5C4 Assignment 1: Speech Classification

Isla Hoe
13319957

October 28, 2019

Abstract This work focuses on developing and evaluating a speech classification system, with the goal of identifying voiced and unvoiced speech for a given sample. This system makes use of application of Gaussian Mixture Models and resources provided from the TIMIT corpus [1]. The final system is evaluated using a Groun Truth reference system built using the phonetic transcription proved by the TIMIT corpus, and the RAPT model for pitch detection [2].

1 Introduction

Since the the invention of the first audio recorder in 1881 by Alexander Graham Bell, Chichester Bell, and Charles Sumner Tainter, the ability to mimic human behaviour and to use it for automation purposes has been an area of interest for many scientists and researchers [3]. Currently the ways in which speaker recognition systems are implemented is constantly increasing and expanding and has vastly improved from first methods of speech analysis and synthesis initially proposed by Homer Dudley in the 1930's [3]. Almost all mobile computing devices have implemented some form of voice controlled interface [4] and humans are becoming increasingly dependent on these devices. In addition to the increasing use of speech recognition software on mobile devices for convenience and entertainment, vast developments have been made in the area of assistive technology to help support and empower students with learning or physical disabilities to succeed academically [5]. All of this technology works on the same premise that a system is built so that a computer can recognize speech as an alternative input and in many systems that speech input is converted into some form of text which can be processed in various ways [14].

This project focuses on exploring some of the underlying methods used in speech recognition systems and developing a system which takes in a speech sample in the form of a *.wav* file and produces a text file consisting of a breakdown of the voiced and unvoiced regions of speech and their corresponding time-stamp. The implementation of this system uses Gaussian Mixture Models (GMM) to develop a simplistic machine learning model which classifies speech signals as to whether they are voiced or unvoiced. The system is trained using resources from the TIMIT corpus, specifically it uses the female speech samples from the 5th dialect region. TIMIT is built using 8 separate dialects, the 5th is built using speakers from the southern region in the United States [1]. These files are ideal for speech processing research as free from most additional noise.

2 Related Work

This section outlines previous research which was carried out in this area and how it lead to the implementation decisions made in the final model. It also outlines all relevant theory associated with this project.

2.1 Background

2.1.1 *Speech Production*

Human sound is produced when vocal muscles are stimulated and air is forced from the bottom of the vocal tract at the glottis and released via the lips. The positions of various articulators (glottis, velum and tongue) throughout the vocal tract determine which sounds are created. These sounds fall into one of three categories voiced, unvoiced and silence. Voiced speech is generated by quasi-periodic vocal fold vibration and unvoiced speech is produced by vocal tract turbulence [7]. More simply put voiced sounds are those which produce a vibration at the base of the vocal tract (where the collarbone is placed), and unvoiced sounds do not create this vibration. Silence is the background signal with no voiced elements.

These sounds (or units of speech), are known as phonemes. Taken individually phonemes provide no information or meaning however when grouped together they form words. These phonemes can then be classified into different categories such as *stops*, *nasals* or *fricatives* and thus can be further classified into voiced and unvoiced categories. Many languages have their own dictionary (in some cases many dictionaries) of phonemes. Multiple corpus' have been developed with the aim of furthering research into speech analysis. In addition to the TIMIT corpus some alternative English language corpus' include the Cambridge English Corpus and the Spoken English Corpus.

In order to process speech, recordings are made which contain of a continuous series of physical realisations of discrete sounds [10]. Different features such as the short time energy or zero crossing rate may be extracted from this wave form and used to analyse speech and speech patterns.

2.1.2 *Feature Extraction*

Features are characteristics embedded in a continuous signal which contribute to creating different sounds but are not noticeable by the listener. These features contain unique information about the sound produce and can be extracted from a signal in order to identify different aspects of that sound. Thus they are a core component in any speech recognition system. By extracting multiple features and comparing them to known features it is possible to highlight unique characteristics in order to automatically identify different information about the speaker [14].

Extensive research has been conducted into extracting useful features from signals. These

include *Linear Predictive Coding (LPC)*, *Mel Frequency Cepstral Coefficients (MFCC)*, and *Pure FFT* [6], [14].

