Adaptive Speaker Recognition

by

Niranjani Prasad (CHR)

Fourth-year undergraduate project in
Group F, 2012/2013

I hereby declare that, except where specifically indicated, the work submitted herein is my own original work.

Signed: _____ Date: _____

Technical Abstract

Adaptive Speaker Recognition

Niranjani Prasad (CHR)

Recent years have seen a significant rise in the interest in speech as a way of interacting with technology, from smart phones to automobiles. The capability to automatically recognise a person from his or her voice can be vital in simplifying and personalizing this interaction. This project involves the design and development of an adaptive speaker recognition algorithm, tailored for implementation in a domestic service robot. It is targeted at the Chinese market, and provides an extension to conventional smart home solutions.

The problem can be described as that of text independent open set identification: the proposed algorithm distinguishes between a small set of known speakers, i.e. the members of the household, and identifies a new speaker as unknown. It is designed to operate in real time and adapt to changes in the voices of the registered speakers. Lastly, measures are taken to make the system robust to both background disturbances and internal robot noise.

Speaker recognition systems comprise two phases, namely speaker enrolment and identification; the latter is the main operational phase. Both stages are defined by a few key modules: front end processing, feature extraction, and the generation of a statistical representation from these features. Here, MFCC feature extraction and Gaussian mixture modelling provide the framework for an initial maximum-likelihood based identification system, designed in Matlab. This system is used as the basis for further development. The fundamental challenge of speaker recognition lies in compensating for session variability, i.e. the differences between recordings of the same speaker during enrolment and identification. The primary source of variability in this application is drift in the speaker's voice itself. This is addressed here using long-term MAP adaptation of the speaker models.

A speech corpus, comprising data recorded directly by the robot from 60 different speakers, was requested for initial testing and optimization of front-end processes. This involved noise attenuation and speech enhancement of the raw data; a range of

approaches were investigated, from filtering techniques to methods based on spectral restoration. These techniques were compared by the gains in the performance of the identification system. The final scheme uses a combination of lowpass filtering and MMSE-LSA estimation, and yields an improvement in accuracy of 13% over unprocessed data. Following this, an energy-based voice activity detector was implemented, to prevent the real-time identification system from running continually even when no speaker is present. This was designed to fail safe, i.e. to ensure no speech is discarded, but gives a false detection rate of 27%.

In order to test long-term speaker adaptation, 60 utterances were collected from each of 5 speakers over a period of 5 weeks. Model adaptation was introduced in two aspects of the problem, first in building a UBM and adapting this to each speaker, rather than training five independent models, then in the incremental adaptation of each speaker model over time. These inclusions yielded an increase in accuracy of 15% and 7% respectively. The choice of parameters for MFCC feature extraction and GMM modelling was also explored. The proposed scheme incorporates 24 dimensional MFCCs with no dynamic information, and GMMs comprising 3 mixture components and diagonal covariances. The final closed set identification system achieved an accuracy of 93.3%.

To extend this to an open set problem, a threshold minimum is set on the value of the maximum likelihood achieved by the registered speaker models. If this threshold is not met, the utterance is classified as originating from outside the speaker set. This requires a trade-off between false rejections of speakers within the set and false acceptances. Maintaining a false rejection rate of below 5% yields a false acceptance rate of 55%. This is improved by introducing the UBM as a competing model in the decision rule. An 18.7% decrease in false acceptances is achieved. The resulting cost on false rejections, and in turn on identification accuracy, is just 1.3%.

Thus the proposed strategy achieves an identification rate of 90% and a rejection accuracy of 64%. 73% of blank samples are correctly discarded. Overall, the algorithm provides a reasonable foundation for robust adaptive speaker recognition. Implementation in the robot is currently underway.

Contents

List of Figures

List of Tables

Nomenclature

CMS Cepstral Mean Subtraction

DCT Discrete Cosine Transform

EER Equal Error Rate

EM Expectation Maximization

GMM Gaussian Mixture Model

IMCRA Improved Minima Controlled Recursive Averaging

LPC Linear Predictive Coding

LTSD Long-Term Spectral Divergence

MAP Maximum A Posteriori

MCE Minimum Classification Error

MFCC Mel Frequency Cepstral Coefficients

MLLR Maximum Likelihood Linear Regression

MMSE-LSA Minimum Mean-Squared-Error Log Spectral Amplitude

OSTI-SI Open-Set Text-Independent Speaker Identification

ROC Receiver Operating Characteristic

SNR Signal-Noise Ratio

SVM Support Vector Machine

UBM Universal Background Model

VAD Voice Activity Detection

WSS Wide Sense Stationary

1 Introduction

Speaker recognition is the use of voice as a biometric, in other words, determining a person's identity by extracting information from speech. Voices can be as distinctive as faces or fingerprints, and can provide an easy and intuitive means of identification. With the merging of telephony and computer networks, the increasing interest and development in speech recognition as a way of interacting with technology, and the ever growing volumes of data in the form of spoken documents, the capability to automatically recognise a person from his or her voice can be of great advantage.[2]

Speaker recognition encompasses two main classes of problems, namely *verification* and *identification*. In speaker verification, the task is to decide from a given voice sample, or utterance, whether a person is who he or she claims to be. This requires us to distinguish the claimed speaker's voice from one belonging to a large, open set of alternatives, or 'impostors'. In speaker identification on the other hand, we have a closed set of registered speakers. Given an unknown utterance (and no prior identity claim), we want to be able to determine which of these known speakers it belongs to. Speaker recognition tasks can also be classified according to the constraints placed on speech used to train and test the system: in a text-dependent system, the speech in the training and test phases is constrained to be the same word or phrase - this requires acoustic modelling techniques from speech recognition to be incorporated, but simplifies the speaker recognition problem by eliminating one source of variation. Text-independent systems impose no such constraints on the utterance to be identified.

1.1 Current Applications

Speaker recognition has a wide range of commercial applications. Verification problems are dominant in most existing systems, for example in security as a supplement to other biometrics, or in voice authentication for telephone based services. These are usually configured to be text-dependent, requiring the user to speak some personalized verification phrase which is first processed by a speech recognition system, before the speaker's voice is verified. In contrast, surveillance applications are likely to employ text independent recognition systems, as in this case the speaker is not

aware of being monitored, and there is no control over the words spoken.[2]

The use of simple closed-set identification is limited to scenarios in which it is known that only enrolled speakers will be encountered, such as speaker labelling or diarization of recorded meetings. It enables more accurate transcription of such data, as speech from multiple simultaneous speakers can be separated. In addition, closed-set identification algorithms can be used to match a new speaker to the most similar stored voice, a principle that is sometimes applied to speaker-adaptive speech recognition in systems with a limited, pre-existing set of registered speaker models. In forensics, often speaker identification is first used to generate a shortlist of best matches before performing a series of verification processes for a conclusive fit.[3]

1.2 Project Aims and Motivation

The aim of this project is to design and develop an adaptive speaker recognition algorithm for implementation in a domestic service robot. More specifically, the algorithm is to be tailored to *Nao*, an autonomous programmable humanoid robot developed by Aldebaran Robotics[1]. The robot is targeted at the Chinese market and is envisioned as a helping hand for families and the elderly, an extension to conventional smart home solutions. The primary requirements of the algorithm are as follows:

- ▷ Able to distinguish between a small set (4-5) of known speakers and identify a new speaker as unknown, in real time;
- ▷ Designed to adapt to change in the voices of known speakers over time;
- ▷ Independent of the words/phrases used in the utterance to be identified;
- ▷ Able to cope with background noise typical of a household environment as well as discard silences;
- ▷ Trained for Chinese speakers, of all age groups.

The problem can therefore be described as that of text independent open set identification (OSTI-SI), a fusion of typical identification and verification tasks which performs like closed-set identification for known speakers, but must also be able to

[1] Aldebaran Robotics: http://www.aldebaran-robotics.com

classify speakers unknown to the system into a 'none of the above' category. This is a useful learning problem, as knowledge of the speaker's identity enables the storage and retrieval of speaker-specific settings in the robot for better usability, and gives scope for long-term adaptation to the preferences of a particular user.

1.3 Structure of the Report

This report will begin with a theoretical overview of the key concepts in speaker recognition, describing the dominant approaches in existing literature and how these can be applied to the task at hand. It goes on to outline the preliminary work carried out in order to obtain a measure of the relevance and effectiveness of these key methods, and gives details of the speech corpus obtained for the training and testing of the algorithm. The report continues by detailing the efforts made to optimise the various components of the speaker recognition system, from front end processing of speech signals and voice activity detection, to honing the parameters of the speaker model and implementing efficient online adaptation. Finally, it evaluates the overall performance of the proposed open-set speaker identification system, and suggests ways in which this could be improved.

2 Theory

2.1 Overview of OSTI-SI

An automatic speaker recognition system operates in two distinct phases, enrolment and identification. During enrolment, each accepted speaker is heard for the first time, and registered by the system. This involves:

- ▷ *Preprocessing* these utterances to attenuate noise and enhance speech;
- ▷ *Extracting features* that best emphasize speaker-specific characteristics;
- ▷ *Modelling* these features with an appropriate statistical representation.

