CS578: Project on Gaussian Mixture Model (GMM), and Speaker Identification with GMM

March 20th 2020

Delivery: April 30th 2020

Questions: yannis@csd.uoc.gr, kafentz@csd.uoc.gr, or hy578-list@csd.uoc.gr

During this project you will develop an automatic speaker identification system. More specifically the identification system is split into two modules; the features extraction module and the classification or machine learning module which you will develop.

Here are the steps.

1. **Material of the project:**

   For this project you will use two data sets; one for training the classification method and one for testing. **Dataset 1** or training set consists of 20 male and 20 female speakers with 9 sentences each one. **Dataset 2** or testing set will be used for the evaluation of the performance of the system you will develop.

2. **Features Extraction Module:**

   The features that you will extract from the speech signals are the Mel-scale Frequency Cepstral Coefficients (MFCCs). See

   `http://en.wikipedia.org/wiki/Mel-frequency_cepstral_coefficient`

   for a description of these coefficients. You had also a lecture where the computation of these coefficients was provided. You can develop your own Matlab function or you can obtain a free version from an auditory/speech/voice toolbox. For instance, in Matlab central, *http://www.mathworks.com/matlabcentral/index.html*, you will find various implementations of MFCC. For example, you may want to check the following (look for mfcc in the provided table)

   `http://www.mathworks.com/matlabcentral/fileexchange/?term=tag%3A"mfcc"`

   where some interesting applications have also been developed (i.e., devise control using speech).

   Create a Matlab function which performs the feature extraction from a speech signal and saves the features in a .mat file. As a suggestion, you may use $20ms$ frame size and $5ms$

time step.

3. **Training:**

   For each provided speaker, create a Gaussian Mixture Model (GMM) using the corresponding MFCCs features. The estimation of the GMM parameters is performed through the Expectation-Maximization (EM) algorithm. For developing your own EM algorithm you need to follow the steps provided during the corresponding lecture. Also, a tutorial paper was also provided to you. Alternatively, you can use any GMM-EM optimization toolbox which is available from the internet. Again, Matlab central is a rich source of implemented algorithms.

   It is highly recommended to apply GMM-EM optimization algorithm to synthetic examples. Thus, firstly create a Matlab function that trains a GMM using EM for synthetic data. It should plot the clusters of GMM. Then, develop a Matlab function that loads all the MFCC features of a speaker and creates a GMM model for that speaker. Save the parameters of GMM. Do the same for all speakers.

   Write down briefly how EM algorithm operates and what model parameters you use (i.e. how many Gaussians are used in the GMM, is the covariance matrix of the Gaussians diagonal, how EM is initialized).

4. **Testing:**

   Bayesian criterion and especially Maximum a Posteriori (MAP) will be used to discriminate between the speakers. MAP says that an input speech signal belongs to speaker X, if this particular speaker X, has the largest a posteriori probability given the input signal, which is equivalent to the largest likelihood when all a priori probabilities are equal. Thus, given a speech signal from the pool of speakers for which training was performed, try to identify to whom this signal belongs to.

   Report the performance of your speaker identification system. Is it able to discriminate all the speakers or not?

5. **Experiments:**

   Construct two speaker identification modules with different order for the GMM (i.e. more gaussians) and/or different covariance matrix (full or diagonal). Do training and testing

using speech signals with variable duration. Add moderate level of noise. Compare the discrimination ability between the classifiers.

Write down very briefly your observations.

6. **Give us your voice:**

   Record 10 signals using your voice (trying to avoid noisy recording conditions) and with 9 of them construct your GMM. Then, with the remaining 10th signal you will test your speaker identification system (i.e., include yourself in the pool of speakers considered before). You may use 16000 Hz as sampling frequency during your recording and 16 bits resolution. You could collaborate and enrich your data set using the recordings of your colleagues.

   Does the identification system correctly finds the speakers? Report briefly.

Answers may be given in Greek or in English. Return the functions you wrote by yourself plus the original (initial) Matlab file with the requested lines filled in.