The use of the Mel-Frequency Cepstral Coefficients (MFCC) as a feature vector has been highlighted by previous research as being one of the most used and successful sets of features when developing voice classification systems [14], [11], [15]. MFCC vectors work well with structured sounds like speech and music signals, but dont respond well when noise is present in a signal [6].

2.2 Related Theory

2.2.1 *Gaussian Mixture Models*

A Gaussian Mixture Model (GMM) is a probability distribution which contains multiple Gaussian distributions. These models can be used to develop simplistic machine learning models by generating large clusters of data which correspond to a specific feature. For a distribution with d dimensions, where d represents the amount of data points, or features in a vector x

$$x = \{c_1, c_2, c_3, ..., c_d\} \tag{1}$$

The Gaussian distribution for d dimensions is defined as follows:

$$N(x, |\mu_k, \Sigma_k) = \frac{1}{2\pi^{\frac{d}{2}}|\Sigma_k|} exp - \frac{1}{2}(x - \mu_k)^T \sum_{k=1}^{k}(x - \mu_k) \tag{2}$$

Where K is the number of Gaussian components, Σ_k is the co-variance matrix for a given component and μ_k is the mean [15]. The probability of a given mixture of K Gaussian's, where a GMM may be represented by λ is given as

$$P(x|\lambda) = \sum_{k=1}^{k} P_k(N(x, |\mu_k, \Sigma_k) \tag{3}$$

The Estimation Maximization (EM) algorithm is a method which is used to find the parameters which best fit the given feature (the feature being tested by the final model). It contains two main steps; the Expectation Step and the Maximization Step. The Expectation Step results in a $n * k$ matrix where each element contains the previous probability of the corresponding observation and component. The Maximization step applies maximum likelihood to those observations and components in order to facilitate convergence.

Each GMM created contains a number of clusters, otherwise known as *mixtures*, these mixtures contain clusters of similar valued vectors and are the key to creating efficient and accurate GMMs.

2.2.2 *MFCC Vectors*

An MFCC feature for an individual frame is created by taking the Discrete Fourier Transform (DFT) waveform, calculating the log amplitude of the spectrum, smoothing and scaling the

spectrum before finally applying the Discrete Cosine Transform (DCT) to the vector [9].

Because this system is developed using the files provided by TIMIT corpus the issue of noise isn't serious thus MFCCs are used as a set of features in this system. During the testing and evaluation stage different aspects of the GMM are varied and the accuracy of the system's results are evaluated in order to construct the most effective set of GMM.

2.2.3 *Short-Time Energy And Short-Time Zero Crossings Rate*

The zero-crossing rate (ZCR) is simply the sum of the times a speech signal crosses the x axis divided by the length of the window [10]. Voiced segments tend to have low ZCR compared to unvoiced segments [11]. Short time energy reflects the magnitude of a speech signal and is calculated by squaring the weighted sum of the speech signal. Generally speaking high levels of energy correspond to voiced speech [8]. In addition to the use of the MFCC features, the ZCR and Short-Time Energy will be calculated as a second 2D set of features, for convenience this set of features will simply be referred to as a ZCR vector. The use of both features and how successful they are in making correct decisions will be evaluated and compared in the testing and results section.

2.2.4 *Reference Systems*

To evaluate the performance of this model, a reference system is used to compare the decisions made by this system to results of a known and trusted model. This assignment makes use of the Robust Algorithm for Pitch Tracking (RAPT), a model which inputs a speech waveform and without using a phonetic transcription, outputs a list of frames and the decision as to whether or not theyre classified as voiced or unvoiced depending on the features extracted from that frame. [2].

In addition to the RAPT model this project makes use of the Groun Truth method, where a Groun Truth of voiced and unvoiced decisions is generated for a given signal depending on its phonetic transcription provided by TIMIT and then compared to the results of the system.

2.2.5 *Voiced and Unvoiced Phonemes*

In addition to converting the *.wav file* into a waveform suitable for processing, using the *readsph.m* function in the VOICEBOX toolkit, the phonetic transcription (phn) and word transcription (wrd) associated with the the given speech sample were stored alongside the corresponding time-stamps in a MATLAB *cell*. Each of the full set of phonemes was manually classified as voiced or unvoiced by physical inspection, where two fingers were placed on the bottom of the vocal tract (a centimeter above the collarbone)and each phoneme was sounded out; if vibrations were felt the phoneme was classified as voiced, if no vibration was felt it was classified as unvoiced. The following phonemes were classified as voiced; *u, ch, s, sh, th, t, k, c, p, h, epi, and pau.* For ease of classification and due to the minimal number of occurrences, any silences *(epi, pau, h)* which occurred during the classification stage were marked as unvoiced.