The identification phase is where given an utterance, we attempt to determine the identity of the speaker. In the case of an OSTI-SI system, the identification procedure is in itself two-fold. The key steps are illustrated in Figure 1. As in enrolment, we start by processing and extracting features from the sample. We then identify the model within the set of known speakers that best matches the observed features, typically in terms of the likelihood of the utterance given each model. Finally, we need a way of determining whether the utterance has in fact been produced by the speaker associated with the best model, or by someone outside the registered set. This is analogous to the problem of speaker verification, and can be thought of as the case where each 'impostor' targets the speaker model in the registered set for which it can achieve the maximum score; rejecting an unknown speaker therefore becomes a much more difficult problem, increasing in complexity with the size of

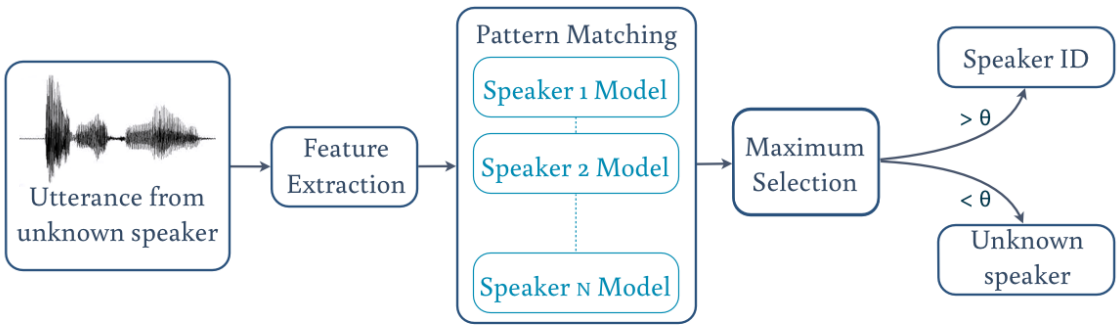


Figure 1: Speaker Recognition System: Identification Phase

the population of known speakers. Though we will be dealing with just a handful of speakers in this application, it is expected that this latter part of the identification process will be most limited in its performance.

For N enrolled speakers with model descriptions $\lambda_1, \lambda_2, ..., \lambda_N$, if X denotes the feature vectors extracted from the test utterance, the decision rule for open-set identification can be stated as follows:[7]

$$\max_{1 \leq n \leq N} \{p(X|\lambda_n)\} > \theta \Rightarrow X \in \begin{cases} \lambda_i, i = \arg\max_n \{p(X|\lambda_n)\} \\ \text{Speaker not known} \end{cases} \quad (1)$$

where θ is a pre-determined threshold on the maximum likelihood score. If the maximum lies below this threshold, the utterance is declared as originating from an unknown speaker. From this rule it follows that, in classifying a given utterance, the system can generate one of three classes of errors:

- ▷ *False Identification*: Utterance X from speaker λ_m yielding the maximum likelihood for λ_n, where $n \neq m$.
- ▷ *False Acceptance*: Assigning X to a model $\lambda_n, 1 \leq n \leq N$ in system when it does not originate from any known speaker.
- ▷ *False Rejection*: Declaring X belonging to λ_m (for which it also yields the maximum likelihood) as originating from an unknown speaker.

These errors occur either due to limitations in the fit of the speaker models or, more importantly, *session variability*, i.e. differences in two recordings of the same speaker. This remains a considerable challenge in speaker recognition. It can arise from channel mismatch/variations in noise present in the signal, as well as simply because of time lapse: the speaker's voice will itself drift over a period of time, as a result of stress, illness or physiological changes. Channel variability is a lesser concern within this application as all recordings to be identified are uniform, obtained via the robot. Strategies to cope with time lapse effects include *data augmentation*[1], where each time a speaker is heard and correctly identified, the new data is used to regenerate the entire model, or *model adaptation*, typically using MAP. This approach gives greater control over how much the model is corrected each time and does not require the storage of hours of data, hence will be the focus in this project.

2.2 Feature Extraction

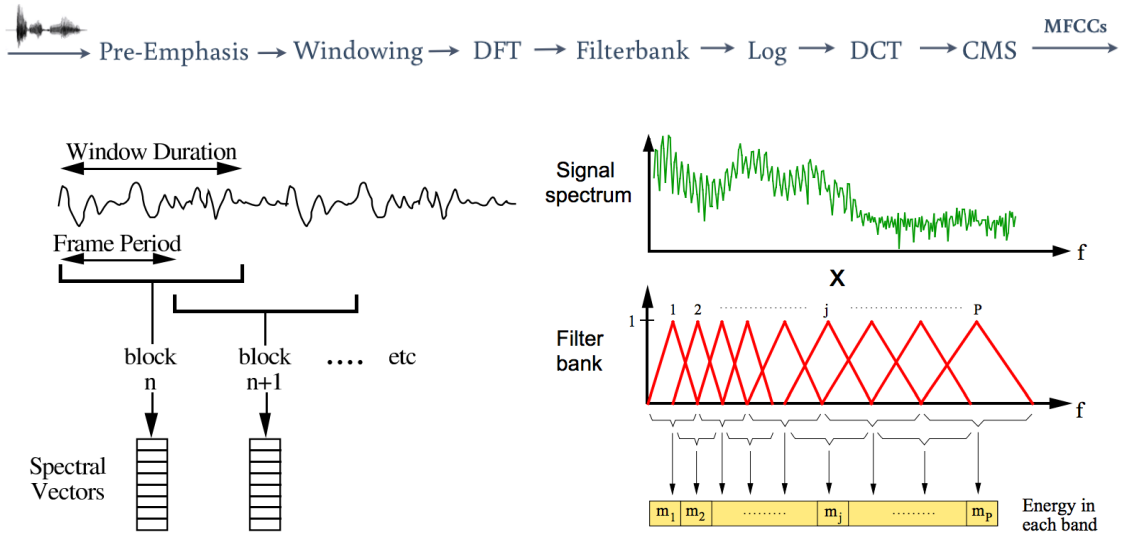
Speech data encodes a variety of information, only a fraction of which convey speaker-specific attributes. Feature extraction is crucial in reducing the data rate and removing redundant information while retaining these attributes. For text-independent speaker identification, we choose features that give large inter-speaker variability and small within-speaker variability (i.e. are invariant to colds/stress, but at the same time are difficult to impersonate). They must be easy to extract and robust against background noise or distortions in the recording channel. Features for a given utterance should be independent of each other, in order to minimise redundancy. Ideally, it should also be possible to interpret these features as an intuitive representation of a voice. Suitable features range from low level, short term spectral properties of the speech sample characterising physical traits of the vocal tract, to prosodic features (which incorporate intonation and rhythm) or high level characteristics such as frequent use of particular words and phrases, reflecting dialect and style, though these attributes tend to be easier to mimic.[9]

2.2.1 MFCCs

Spectral features, in particular Mel-frequency cepstral coefficients, have been shown to largely satisfy the above requirements, and dominate most speech-related work in existing literature. The steps involved in the extraction of MFCCs are as follows: the input speech signal first undergoes pre-emphasis of high frequency information as, in the raw signal, lower formants contain much more energy and therefore tend to be modelled more accurately than higher frequencies. The signal is then analysed using 25ms windows progressing at a 10ms frame rate (resulting in overlapping windows; this *block processing* technique is illustrated in Figure 2a). Once the speech signal has been windowed, we take its DFT to obtain a power spectrum. Short term fluctuations in the spectrum are discarded, and just the spectral envelope retained, by multiplying the power spectrum by a Mel-scale filter bank (Figure 2b). This comprises a series of triangular bandpass filters with centre frequencies spaced according to the Mel scale, $f_{Mel} = 2595 log_{10}(1 + f/700)$, which closely approximates the spectral resolution of the human ear.[13] Finally, we take the logarithm of this smoothed spectrum and apply the DCT. This can be thought of as the compression

step, orthogonalizing and reducing the number of parameters required to represent a frame of speech, which in turn reduces memory and computational costs.

MFCC feature extraction is popular in both speech and speaker recognition applications as it provides a compact representation of speech samples. The cepstral features are decorrelated as a result of the DCT, which not only minimises the amount of redundancy but also allows us to model these features using Gaussians/GMMs with diagonal covariance matrices (which cannot explicitly model dependencies between elements of the feature vector). In addition, these statistical models usually



(a) Block Processing *(b) Mel Scale Frequency Bank*

Figure 2: Extracting Mel Frequency Cepstral Coefficients

assume stationary speech, with no dependencies between frames. MFCCs provide scope to relax this assumption and model co-articulation effects by incorporating dynamic information using Δy_t and $\Delta^2 y_t$ cepstra values, which are essentially the first and second order frame derivatives of the feature vectors (y_t is the feature vector for frame t) and can be concatenated to produce the final observation vector. Lastly, cepstral mean subtraction (CMS)[9] eliminates static channel noise in the utterance to some degree; normalized MFCCs have been found to be noticeably less sensitive to additive noise than other spectral features.

Alternatives

Variants of MFCC such as spectral analysis based on linear predictive coding (LPC) follow similar procedures, and performance is comparable but does not surpass that from MFCCs. In addition, recent years have seen a growing interest in exploiting high-level features for text-independent speaker recognition systems, driven by developments in phone and language modelling, though their use is still limited by high computational costs. Hence, MFCCs will be chosen to provide the basis of speaker discrimination in our system.

2.3 Modelling the Speaker

Following the extraction of speaker dependent features from raw speech, we must decide how best to distinguish between speakers. This can be done either by training generative models, which specify full probabilistic representations of each class/speaker and identify utterances based on the model that yields the maximum likelihood, or discriminative classifiers, which can directly map observations, or utterances, to a speaker. The choice of model depends on the nature of the speaker recognition problem.

