Data Analytics for the Smart Society

Lab 3: Speaker Identification

Task 1

For the first part of this task the Matlab script SpeakerIdentificationStudents.m was used as a template and the implementation was added to create a program that evaluates a number of Gaussian Mixture Models in order to identify the speaker of a speech file. This problem is a close set problem and the program selects the speaker of the test file by evaluating the Gaussian Mixture Model corresponding to each speaker that was used in the training stage and selecting the speaker with the largest Maximum Log Likelihood.

For the baseline program the configuration variables are set up with number of speakers in the closed set (16), the sampling frequency of the audio files (16 kHz), the number of MFCC components that you wish to extract from the files (20), the window size (20 ms), the frame period (10 ms) and the number of Gaussians you wish to contain in each of the mixture models (8).

The training files are then loaded using the script load\_train\_data.m. The cesptra components are then extracted for the training test file using the script melfcc.m from the RASTAMAT tool. As we have set our desired number of MFCC components to 20 this function will return 20 cesptra components.

The next step is to fit the GMMs using the extracted features. As there is a level of randomness when fitting these models, in order to track improvements to the model in the later part of the assignment I added a random seed to the script to make the results repeatable. The GMMs are generated with a diagonal covariance and 8 Gaussians using the extracted features from the training file. The GMM models are generated for each of the 16 speakers and stored to be used for the evaluation of test files in the test stage.

In the test stage of the program the test files are loaded and once again the features are extracted for the test files using the same RASTAMAT function as before returning the same number of features. Then for each of the GMMs stored during the test phase a probability density function is calculated using the test features. The maximum log likelihood is calculated for each of the GMM models, which each correspond to one of the speakers. The program then selects the identified speaker as the speaker which has the largest corresponding maximum log likelihood value.

Task 2

For this task the accuracy of the identification system had to be analyzed for both clean and noisy conditions. The program loops through test files which fit each of these categories. The accuracy was calculated by getting the percentage of correctly identified speakers out of the total number of test files. For the files in clean conditions the program returned an accuracy of 98.125% while with files that contained noisy conditions the accuracy had dropped to 58.125%.

Task 3

In this task I tried to improve the accuracy of the model, I did this in a separate script titled SpeakerIdentificationImprovement.m. In order to try to improve the model I first looked at changing the number Gaussians used in each mixture model to see whether more or less Gaussians would return more accurate results. I found that in this case the optimal outcome for the number of Gaussians was to increase the number from 8 to 20 Gaussians. Following this I looked at changing the number of MFCC components, I found that if I increased the number of components to 45 that this would increase the accuracy of the model overall, but with a slight decrease in the accuracy of clean files. In order to increase the accuracy further I added a Regularization term to the GMM model. I found that a value of 0.22 returned the highest accuracy. I tried to play with the number of filters in the filter bank but was unsuccessful in finding a value which increased the accuracy more than the original setting of 40. With these changes I was able to increase the accuracy of the model overall, with the clean files taking a slight decrease to 96.8765% but the noisy files accuracy greatly increasing to 83.75%.