3 Methodology

3.1 Implementation

The full system was developed and built using MATLAB, the open-source toolkit VOICEBOX [12] and various other inbuilt matlab functions. VOICEBOX was used to read in the speech files in the form of .wav files and corpsponding phn and word files. It was also used to extract the MFCC vectors using *melcepst.m* and the *fxrapt.m* function was used to implement the RAPT reference system.

3.2 Algorithm

This algorithm encompasses three main components; training, testing and accuracy checking.

Training

Multiple speech samples in the form of *.wav* files are input to the training function. In order to add feature vectors to the GMM, the full waveform was processed by individually processing frames one by one. Each frame consisted of a 30ms sample of the full waveform with an overlap of the previous frame of 10ms, and was processed using the following step. Each of these steps can be seen in Figure 1.

1. *Feature Extraction*: A set of features is extracted using either the *melcepst.m* function or the *dsp.ZeroCrossingDetector* added to an $2 * n$ array, where n is the number of frames. The time-stamp of the frame is also added to the array at this point.
2. *Classification*: The phonetic transcription is found for the frame using the time stamp and comparing the time-stamp with the phn file. The phoneme given for that time-stamp is then classified as either voiced or unvoiced using the list of manually generated voiced phonemes. Features which are classified as voiced are added to an array of voiced features and the rest are added to an array of unvoiced features.
3. *GMM Creation*: The arrays from the *classification step* were then used as the data points for GMM creation. The *fitgmdist* is used to generate two GMM's, one which contains voiced features and one which contains unvoiced features. This function allows for the specification of mixture sizes and the amount of iterations of the EM algorithm. The amount of mixtures was set to 2 and the amount of iterations was set to 1000.

Testing

Testing was carried out using a similar process to the training method. Each waveform was processed frame by frame, features were extracted using the *melcepst.m* function (or the ZCR function). The difference between the training and testing set is that once features from a frame are extracted, the log likelihood of those features being represented by each of the two GMM's is calculated. This is done by using the pdf function and subtracting the log of the probability of the feature vector belonging to

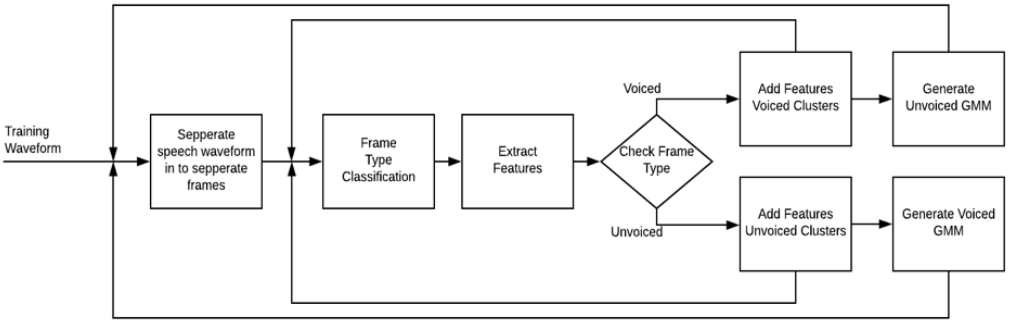

Figure 1: Block Diagram of the GMM Training Process

the voiced GMM from the log of the probability of the feature vector belonging to the unvoiced GMM. If the result of this subtraction is positive the frame classified as voiced otherwise it is classified as unvoiced. The result of each classification was stored in an array which was printed to a *.txt* file along with its time-stamp represented in samples.

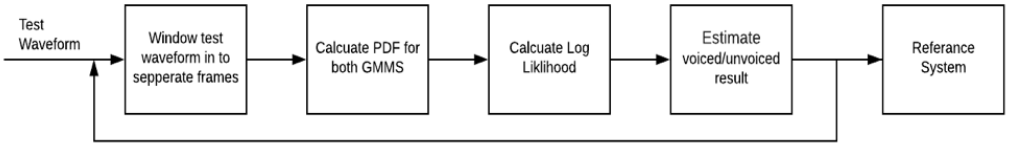

Figure 2: Block diagram of the prediction classification stage

Accuracy Checking

The accuracy of the system is evaluated by using the RAPT pitch detector and a Groun Truth model built using the phonetic transcription and compiling two lists of results consisting of the time-stamp of the frame and its voiced or unvoiced classification.