2.3.1 Generative Models

In text-independent speaker recognition, where no constraints have been posed on what is said by the speaker, the most popular form for the statistical representation derived from feature vectors is the Gaussian Mixture Model. A GMM, denoted here by λ, is composed of a finite weighted mixture of multivariate Gaussians, and is characterised by the following probability density function:

$$p(x|\lambda) = \sum_{m}^{M} P_m \mathcal{N}(x|\mu_m, \Sigma_m) \tag{2}$$

Here, M is the total number of mixture components, P_m is the mixing weight (the prior probability of the m^{th} component), and $\mathcal{N}(x|\mu_m, \Sigma_m)$ denotes a multivariate Gaussian PDF with mean vector μ_m and covariance matrix Σ_m. The covariance

matrices are usually constrained to be diagonal, as a full covariance GMM requires much more data and is computationally expensive to train. Furthermore, though a large number of mixture components would be needed to model highly asymmetric distributions using just diagonal covariance matrices, MFCC features are almost perfectly orthogonal and do not need explicit modelling of correlations, so can be reasonably well represented by small M. The parameters of the model for each speaker are estimated from the available training data using Expectation Maximization. This is an iterative procedure which involves first estimating the class posterior probabilities from current parameter estimates, then refining parameters to incrementally increase the likelihood of the parameters given the data. This method is used where direct maximisation of the likelihood is not possible because the latent variables, in this case the components of the GMM associated with each feature vector, are not known. Models for each speaker can either be trained independently (producing a series of *decoupled* GMMs), or by adapting from a single Universal Background Model (UBM), built on representative samples of speech from a large pool of speakers.[6]

The Hidden Markov Model is an extension of GMMs which models each speaker with a series of states, each described by a GMM. Though widely used in speech recognition, existing work shows no performance gains over GMMs for text independent speaker identification; it will therefore not be considered in this work.[9]

2.3.2 Discriminative Classifiers

While GMMs deal with modelling each individual speaker, and minimising intra-speaker variance, discriminative classifiers such as support vector machines (SVMs) model the boundaries between speakers. An SVM is a binary linear classifier which looks to find a separating linear hyperplane that maximises the margin between the nearest samples of two classes. Applied to speaker verification, one class is formed by the target speaker training vectors, and the other consists of the feature vectors from the background population. The decision hyperplane can be written in the form $f(\boldsymbol{x}) = \boldsymbol{w}'\boldsymbol{x} + b$, where all data points $\boldsymbol{x}_i$ satisfy the constraint $(\boldsymbol{w}'\boldsymbol{x}_i + b)y_i \geq 1$ (assuming points are linearly separable). Maximizing the margin between points in different classes is equivalent to minimizing the objective function $E = \frac{1}{2}||\boldsymbol{w}||$; this

can be combined with the earlier constraint to formulate the following Lagrangian equation:[4]

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}||\boldsymbol{w}|| - \sum_{i}^{n} \alpha_i (y_i(\boldsymbol{w}'\boldsymbol{x} + b) - 1) \tag{3}$$

Solving for the minimum $\mathcal{L}$ gives us the following equation for the optimum decision boundary:

$$f(\boldsymbol{x}) = \boldsymbol{w}'(\boldsymbol{x}) + b = \sum_{i}^{n} \alpha_i y_i (\boldsymbol{x}'\boldsymbol{x}_i) + b \tag{4}$$

Here, x_i are the support vectors (the points lying closes to the linear boundary), α_i the corresponding weights and b the bias term. Points that are not linearly separable in the original (MFCC) feature space can be transformed into a higher dimensional space using the 'kernel trick': a kernel function can be expressed as $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \phi(\boldsymbol{x}_1)'\phi(\boldsymbol{x}_2)$, where $\phi(\boldsymbol{x})$ is the basis function mapping $\boldsymbol{x}$ from the input space to the kernel space. A linear hyperplane in the high-dimensional kernel feature space corresponds to a non-linear decision boundary in the original space, allowing us to model more complex datasets, that are not linearly separable in the input space.

2.3.3 Fusion Schemes

Recent work has shown an increasing interest in the application of supervector methods to speaker recognition. A supervector is any high and fixed dimensional representation of a varying-length utterance. Essentially, supervector methods involved combining many low dimensional feature vectors into a higher dimensional vector, for instance by stacking the d-dimensional mean vectors of a M-component adapted GMM into a Md-dimensional GMM supervector. These supervector models of speakers can then be used as inputs to a support vector machine. As SVMs are essentially binary classifiers, this is still only straightforward for speaker verification problems. When dealing with OSTI-SI for a small set of registered speakers, variations such as a series of One-vs-All or One-vs-One classifiers can be considered.

Table 1[9] is a compilation of results from speaker verification using different classifiers, including GMMs with various channel compensation techniques and SVM with a range of kernels, as well as fusion of these GMM and SVM methods. It gives

us a clear overview of the relative performance of these techniques; we can see that maximum performance/minimum error is achieved by the fusion scheme.

	Tuning set EER	Evaluation set EER
Gaussian Mixture Models		
GMM-UBM	8.45	8.10
GMM-UBM+EIG	5.47	5.22
GMM-UBM+JFA	3.19	3.11
Support Vector Machines		
GLSD-SVM	4.30	4.44
GSV-SVM	4.47	4.43
FT-SVM	4.20	3.66
PSK-SVM	5.29	4.77
BK-SVM	4.46	5.16
Fusion	2.49	2.05

Table 1: Performance of Various Classifiers

2.4 Adaptation Strategies

The term *model adaptation* can refer to one of two aspects of the speaker identification problem here. The first is the initial adaptation of a speaker-independent UBM constructed from a large number of speakers, to a speaker specific model using available enrolment data. The other refers to the incremental online adaptation of speaker models over time, in order to cope with drift in the voice characteristics and improve the model fit as more data becomes available. Similar approaches are used in both cases, typically based on MAP (Maximum A Posteriori), MLLR (Maximum Likelihood Linear Regression) or MCE (Minimum Classification Error) criteria. MAP adaptation is effective in combining prior information from a previously trained system with the ML estimates of model parameters obtained from new data, allowing us also to incorporate relative weighting of the existing model and the new information. Though MAP estimation does not guarantee the highest performance for reducing the recognition errors, it avoids the problem of over-training that is common with techniques such as MCE.

MLLR works by adapting the means (and possibly covariances) of the initial model by applying a linear transform. The parameters of this transformation are calculated by linear regression of the adaptation data. Though this is effective for fast adaptation with very limited new data (2-4s), if more data is available, its accuracy is surpassed by MAP, which is defined at the component level, in contrast to the pooled Gaussian transformation approach of MLLR.[14] It is possible to combine the two processes to improve performance further, for example by using the MLLR transformed means as the priors for MAP adaptation.[8] However, gains are small when sufficient data is available. Hence, we will begin here with pure MAP.

In the context of adapting a UBM to a particular speaker, the steps involved in MAP adaptation can be described as follows: given a background model and the enrolment/training data from the speaker, we first align the training vectors x_t with the components of the background model. That is, for mixture component i in the UBM, we find:

$$P(i|\boldsymbol{x}_t) = \frac{w_i p_i(\boldsymbol{x}_t)}{\sum_{j=1}^{M} w_j p_j(\boldsymbol{x}_t)} \tag{5}$$

We then use this probabilistic alignment to compute the sufficient statistics for the weight, mean, and variance parameters:

$$\begin{aligned} n_i &= \sum_{t=1}^{T} P(i|\boldsymbol{x}_t) \\ E_i(\boldsymbol{x}) &= \frac{1}{n_i} \sum_{t=1}^{T} P(i|\boldsymbol{x}_t)\boldsymbol{x}_t \\ E_i(\boldsymbol{x}^2) &= \frac{1}{n_i} \sum_{t=1}^{T} P(i|\boldsymbol{x}_t)\boldsymbol{x}_t^2 \end{aligned} \tag{6}$$

This can be interpreted as equivalent to the expectation step in the EM algorithm. These new sufficient statistics for the enrolment data are then used to update the sufficient statistics of the old background model for each mixture component i, to generate the final parameters of the now speaker-specific model.

The update formulae are as follows:

$$\begin{aligned} w_i &= [\alpha_i n_i/T + (1-\alpha_i)w_i]\gamma, \\ \mu_i &= \alpha_i E_i(x_t) + (1-\alpha_i)\mu_i \\ \sigma_i^2 &= \alpha_i E_i(x_t^2) + (1-\alpha_i)(\sigma_i^2 + \mu_i^2) - \mu_i^2 \end{aligned} \tag{7}$$

The scale factor γ is computed over all adapted mixture weights to ensure that they sum to unity. The adaptation coefficient controlling the balance between old and new estimates is defined as $\alpha_i = n_i/(n_i+r)$, where r is a fixed 'relevance factor'. For application in long-term speaker adaptation, the relative weighting of the existing model parameters is simply higher.

Experimental results in existing work have shown considerable performance gains from MAP both in the use of adapted over decoupled GMMs[6], and in compensating for variability due to time lapse.[1]

3 Preliminary Work

3.1 Proof of Concept

MFCC feature extraction followed by Gaussian Mixture Modelling emerged from our research the most reliable and widely applied methods in speaker recognition. Using these as a framework, a simple recognition system was implemented in Matlab, intended as a demonstration or proof of concept. Pairs of clean speech samples (one for the enrolment phase and one for testing) were recorded directly from a group of 20 speakers using Matlab. Each utterance was just 3 seconds in length, recorded at a sampling frequency of 44.1kHz and stored in .wav format.

The Voicebox[2] toolkit function melcepst.m is used to extract Mel frequency cepstral coefficients from each sample, and Matlab's gmdistribution.fit - an optimised EM algorithm - trains GMMs for each speaker. 12 dimensional MFCC feature vectors were used, i.e. with 12 cepstral coefficients extracted per frame. Each GMM comprised 3 mixture components and was constrained to diagonal covariance matrices.

After enrolment, the set of test utterances O were compared with each of the 20 speaker models λ_i $(i = 1...20)$ in turn, and the likelihood $P(\lambda_i|O)$ calculated for each. The identity of the speaker was assigned to the model that yielded the maximum likelihood, or the minimum 'negative log likelihood'. Figure 3 shows the negative log likelihood scores of a selection of 6 test speakers, plotted over the 20 speaker models; the red lines indicate the true speaker while the circled points tell us the speaker chosen by the system - 4 of the 6 speakers shown have been identified correctly. Taking the case of Speaker 4, we can see by inspection that model 4 gives the lowest likelihood value. With others, it is less distinct: for Speaker 10, models 10 and 18 seem to yield very similar likelihood values, with model 18 marginally lower and hence generating a misclassification.

The GMMs for this system were trained independently, rather than adapted from some common starting point; the results were therefore somewhat sensitive to the random initialisation of the model parameters. On average, however, 70% of speakers were correctly identified.

