is speaking or whether multiple people are speaking at a given point of time. The system would contain 2 individual HMMs, one capable of identifying if a single person is speaking and another HMM that is capable of identifying if multiple people are speaking.
b) Build a sub-system that is capable of generating a time-line indicating which participant is speaking at a given point of time. This system would contain 'n' HMMs, where 'n' is equal to the number of distinct participants. Each individual HMM will be capable of identifying a particular participant.
c) Use the 2 time-lines generated by the sub-systems to calculate how much time each participant spoke in an uninterrupted way. You can ignore the time wherein there are multiple people speaking.

Note: Look at the rate at which the quantization/MFCC generation occurs to get an idea as to how to create the timeline.

Dataset Preparation:

1. You are given the dataset prepared by students of Batch 1.
2. The folder 'Training Dataset' consists of the audio clips for training the individual subsystems.
3. The folder 'Testing Dataset' consists of the audio clips for testing the system.

The steps involved to build the sub-systems are similar to the steps followed by you during the first evaluation of the course. You are expected to follow a similar approach to build the sub-systems.

Report to the invigilator the results obtained for the testing clips.

Deliverables:

- Source code of all your py modules
- Pickled file of the intelligent system (or pickled files of all individual HMMs)
- File containing the participant time distribution for each testing audio clip and accuracy of the system for each of them (Name of the file – Results.txt)
- File answering the following questions with respect to each subsystem (Name of the file – Design.txt):
    - What is the number of symbols or mixtures per GMM chosen by you? Why?
    - What is the number of hidden states chosen by you? Why?

Problem 3:

You are expected to build a system that is capable of detecting certain words. Students are encouraged to use Continuous HMMs to build this system. The system should be capable of identifying the following words/phrases:

- Thank You
- One Second
- Mister

We will now discuss one out of the many approaches that can be taken to solve the problem at hand.

The high-level steps involved in developing such a system are as follows:

d) The system consists of 4 HMMs. 3 HMMs capable of identifying the 3 words and the last HMM identifying words that are not among the 3 words. In simpler words, you will have one HMM to identify the word 'Thank You', another HMM to identify the word 'One Second' and so on.
e) Given an input audio clip, the clip has to be passed via each HMM and the probability of each HMM emitting the input audio clip is calculated. The HMM corresponding to the maximum probability will be equal to the word.

The high-level steps involved in building each individual HMM are as follows:

c) Train the individual HMM with audio clips that the HMM is supposed to identify. In simpler terms, train the HMM that needs to be capable of identifying 'Thank You' with a number of audio clips that have only 'Thank You' being uttered and so on.
d) The training audio clips, which are WAV files need to be converted to a form wherein they can be processed by a HMM. The steps involved in the conversion process are as follows:
    iv. Use the dataset generation module provided to convert the WAV files into a form that can be processed by the HMMs.
    v. This module can convert the WAV files into a series of vectors, formally known as MFCC vectors.
    vi. Instructions as to how to use the module are given in the README file that can be found in the module folder.

You can either implement the design described above or come up with your own architecture.

Dataset Preparation:

1. You are given the dataset prepared by students of Batch 1.
2. The folder 'Training Dataset' consists of the audio clips for training the individual subsystems.
3. The folder 'Testing Dataset' consists of the audio clips for testing the system. The folder will contain two folders:
    - One of them will contain test clips of duration 500ms. The system will assign a word to each file.
    - The other folder will contain audio clips of duration 30 seconds. You are expected to generate a timeline for each test file.

Report to the invigilator the results obtained for the testing clips.

Deliverables:

- Source code of all your py modules
- Pickled file of the intelligent system (or pickled files of all individual HMMs)
- Files indicating the accuracies obtained (Name of the file – Results.txt)
- Files indicating the design parameters of the system and the rationale behind it (Name of the file – Design.txt)

Best wishes from your faculty ☺