The *XOR* function in MATLAB is used to decipher which voiced/unvoiced decisions match the RAPT and Groun Truth results, by applying the *XOR* function to the prediction result and the corresponding Groun Truth and RAPT results for a given time frame. A zero result indicates that this systems decision matches the result from the reference system.

It should be noted that the Ground Truth method was not used for accuracy checking during the challenge session as its dependent on the phonetic transcription of the waveform which is why the RAPT model has also been included for evaluation.

4 Testing and Results

Each GMM was trained using the 104 different speech samples. A further 20 of these samples were reserved for testing the system. These testing files were not used as part of the training

set, as this would create a situation where a test signal would result in having an extremely high probably of belonging to the correct distribution because it would match a set of features already contained in that distribution. The average accuracy used in each set of results is calculated by taking the accuracy of multiple sections of the signal and finding the average accuracy of the full signal.

4.1 GMM Manipulation

The amount of mixtures used in GMM generation was set to two for the final session challenge and for the remainder of the evaluation and testing period. Eleven tests in total were carried out which recorded the average accuracy of 20 test signals. The average accuracy is calcuated by the sum of the accuracy for each test sample and divided by the total amount of test samples (20). For each test the amount of mixtures was set from 1 up to 10 and a final test was conducted where the amount of mixtures was set to 20. Table 1 displays the results of the tests.

Overall the number of mixtures had a very small effect on the overall accuracy of the system, however two mixtures provided the most accurate results for the Ground Truth reference system with 88.12%Àlthough there werehigher accuracy values for the RAPT model with 3 and 10 mixtures, two mixtures were chosen because the Ground Truth system is expected to be more accurate than the RAPT system.

	Referance System		
Mixtures	*GT*	*RAPT*	*Time*
1	87.63	85.1	74.68
2	88.11	86.11	85.64
3	87.9	86.22	73.3
4	87.92	85.97	97.78
5	87.6	86.19	86.63
6	87.55	85.73	108.7
7	87.33	86.08	92.91
8	87.33	85.84	85.16
9	87.21	85.9	102.12
10	87.79	86.25	107.6
20	87.82	86.82	94.2

Table 1: Average Accuracy for GMMs with 1 - 10 Mixtures

Each iteration of the EM algorithm carries out either an Expectation Step or a Maximization Step, and these steps are repeated until convergence is reached. To investigate how the number of iterations affects the accuracy of the system, multiple tests were carried out using both feature sets and three different iteration values; 100, 1000 and 5000. Table 3 displays the average accuracy for 20 speech signals, using two different sets of features; the zero crossing rate and short time energy (ZCR) and the MFCC vectors. This set of results also

displays the time it took to compute the full procedure of training, testing and accuracy checking.

	RAPT		**Groun Truth**		**Time**	
Itterations	*ZCR*	*MFCC*	*ZCR*	*MFCC*	*ZCR*	*MFCC*
100	79.42%	85.85%	83.90%	88.04%	62.67s	85.78s
1000	79.43%	85.87%	83.91%	88.06%	64.41s	95.26s
5000	79.43%	88.66%	83.91%	86.59%	66.04s	108.81s

Table 2: Average Accuracy Values for four sets of feature vectors

Initially it was expected that as the number of iterations increases the accuracy would also increase, however it is clear that this is not the case. There was a noticeable increase in time, with intervals of 2 and 4 seconds for the ZCR features and 10 seconds for the MFCC vectors, likely due to the fact that there are 10 more features in each MFCC vector compared to the ZCR vector. It is noted by Douglas Reynolds in a 1995 paper that 10 iterations is often sufficient to reach convergence [13], however this proved false in a number of cases where the number of iterations was set to 100 and a warning appeared that the GMMs had not yet reached convergence.

Because the number of iterations have clearly had a limited effect on the accuracy of the system all other testing was done using 1000 iterations, this was used instead of 100 in order to avoid not reaching convergence. The likely reason that the accuracy didn't improve after 100 iterations is that in most cases 100 iterations were sufficient to reach convergence and therefore after a certain number it is likely the iterations were automatically stopped.