[2]VOICEBOX: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

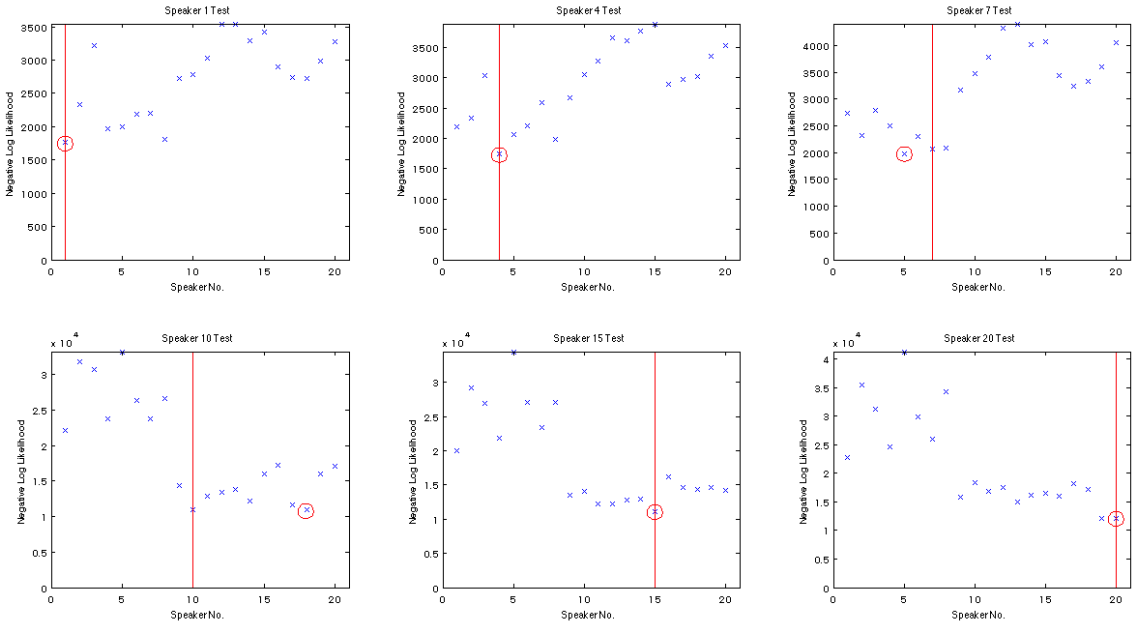
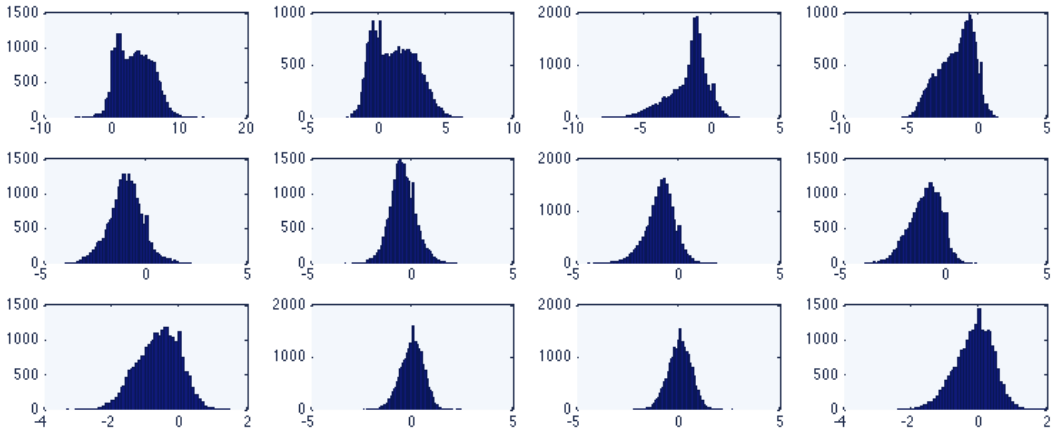


Figure 3: Proof of Concept: Plotting Model Likelihoods

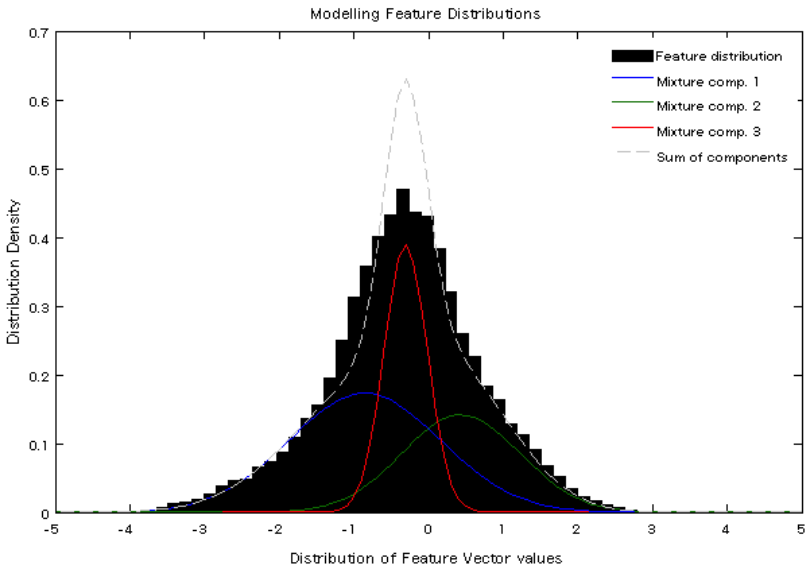
3.2 Initial Data Collection

Following the initial work described above, a request was made (to the firm for which the algorithm is being developed) for a speech corpus from the target market, Chinese speakers, recorded by the robot to enable more relevant training and testing of the speaker recognition algorithm. It was decided to focus first on improving discrimination between a large number of speakers, and optimise for this closed set problem, before extending to long-term, open-set identification of a small set of registered speakers. A collection of utterances from 60 speakers was provided, comprising 20 middle aged adults, 20 children and 20 elderly people, and split fairly evenly between male and female voices. For each speaker, three samples of speech were provided: the first was an utterance of the same phrase by all 60 speakers, recorded in a noise-free environment. This was done with the intent of enforcing some uniformity in the training of the speaker models. The second was a random utterance, again with no noise present, while the third comprised a random spoken phrase against background noise, for example a ringing phone, kitchen appliances or the presence of background speakers, allowing us to test also the robustness of the system to typical sounds in a household environment.

At this stage, it is interesting to look at the distribution of features extracted from utterances in the speech corpus. Figure 4*a* plots a histogram for each of the 12 dimensions of the feature vectors, accumulated over all available data. These MFCC feature vectors were extracted in the same way as in the earlier demonstration. We can see that most are almost symmetrically distributed, and those that are not should still be accurately modelled using GMMs with a small number of weighted components. Figure 4*b* illustrates the result of fitting a 3-component GMM, for a single dimension; the fit is reasonable, though there is room for improvement.



(a) Distributions of 24-dim MFCC feature vector



(b) GMM of Distribution

Figure 4: Modelling MFCCs using Gaussian Mixture Models

4 Front End Processing

4.1 Speech Enhancement

The utterances in our speech corpus were recorded at a sampling frequency of 48kHz, and each comprise information from four interleaved microphone channels. In order to simplify subsequent processing, the first step taken was the summation of signals from all four channels, reducing each utterance to a vector. Though information from individual channels is crucial in enabling tasks such as sound localisation, it was decided that little speaker-specific information would be lost in combining them.

Next, we found that a high level of white noise was present in the recordings, due to internal fan noise in the robot as well as other channel distortions. When utterances from a subset of 20 speakers from the new data were input to the algorithm used in our initial system, the mean accuracy of identification dropped from 70% to 45%, and just 40% when comparing all 60 speakers. Figure 5*a* illustrates the spectrogram for the raw signal of a single utterance[3] and we can see that the vibrations indicating the presence of speech are relatively indistinct, amongst significant disturbances. It is therefore crucial to reduce the level of noise and enhance the speech component of each sample, before attempting to discriminate between speakers.

Approaches to speech enhancement can be divided into three classes, namely filtering techniques, spectral restoration and model-based methods. We consider here mainly variations of the first two approaches, as the latter requires the development of a statistical representation for speech and noise, and can be much more difficult to implement and train. Evaluating the performance of these techniques using measures such as increase in SNR are difficult to apply as they require the statistics of the corresponding 'clean' speech signals. Therefore, the effectiveness of the denoising scheme will be evaluated instead based on gains in the accuracy of the speaker recognition system, and by judging the qualitative improvement in the clarity and intelligibility of speech.

[3]Spectrogram generated using the Photosounder application, to give a 2D representation of the utterance: http://photosounder.com

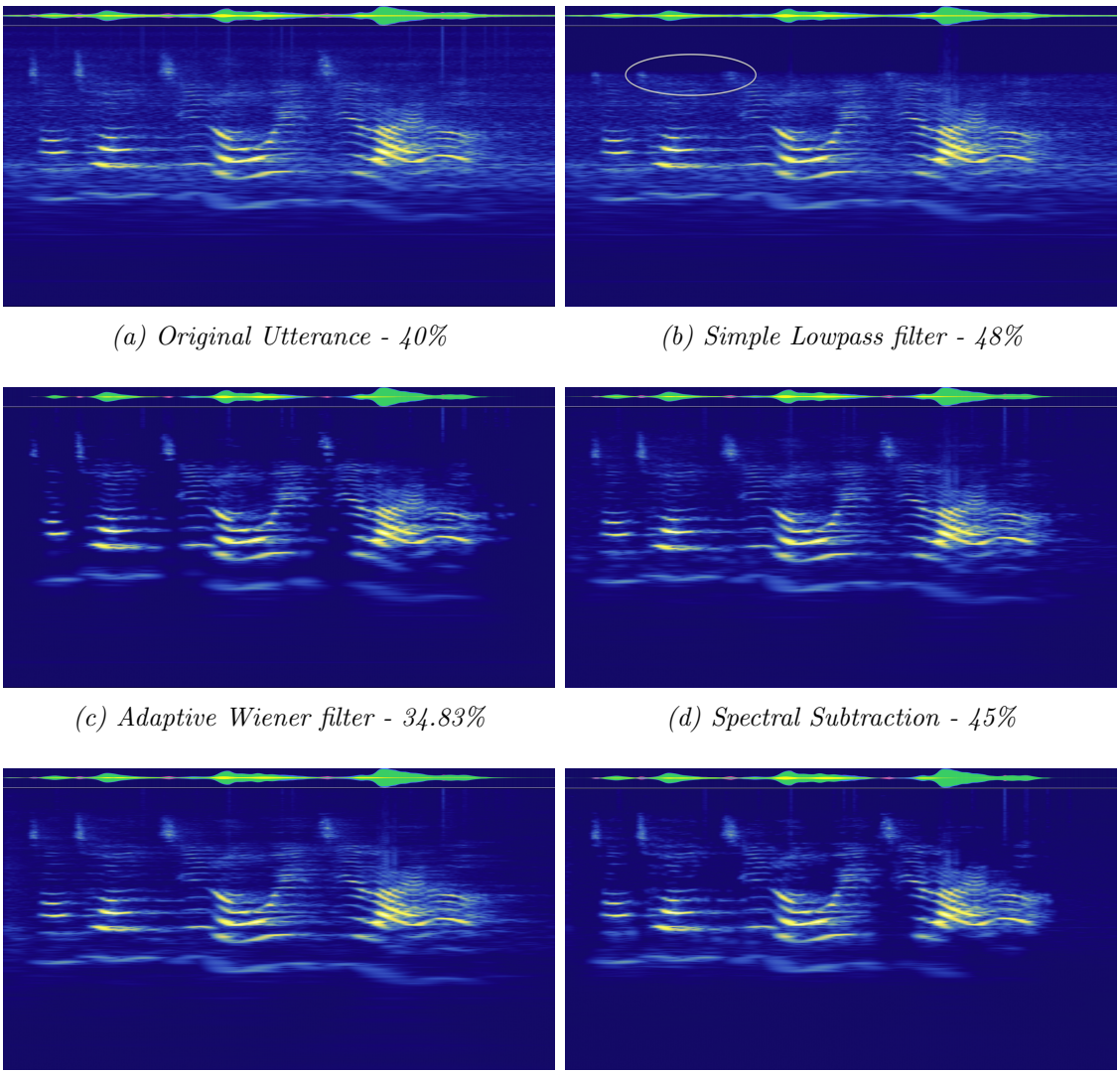
4.1.1 Filtering Techniques

The basic principle behind this class of techniques is to design a linear filter/ transformation such that, when the noisy speech is passed through, the noise component is attenuated. It was decided to begin by looking at the effect of simple lowpass filters, intended to remove high frequency stationary noise, beyond the frequency range of speech in the signal. The typical range of human speech is 300 to 4000 Hz, though harmonics in the voice can go beyond this value. The filter design tools in Matlab's signal processing toolbox were used to create a lowpass filter with a cut-off frequency here of approximately 8 kHz. The spectrogram of the filtered utterance is given in Figure 5*b*. We can see that the filter indiscriminately removes all detail above the frequency threshold (circled), but has no effect on the region containing speech. This is supported by the fact that there is little noticeable improvement in the audio quality of the utterance. Surprisingly, however, the lowpass filter yields a considerable gain in the performance of the speaker identification system, the accuracy increasing from 40% to 48% (comparing all 60 speakers). This suggests that the filter has been effective to an extent in removing noisy features, common across all utterances, and retaining information that is useful for differentiating between speakers.

The Wiener filter is one of the most fundamental approaches for noise reduction, and can be formulated in either the time or frequency domains. Here, we will be using a time-domain Wiener filter based on the algorithm by P. Scalart[4], which operates in two stages: an optimal noise estimate is first generated from the first few frames of each utterance, assumed to be silence; this estimate is then subtracted from our noisy observations. The output of this filter is shown in Figure 5*c*. We can see that the spectrogram looks considerably 'cleaner', and the speech more distinct. However, when the speaker recognition is applied, the accuracy in fact *decreases* to 34.83%. The processed utterances are also found to have a tinny quality, emphasising the fact that important speaker information has been eroded by the filter.

Spectral subtraction is based on similar principles, but often favoured over Wiener filtering for its ease of implementation: the Wiener filter is based on the *ensemble*

[4]http://www.mathworks.co.uk/matlabcentral/fileexchange/24462-wiener-filter-for-noise-reduction-and-speech-enhancement/



(a) Original Utterance - 40%

(b) Simple Lowpass filter - 48%

(c) Adaptive Wiener filter - 34.83%

(d) Spectral Subtraction - 45%

(e) MMSE-LSA Estimator - 48.33%

(f) Recursive Averaging - 46.67%

Figure 5: Investigating Denoising and Speech Enhancement Techniques

average spectra of the signal and the noise, whereas the spectral subtraction filter assumes that the signal and noise are WSS ergodic processes, and uses the instantaneous spectra of the noisy signal and the time-averaged spectra of the noise.[12] Therefore, the mean noise power is simply subtracted from the original spectrum to obtain a least squares estimate of the speech power spectrum. Figure 5*d* gives the spectrogram for the test utterance after spectral subtraction, applied using the Voicebox specsub.m function. The level of detail preserved lies somewhere between that in the case of the lowpass filter and the Wiener filter. This approach gives

relatively modest improvements in the speaker recognition system, identifying 45% of speakers correctly.

4.1.2 Spectral Restoration

Spectral restoration techniques address noise reduction in the framework of estimation theory, i.e. formulate the problem as that of obtaining a robust estimate of the spectrum of clean speech from a noisy spectrum. The most widely used estimator is the MMSE spectral amplitude estimator (MMSE-LSA) based on the method developed by Ephraim & Malah[5], and involves modelling the speech and noise spectral components as statistically independent Gaussian random variables. Figure 5*e* gives the spectrogram of the utterance after this restoration process, which uses ssubmmse.m from the Voicebox toolkit. We find that it shows similar features to that after spectral subtraction, in that the spectrogram is somewhat clearer and the speech is largely preserved, though a low level of white noise persists. The subjective quality of the final utterance is also similar, with the channel noise noticeably lower, without significant distortion in the speech. However, this method achieves a more substantial gain in performance, yielding an accuracy of 48.3% for the speaker identification system.

Finally, we look at speech enhancement via Minima Controlled Recursive Averaging (MCRA) where the noise estimate is given by averaging past spectral power values and using a smoothing parameter that is adjusted by the speech presence probability, which in turn is determined by the ratio of the local energy of the noisy speech and its minimum within a specified time window. The noise estimate is computationally efficient, robust to both the input SNR and the nature of underlying additive noise, and is characterised by the ability to follow abrupt changes in noise.[2] Here, we use an *optimally modified* log spectral amplitude (OMLSA) estimator in conjunction with Improved MCRA, a refinement of the original algorithm, developed by I. Cohen[5]. The resulting spectrogram is given in Figure 5*f*. Again, there is noticeable enhancement of the regions containing speech, and the noise component has diminished considerably. It is however possible to detect some distortion in the speech. The increase in accuracy is reasonable, with 46.67% now correctly identified.

[5] http://webee.technion.ac.il/Sites/People/IsraelCohen/Download/omlsa.m

4.1.3 Final Preprocessing Scheme

From the above experiments, the methods that yielded the best results for this corpus were speech enhancement based on MMSE-LSA estimation, and simple lowpass filtering for the removal of high frequency noise. It was therefore decided to attempt a combination of the two methods, i.e. lowpass filtering followed by MMSE-LSA estimation. Finally, the signal was scaled to ensure that the total energy of the initial and final signal is conserved. After some adjustment of the parameters involved, an accuracy of 53.33% was achieved. Figure 6 illustrates the spectrogram for the final utterance. Analysing this performance further, we find when comparing the sixty

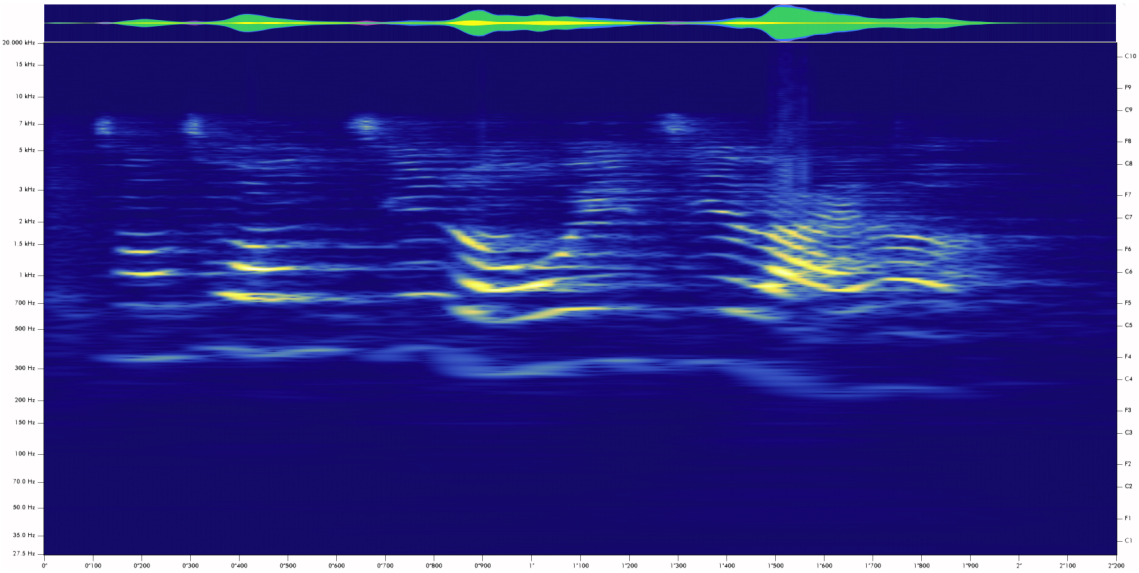


Figure 6: Final Denoising Scheme

speakers that just 7 of 20 middle aged adults were correctly identified, compared with 13 of 20 elderly people and 12 of 20 children. That is, speakers of middle age were found to be the most difficult to differentiate. In addition, as we would expect, the system is more likely to confuse people of similar age or gender. If a random subset of 10 speakers is chosen from our corpus of 60, on average 76% are correctly identified, compared with 60% when discriminating between just middle aged male adults. Hence, it was decided that a group of five middle aged adults would be chosen for the narrowed sample of 'household members' required for the development of the adaptive algorithm, as this will present the greatest challenge.