4.2 Frame Overlap

Each waveform processed frame by frame with a standard overlap of 10ms (meaning that 10ms of the previous frame is always part of the current frame, excluding the initial frame) this overlap means that no set of features for an individual frame for training or testing are classified in isolation. However since the accuracy for both reference systems is similar, the frame overlap was reduced for both the training and testing samples to investigate the effect it has on the accuracy of the system.

A value of of 0.25, 0.5 and 0.75 correspond to a reduction of the overlap by 25% , 50% and 75% of each frame respectively. A value of 0 indicates that there was no overlap and the following frame started at the end of the current frame, and a value of 1 indicates that the overlap hasn't been changed from the original settings. The training column represents the change in overlap when creating the GMMs for training frames and the testing column represents change in frame size while processing the test samples. The results of these tests can be seen in Table 2.

Comparing these results with the results in Table 3 there is a clearly a benefit to using more than two features because this allows for more variance within a given frame, which

Frame overlap %		ZCR		MFCC	
Training	**Testing**	*GT*	*RAPT*	*GT*	*RAPT*
0.75	0.5	54.335	58.675	56.779	61.503
0.75	0.25	51.407	57.998	52.548	57.683
0.75	0.75	54.195	57.651	56.477	59.793
0.5	1	54.308	57.764	88.005	85.963
0.25	1	54.252	57.708	88.223	86.247
0	1	54.252	57.708	88.206	86.247

Table 3: Frame Overlap % and Average Accuracy Values for four sets of feature vectors

therefore increases the probability the the correct decision will be made. It became evident when carrying out these tests that when the frame overlap is reduced for the training frames it only has a marginal effect on the accuracy of the system using MFCC vectors but has a drastic effect on the accuracy using the ZCR vectors. It is also clear from these results that by reducing the testing frame overlap the drop in accuracy is reflected in the results of both sets of features. This is due to the fact that only a single test frame is used for decision judgments where as there are multiple frames used for training so the frame overlap size is not as significant.

4.3 Test Set Accuracy Evaluation

The graph in Figure 3 displays the plot of the average accuracy value for both reference systems and both sets of vectors for each test signal. The X axis displays the average value for the accuracy of the system for each test signal, and the Y axis shows the values for accuracy. The plot highlights four sets of results, each taken from the results for the two reference systems and the two sets of vectors, based on a frame overlap percentage of 10ms and 1000 iterations of the EM algorithm. The blue lines represent the results of the MFCC vectors and the orange and red lines represent the results of the other set of vectors. From observation of this graph and previous results it is evident that the use of MFCC vectors provide consistently more accurate results.

Across the entire spread of results, the highest accuracy achieved was 92.9% using the Ground Truth reference system and the 93.8% using the RAPT reference system, however this test file, SX280.WAV from the WF0 folder, consistently results in in the highest accuracy scores. This is likely due to the fact that the test file contains very similar features to some of the training files. This may be avoided by using the process of cross validation where the data used for testing and training is rotated, for example 10% of the each sample is reserved for testing and 90% is used for training. It is expected that using cross validation would improve the accuracy of the entire system however it was not implemented due to a lack of technical skills.