4.2 Voice Activity Detection

Another important component of front end processing for a real time speaker recognition system is voice activity detection (VAD). This is required to prevent the robot, while online, continually recording and running the speaker identification algorithm even when no speaker is present. Though a variety of features or quantities can potentially be used to signal the presence of speech, the main approaches involve either measuring the energy of the sample, or extracting periodicity. We will look at use of energy here, as it is easy to implement and has been shown to give good results. As well as the quantity, we need to decide the threshold value for detection - as with any decision rule, we must compromise between false negatives (having voice detected as pure noise) and false positives (detecting a voice when there are none). For frame by frame VAD, one approach is to calculate the energies of all frames in the utterance, select the maximum, and set the detection threshold to be 30dB below that maximum.[9] In our system, the energy is calculated instead over each 3s utterance, and the threshold set according to the desired ratio of false positives to false negatives for our corpus. Figure 7 plots both the percentage of false negatives

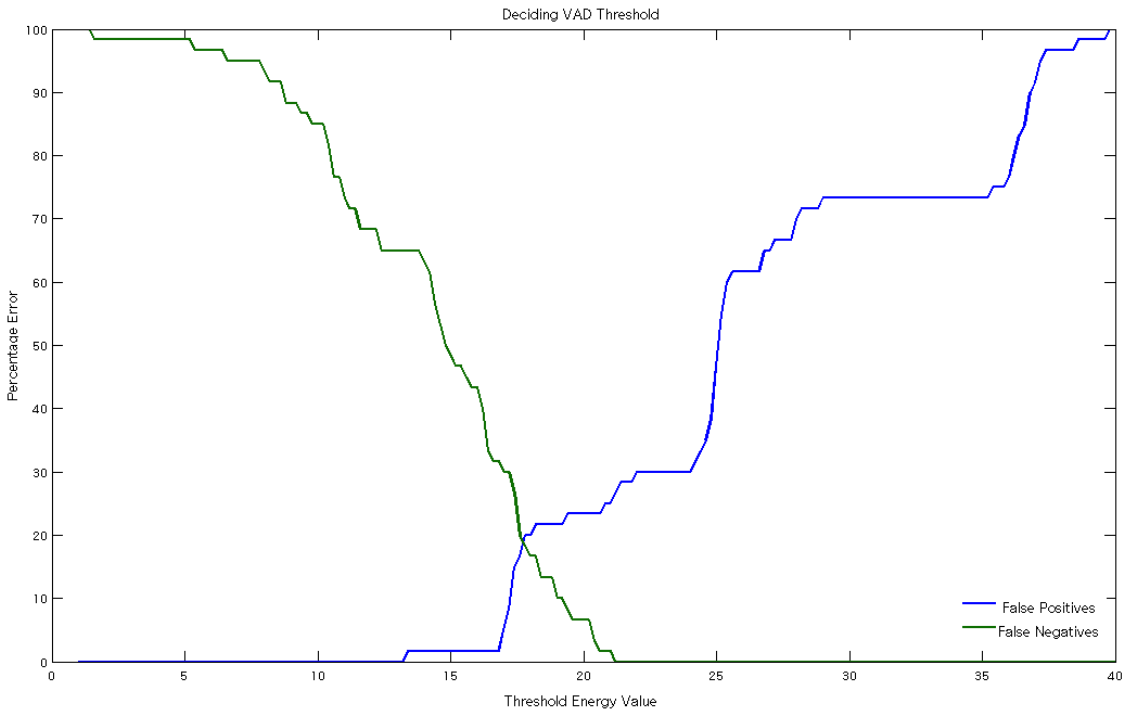


Figure 7: Choosing Threshold Value for VAD

generated by a collection of samples containing speech, and the percentage of false positives from blank samples, for a range of threshold energy values. If the costs of both classes of error were equal, a reasonable solution would be to choose the threshold to be the point at which the two lines intersect, in this case at an energy of 17.8 (or -17.8dB), to give equal error rates of approximately 20%. However, in this application, it is preferred that the VAD fails safe, indicating speech detected when in doubt to lower the chance of losing speech segments. The threshold is therefore set instead to a value of 22, corresponding to an energy of -22dB. This brings the probability of false negatives to zero for the utterances in the corpus, though yielding an increased rate of 28.3% for false positives.

The principles here can potentially be extended to incorporate an adaptive threshold and account for non-stationary noise: many existing real-time systems implement voice activity detection using LTSD, or Long-term Spectral Divergence. In this method, once the input speech has been de-noised, its spectrum magnitude is processed using windowing. Spectral changes around an N-frame neighbourhood of the actual frame are analysed using the N-order long-term spectral envelope. The VAD decision rule is formulated in terms of the deviation of the spectral envelope with respect to the residual noise spectrum, i.e. the LTSD, and a threshold is set on this value. The estimate of the noise spectrum is updated during non-speech periods.[10]

5 Refining the Speaker Model

5.1 Using Long-Term Data

Following these efforts in preprocessing the speech corpus to improve the performance of the system in discriminating between 60 speakers, we shift our focus to long-term speaker identification for a smaller group of registered voices. To this end a second data request was made for utterances from 5 middle-aged speakers, a subset of the previous 60, with a mixture of speakers both correctly and incorrectly identified by the current system. These utterances were collected every 2 to 3 days over a period of 5 weeks. A corpus of 60 3-second utterances for each of the 5 speakers was provided; the first 40 utterances contain just speech from the voice to be identified, while the final 20 were a mixture of pure speech and speech recorded against various background household noises. Considerable additional variability was introduced in terms of the state of the speakers; for example, one of the speakers developed a cold during the 5-week period, allowing us to observe the impact of such temporary changes, if any, on the accuracy of speaker recognition.

The existing algorithm was then modified as follows: the first 5 utterances from each speaker were first concatenated to provide an enrolment utterance of effectively 15s in length, sufficient to train initial GMMs for the five speakers. MFCC feature vectors were then extracted from these, and used to generate a Gaussian Mixture Model for each speaker. In the test phase, the 60 utterances was input alternately for each speaker, and compared with each of the 5 enrolled models. As before, the model with the maximum likelihood gave us the identity of the speaker.

5.2 Variation of MFCC Features

As of now, the MFCC features extracted have been 12-dimensional feature vectors, comprising just the first 12 cepstral coefficients, with no dynamic information introduced. In order to investigate whether this can be improved upon, our modified algorithm is run for a range of feature vector lengths, each with and without higher order coefficients. Table 2 summarises the results for the accuracy of each system. We find that when no dynamic information is incorporated, the performance in-

No. of Coefficients:	**8**	**12**	**16**	**24**	**30**
No dynamic information - $[y]'$	35	38	52	62	63
First derivatives - $[y\ \Delta y]'$	37	58	57	45	20
First, second derivatives - $[y\ \Delta y\ \Delta^2 y]'$	23	25	21	18	5

Table 2: Effect of MFCC Variations on Percentage Accuracy

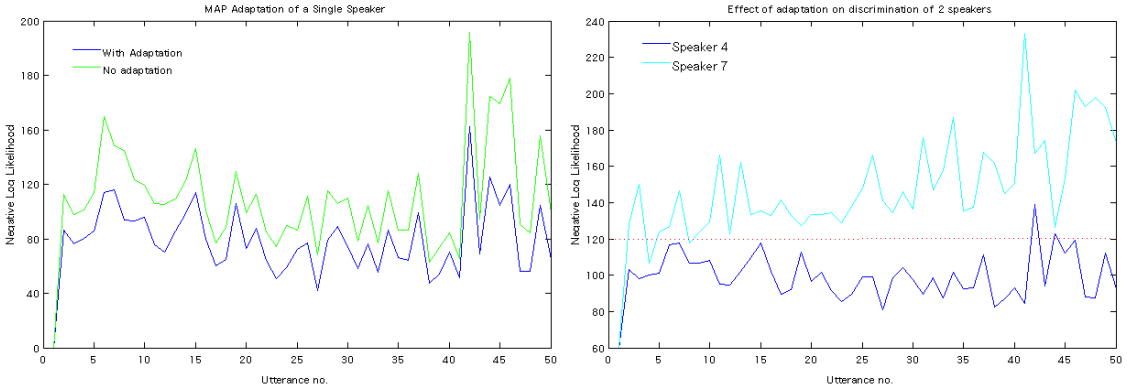
creases as the number of coefficients is increased, though the performance gains are fairly saturated by 24 coefficients. The inclusion of Δ and Δ^2 coefficients doubles and triples the length of the feature vector, respectively. In each of these cases, the maximum accuracy occurs with 12 cepstral coefficients (hence, 24- or 36- dimensional feature vectors), though the overall maximum was still achieved with no dynamic information. Therefore, the use of 24-dimension feature vectors with no higher order coefficients was decided, as increase in performance beyond this length does not justify the costs of training higher dimensional GMMs. The accuracy achieved using 12 cepstral coefficients with first order derivatives is comparable (58% rather than 62%), for the same length feature vector, so may be worth investigating in the future, as the feature extraction step is likely to be less computationally expensive in this case.

5.3 Model Adaptation

We will now go on to incorporate MAP adaptation into the speaker recognition algorithm, using the steps outlined in Section 2.4. Consider first the creation of speaker-specific models from a single universal background model: the UBM is obtained by training a GMM with data from the set of 60 different speakers initially obtained. The 15s of enrolment data for each speaker is then used to adapt the means of the UBM and create five speaker-specific models. Running the identification algorithm shows a rise in accuracy from 61.67% for decoupled GMMs, to 86.67%. This is due to the fact that the five speaker models now have a common basis and can therefore now be more fairly compared. The adaptation coefficient was chosen to give the best possible performance, finally set such that the relative weighting of the speaker enrolment data is much greater than that of the UBM. Second order

statistics, i.e. the covariance matrices, are kept the same as they have relatively little impact on the model. In fact, when we adapt both means and covariances with the adaptation coefficient set, the accuracy is just 63.33%, barely above that for the decoupled models. This may be because the adapted covariances overfit the speaker data and fail to generalize for the test utterances. Weighting the covariance updates differently is one possible solution, but is unlikely to give significant improvements.

The approach used here can be extended easily to the incremental adaptation of speaker models over time. In order to assess the effect of long-term adaptation, a single speaker model was considered to begin with. Utterances from this speaker were fed through the system one by one; after each utterance, the means of the speaker GMM were adjusted to account for the new data. Figure 8*a* plots the negative log likelihood of both the continuously adapted model and the constant model, with respect to each of the first 50 utterances. We can see that for each



(a) Adapting a single speaker *(b) Adapting 2 speakers*

Figure 8: Long-Term MAP Adaptation

utterance, the negative log likelihood values obtained are lower (hence the likelihood is higher) for the adapted model, and that there is a slight downward trend in values until around utterance 40. Adaptation therefore does appear to improve the fit of the model to the speaker. The spike in the likelihood values after utterance 40 can be explained by the presence of background noise in the last 10 samples, which the MAP algorithm then attempts to correct for. Figure 8*b* illustrates the effect of adapting two speaker models simultaneously. Test data from two speakers, 4 and 7, is input to the identification system, and the figure plots the likelihood that each utterance belongs to the Speaker 4 model. After each utterance from speaker 4, the

model is adapted; as it improves, we find that there a divergence in the values for the two speakers, and therefore they become more easily separable.

Finally, we integrate the continuous adaptation of all 5 speakers into our identification system: when an utterance is recorded, it is first identified according to the model for which it yields the maximum likelihood. If it is correctly classified, the system goes on to use this utterance to update the speaker model. If it is incorrect, the 'user' then tells the system the true identity, and the utterance is instead used to adapt the model corresponding to the real speaker. An accuracy of 93.33% is achieved, from 86.67% without long-term adaptation. Given the quality and volume of data available, this level of performance is reasonable for the closed set identification problem.

5.4 Choice of GMM Complexity

It is interesting at this point to evaluate our choice of design or constraints on the GMMs used to model speakers, specifically in terms of the number of mixture components. Figure 9 plots the accuracy of the system, both with and without incremental adaptation, for Gaussian mixture modelling with the number of mixture components in the range of 2 to 32. In order to train GMMs with more than 12 components, a small regularization term was required in gmdistribution.fit to ensure convergence. We can see from the graph that there is a large jump in accuracy when moving from 2 to 3 components - 93.33% from just 66.67%. Beyond this, for the volume of training data available here, there is almost no gain from increasing the model complexity. The maximum accuracy achieved is 93.67% by the 12-component model; this corresponds to the correct identification of one additional utterance. The choice of 3 mixture components per speaker model is therefore justified.

5.5 Open-set Identification

Finally, the algorithm must be extended to enable open-set identification: thus far, the system has not included the option of classifying an utterance as originating from a new/unknown speaker. The utterance has simply been assigned to the speaker model with the minimum negative log likelihood, with no condition set on this

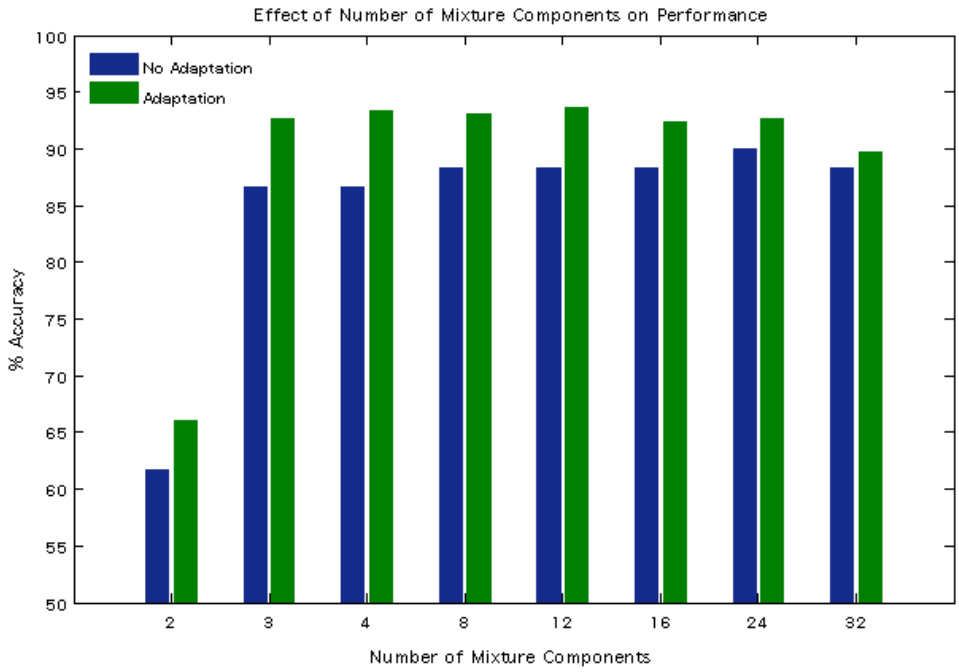


Figure 9: Effect of Number of GMM Mixture Components on Accuracy

minimum value. We therefore need to implement the second stage of the decision rule described in Section 2.1. In order to determine an appropriate value for the threshold θ on the likelihood, an ROC (Receiver Operating Characteristic) curve is generated for the corpus, as shown in Figure 10. The percentage of true positives output by the identification system (i.e. the number of utterances correctly recognized as from one of the 5 speakers) is plotted against the false positives (the number of utterances from background speakers identified as part of the registered set), over a range of threshold values from 40 to 200. The optimum threshold is that which gives the minimum error rate; on the ROC, it is the point at the minimum Euclidean distance from the point (0,1), which represents a perfect classifier. If it is assumed that the costs of misclassifying positive and negative cases are the same, the concept of equal error rate (where the false positive rate is equal to the false negative rate, denoted by the red line in Figure 10) can be used. The threshold then corresponds to the intersection between the ROC and the red line. With this criteria, the decision threshold is set to 95, giving an EER of approximately 0.74. This means just 74% of utterances from registered speakers are recognized as known; the actual identification accuracy is even lower, due to confusion within the 5 speakers. If

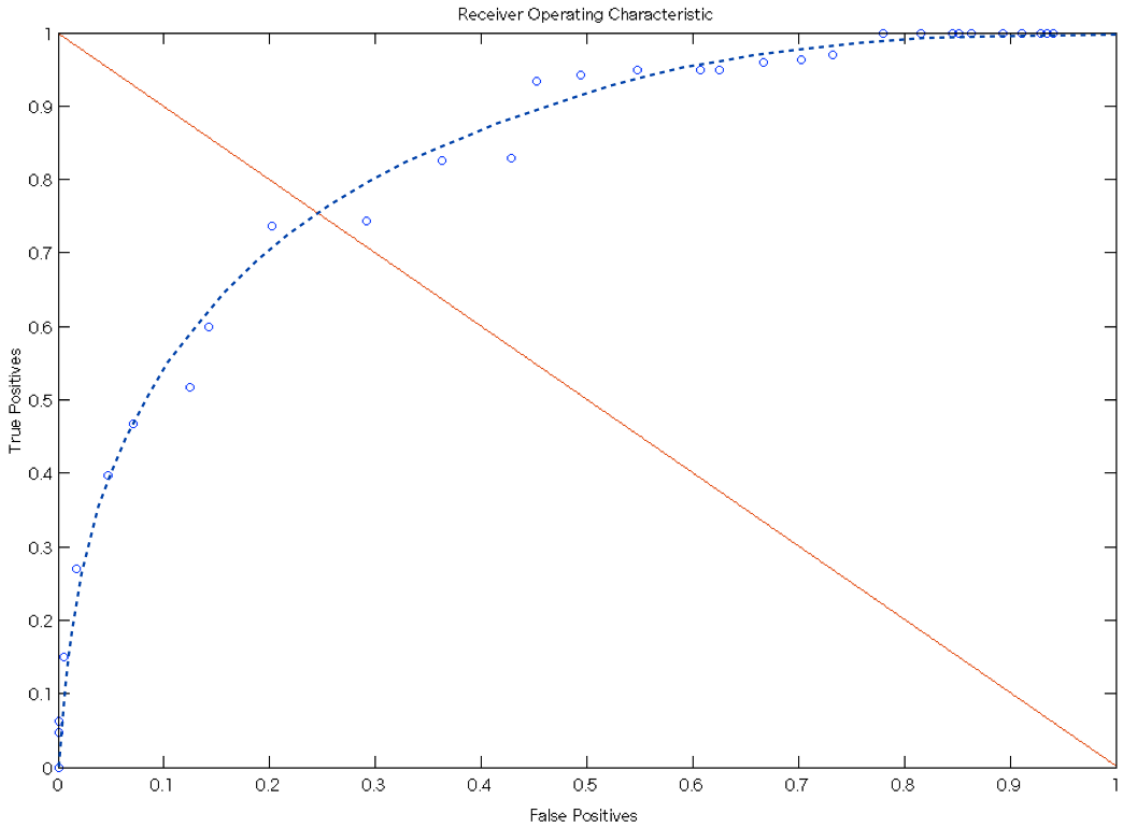


Figure 10: Receiver Operating Characteristic

instead we state that at least 95% of utterances from registered speakers should be accepted, and increase the threshold accordingly to 120, 90.3% of utterances from these five speakers are correctly identified, but the system is able to reject only 45% of imposters. A different representation of these trade-offs is shown in Figure 11, which plots the effect of the threshold value on each of the three classes of errors: false identification within the closed set, false acceptance of imposters, and false rejection of registered speakers.