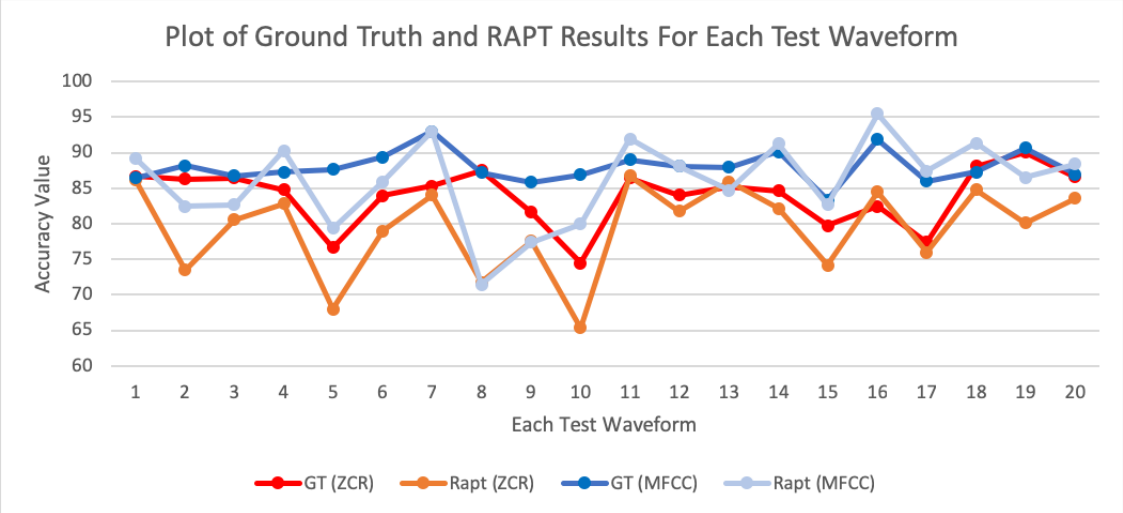

Figure 3: Plot of the Average Accuracy For Each Test Sample.

5 Conclusions

The implementation of this system was reasonably successful. Using the optimum number of of iterations (1000), either of the two sets of feature vectors and some level of frame overlap for the testing decisions, the average accuracy scores fell between 83% and 93%MFCC vectors are the ideal choice for future work on this system as it allows for a smaller dependency on the content of the individual frames, which means that computational complexity may be reduced by reducing the frame overlap. The number of iterations used for GMM convergence doesn't have much of an effect on the overall performance of the system, the only stipulation is that enough iterations need to take place to allow the GMM to fully converge.

References

[1] TIMIT Acoustic-Phonetic Continuous Speech Corpus -Linguistic Data Consortium. [Online]. [Accessed: 02-Dec-2018].

[2] D. Talkin, A robust algorithm for pitch tracking (RAPT), in Speech Coding and Synthesis (W. B. Klein and K. K. Palival, eds.), Elsevier, 1995.

[3] B. H. Juang and L. R. Rabiner, Automatic Speech Recognition A Brief History of the Technology Development, p. 24.

[4] A. E. Moorthy and K.-P. L. Vu, Privacy Concerns for Use of Voice Activated Personal Assistant in the Public Space, International Journal of HumanComputer Interaction, vol. 31, no. 4, pp. 307335, Apr. 2015.

[5] K. E. Forgrave, Assistive Technology: Empowering Students with Learning Disabilities, The Clearing House: A Journal of Educational Strategies, Issues and Ideas, vol. 75, no. 3, pp. 122126, Jan. 2002.

[6] U. Shrawankar and V. Thakare, Feature Extraction for a Speech Recognition System in Noisy Environment: A Study, in 2010 Second International Conference on Computer Engineering and Applications, 2010, vol. 1, pp. 358361.

[7] J. Campbell and T. Tremain, Voiced/Unvoiced classification of speech with applications to the U.S. government LPC-10E algorithm, in ICASSP 86. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1986, vol. 11, pp. 473476.

[8] M. Jalil, F. A. Butt, and A. Malik, Short-time energy, magnitude, zero crossing rate and autocorrelation measurement for discriminating voiced and unvoiced segments of speech signals, in 2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), 2013, pp. 208212.

[9] B. Logan, Mel Frequency Cepstral Coefficients for Music Modeling, in In International Symposium on Music Information Retrieval, 2000.

[10] L. R. Rabiner and R. W. Schafer, Theory and applications of digital speech processing, 1st ed. Upper Saddle River: Pearson, 2011.

[11] P. Harding and B. Milner, On the use of Machine Learning Methods for Speech and Voicing Classication, p. 4, 2012.

[12] M. Brooks, Voicebox: Speech processing toolbox for matlab - http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html, 2018.

[13] D. A. Reynolds and R. C. Rose, Robust text-independent speaker identification using Gaussian mixture speaker models, IEEE Transactions on Speech and Audio Processing, vol. 3, no. 1, pp. 7283, Jan. 1995.

[14] S. Narang and D. Gupta, Speech Feature Extraction Techniques: A Review, p. 8, 2015.

[15] T. Kinnunen and H. Li, An overview of text-independent speaker recognition: From features to supervectors, Speech Communication, vol. 52, no. 1, pp. 1240, Jan. 2010.