Another strategy with which this can potentially be improved is by introducing the UBM as a competing model. That is, alongside the 5 speaker models, we calculate the likelihood of each utterance with respect to the UBM. If the background model yields the maximum likelihood, then the utterance is classified as belonging to an unknown speaker. This measure is in addition to the threshold on maximum likelihood determined above. The combined method increases the imposter rejection rate

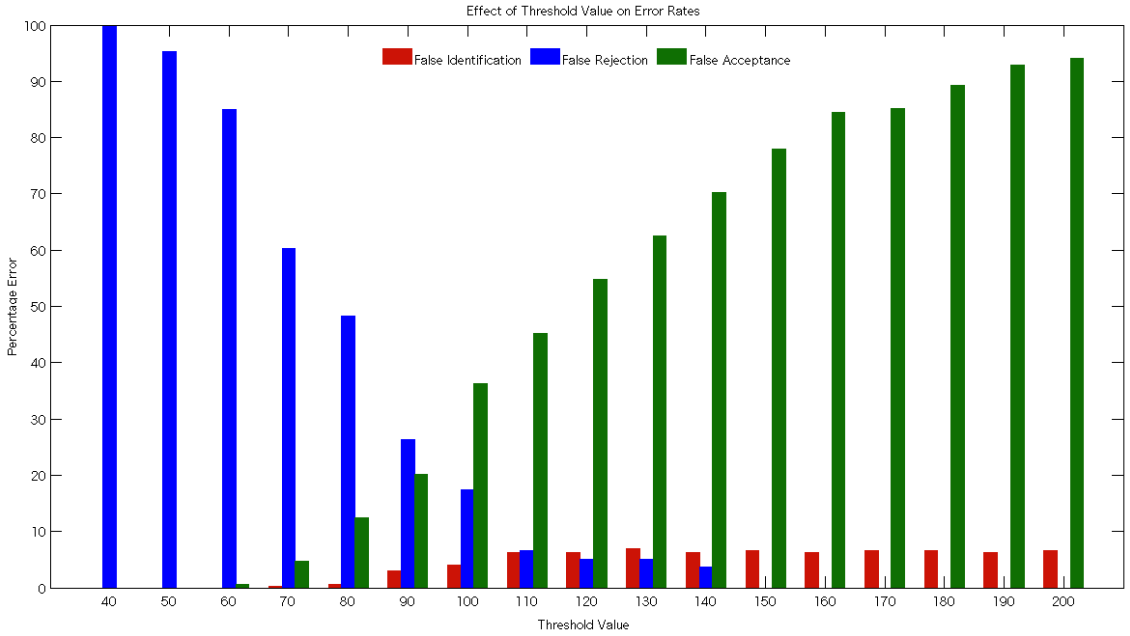


Figure 11: Effect of Threshold Value on Error Rates

from 45% to 63.7%, and has a relatively small toll on the accuracy of identification, which decreases to 89.7%. Of those not identified, 5% arise from false rejections and 6.3% from false identification.

5.6 SVM-Based Identification

As described in Section 2.3.2, systems based on the use of support vector machines to model boundaries between speakers are becoming increasingly popular. The use of SVMs was investigated here using Matlab's svmtrain and svmclassify functions. Considering just closed-set identification, in order to choose between the five registered speakers, the multi-class problem must first be formulated in terms of multiple binary classifiers. The main ways in which this is approached are via one-vs-one or one-vs-all classification, as mentioned previously. In the former, an SVM is trained for the boundary between every pair of speakers (a total of 10 are required to discriminate between five speakers). One issue in the training of these SVMs is the use of the variable length enrolment data. For example, features extracted from data for each speaker can be mapped to a fixed length space using dynamic kernels, and these transformed features are then input to the SVM. Here, we try simply truncat-

ing the feature vectors from each enrolment utterance to the same length of frames. In the test phase, for every SVM, each frame of the test utterance is classified to one of two speakers, and the output for that SVM is the speaker to which the majority of frames have been assigned. The speaker that yields the greatest fraction of the 10 available votes (one from each SVM) corresponds to the final identity of the utterance. With this method, we obtain an identification accuracy of just 41.7%.

A form of one-vs-all classification was also attempted, which involved training boundaries between each speaker and a combination of the other four speakers. Given a test utterance, for each of the five SVMs, we map each frame to either the target speaker or to the class of alternative speakers. The identity of an utterance is assigned to the speaker for which the corresponding classifier maps the maximum ratio of frames to the target speaker rather than the alternatives. This approach yields a marginally higher accuracy of 45%, as well as requiring fewer classifiers, but performance is still poor when compared with the that of a basic GMM based system (identification rate using just decoupled GMMs and no long-term adaptation was approximately 62%). Furthermore, unlike GMMs discriminative classifiers are not intrinsically adaptive. A fusion scheme could be considered, in which the fixed dimension stacked means of speaker GMMs are used as inputs in SVM training (and all classifiers regenerated each time we update the GMMs), but this does not provide sufficient data when dealing with simple 3-component mixture models. It was therefore decided that SVM-based algorithms would not be taken further.

6 Conclusions

6.1 Proposed Strategy

To conclude, our proposed algorithm for this adaptive speaker recognition system can be summarised as follows:

- ▷ Lowpass filtering and speech enhancement using MMSE-LSA estimation, followed by scaling to conserve the total energy of each utterance.
- ▷ Energy-based Voice Activity Detection, with the energy threshold chosen to err on the side of false detection, to ensure no speech is discarded.
- ▷ Extraction of 24-dim MFCC features; no dynamic information used.
- ▷ Training a Universal Background Model with 3 mixture components and diagonal covariance matrices, using data from a set of 60 speakers.
- ▷ MAP Adaptation of UBM to generate speaker-specific models using 15 seconds of enrolment data from each of 5 registered speakers.
- ▷ Open-set identification decision rule: calculate likelihood of utterance with respect to each speaker model and the background model, apply threshold on maximum likelihood (chosen in favour of minimizing false rejections).
- ▷ Incremental MAP adaptation of speaker models using each new utterance.

The algorithm is tested using 60 3s utterances from each of the 5 speakers, 60 from speakers outside this set. as well as approximately 3 minutes of blank, noisy data). Using this data to evaluate the final performance, we find 90% of speakers in the registered set are correctly identified, 64% of utterances from imposters are rejected, and 73% of blank samples discarded. Implementation of this algorithm in the robot is currently underway.

6.2 Further Work

Overall, this provides a reasonable foundation for the speaker recognition system, given constraints on the volume of data available, etc., but there is much scope for improvement. For example, further work may involve deeper analysis of the noise

spectrum in recordings in order to obtain a more customized speech enhancement solution. In terms of adaptation, pure MAP works well for the initial adaptation of the UBM (assuming sufficient enrolment data) but MLLR-MAP fusion can be considered for fast online adaptation to short utterances, over time. An alternative may be to accumulate speech from a particular user over several utterances, and use this accumulated data to carry out less frequent MAP adaptations. In addition, though this has not been implemented within the period of 300 utterances here as there is some toll on accuracy, ideally, after an initial burn-in period (judged by the extent of change in the model during each adaptation) the frequency of adaptation should be gradually reduced, particularly following correct identification.

6.2.1 Improving Imposter Recognition

The weakest aspect of performance is the rejection of speakers outside the registered set, with an accuracy of just 64%. However, it is likely that this would improve significantly with the implementation of a more robust background model. The UBM here is built from just 180 seconds of speech data, which limits its ability to generalize. Additionally, incorporating some form of score normalization should also improve overall performance, though the use of the UBM as a competing model can be interpreted as a kind of 'world model normalization'. Methods that have proven to give significant gains include zero normalization (which compensates for inter-speaker variability), test normalization for inter-session variability, or combinations of the two.[11]

6.2.2 Extensions

The results and principles of speaker recognition can be used by the robot in a number of ways. Most extensively, in combination with speech recognition to adapt to users and improve accuracy, but also to enable, for example, emotion detection from voice or the recognition and tracking of multiple, simultaneous speakers. All of these contribute to the context information available to the robot, and go a long way in enhancing human-robot interaction.

References

[1] H. Beigi (2009). 'Effects of Time Lapse on Speaker Recognition Results'.

[2] Benesty, et al. (2008). 'Springer Handbook of Speech Processing'.

[3] Biometrics Institute Limited (2013). 'Types of Biometrics'. Available at: http://www.biometricsinstitute.org/pages/types-of-biometrics.html.

[4] B. Byrne (2013). '4F10: Statistical Pattern Processing'.

[5] Y. Ephraim & D. Malah (1985). 'Speech Enhancement using a Minimum Mean-Square Error Log-Spectral Amplitude Estimator'.

[6] J. Fortuna, et al. (2005). 'On the Use of Decoupled and Adapted Gaussian Mixture Models for Open-Set Speaker Identification'.

[7] J. Fortuna, et al. (2004). 'Relative Effectiveness of Score Normalization Methods in Open-Set Speaker Identification'.

[8] S. Goronzy & R. Kompe (1999). 'A Combined MAP + MLLR Approach for Speaker Adaptation'.

[9] T. Kinnunen & H. Li (2009). 'An Overview of Text-Independent Speaker Recognition: from Features to Supervectors'.

[10] J. Ramirez, et al. (2004). 'Voice Activity Detection with Noise Reduction and Long-Term Spectral Divergence'.

[11] R. R. Shou-Chun Yin and & P. Kenny (2005). 'Adaptive Score Normalization for Progressive Model Adaptation in Text Independent Speaker Verification'.

[12] S. V. Vaseghi (2000). 'Spectral Subtraction'. In: Advanced Digital Signal Processing and Noise Reduction.

[13] P. Woodland (2013). '4F11: Speech and Language Processing'.

[14] S. Young, et al. (2006). 'The HTK Book'.

A Risk Assessment Retrospective

As anticipated, the level of risk involved over the course of this project was limited, as the work was entirely computer-based. Appropriate measures were taken to ensure that the strain from extensive computer use was limited.

In addition, the work required little direct contact with the robot while online, hence no additional precautions were necessary.