KUNGL
TEKNISKA
HÖGSKOLAN

Centre for
Speech Technology

Text Independent Speaker Verification Using Adapted Gaussian Mixture Models

Textoberoende talarverifiering med adapterade
Gaussian-Mixture-modeller

Daniel Neiberg
Centre for Speech Technology (CTT)
Department of Speech, Music and Hearing
KTH, Stockholm, Sweden
supervisor: Håkan Melin

2001-12-11

Abstract

The primary goal of this master thesis project is to implement a text independent speaker verification module for GIVES. Secondary goals are to implement a fast scoring method and compare performance between the implemented text independent module and an available text dependent module. The project also includes a literature study. The text independent module is based on adapted Gaussian Mixture Models and the adaptation equations are derived. Evaluation results show that the text independent module and the text dependent module have almost equal performance on a text dependent recognition task. The results are analyzed and summarized, and improvements are suggested. Unfortunately, the fast scoring method did not work together with all the components in GIVES.

Sammanfattning

Det primära målet med detta examensarbete är att implementera en textoberoende talarverifieringsmodul för GIVES. Sekundära mål är att implementera en snabb verifieringsmetod och att jämföra prestanda mellan den implementerade textoberoende modulen och en befintlig textberoende modul. Examensarbetet inkluderar också en litteraturstudie. Den textoberoende modulen baseras på adapterade Gaussian-Mixture-modeller och adapteringsekvationerna härleds. En utvärdering visar att den textoberoende modulen och den textberoende modulen har likvärdiga prestanda på en textberoende igenkänningsuppgift. Resultaten analyseras och summeras, och förbättringar föreslås. Tyvärr så fungerade inte den snabba verifieringsmetoden med alla komponenterna i GIVES.

Contents

Chapter 1

Project Specification

1.1 Background

GIVES (General Identity VErification System) is a software package built for research in automatic speaker verification at the Centre for Speech Technology (CTT) and Department of Speech, Music and Hearing, KTH. Speaker verification is the task of deciding whether a speech utterance is delivered by a given claimant speaker or not. Existing modules for GIVES are mainly targeted for text dependent speaker verification and the main goal of this master thesis project is to develop a text independent module. The project is supervised by Håkan Melin, who is also the main developer of GIVES. The examiner is professor Björn Granström. Formally the project is done at Department of Speech, Music and Hearing, KTH, but by commission of CTT.

1.2 Specification and Goals

The goal of this project is to implement and evaluate a text independent speaker verification system module for GIVES. The system will be based on adapted Gaussian Mixture Models (GMM) inspired by Reynolds, Quatieri and Dunn [1] which is considered as state-of-the-art. The system will also incorporate a special fast scoring procedure. Finally an evaluation and performance comparison against a text dependent system (described in Section 4.1) will be carried out using the GANDALF speech database (discussed in Section 4.3). Early experiments will be performed in Matlab and the final implementation will be done in C++. The project also includes a literature study and the main part is presented in Chapter 2. The project will not cover front-end processing (i.e. feature extraction from speech signals). Project duration are 20 weeks by definition of master thesis projects at KTH.

Chapter 2

An Overview of Speaker Verification

2.1 Introduction

Identity verification is a part of everyones life. An ever increasing number of personal identification codes (PIN-codes) are used everywhere and written signature based verification is an integrated part of our modern society. The recent development of technology has raised the interest in science fiction inspired *biometric* verification. That is verification based on individual biological features such as fingerprints, retinal scan, written signature, DNA-analysis, smell and voice. The perhaps greatest advantage of biometric verification is that you can forget a PIN-code, but you will never "forget" your body. Moreover, if the biometric properties are unique then verification could be rather safe if the technology can measure these properties accurately. Traditional verification can also be combined with biometric verification in order to make the verification even more safe.

The widespread use of telephone systems, fixed and mobile, and the services provided through these, raise the need for verification based on a speaker's voice. Recently, the advance of technology and theory has made speaker verification possible. Some overview papers of speaker verification are: Melin [5], Doddington [9], Campbell [14] and Furui [15].

2.2 Speaker Verification

Speaker verification is the task of deciding whether a speech utterance is delivered by a given claimant speaker or not. More formally, it is the task of deciding, given a speech signal $\mathbf{x}$ and a hypothesized speaker S, whether $\mathbf{x}$ was spoken by S. This is also referred to as speaker detection or single-speaker detection. The binary decision can be reformulated as a hypothesis test between the following statements:

H_0 : $\mathbf{x}$ is from the hypothesized speaker.

H_1 : $\mathbf{x}$ is not from the hypothesized speaker.

Then the decision in an optimal manner is [25]:

$$T(\mathbf{x}) = \frac{f(H_0|\mathbf{x}))}{f(H_1|\mathbf{x})} \begin{cases} \geq \eta, & \text{accept } H_0 \\ < \eta, & \text{reject } H_0 \end{cases} \tag{2.1}$$

if the probability functions $f(..)$ are known exactly and for a threshold η. $T(\mathbf{x})$ is denoted as the test ratio[1]. Some common choices of $f(..)$ are Hidden Markov Models (HMM), Gaussian Mixture Models (GMM) and Artificial Neural Networks (ANN). A typical speaker verification system operates as follows (Figure 2.1): The model defined by the function $f(H_1|\mathbf{x})$ is trained on speech from many different speakers and it is denoted as the Universal Background Model (UBM) or the reference model. The speaker model defined by $f(\mathbf{x}|H_0)$ is simply trained on the speaker's voice in a procedure denoted as enrollment. Finally, a speaker claims an identity, a test utterance is recorded and a decision is made.

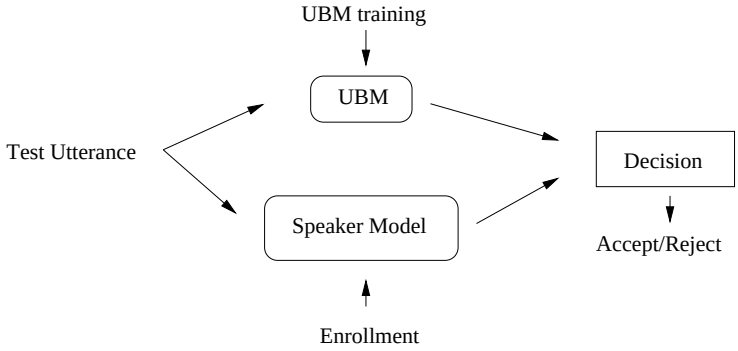

Figure 2.1: A typical speaker verification system.

2.3 Text-dependence

A speaker verification system operates in either text dependent (TD) mode or text independent (TI) mode. In TD mode the speaker uses the same utterance during enrollment and testing while in TI mode, the test utterance is different form the utterance used during enrollment. This boundary is not clear and some systems, such as digit-based, lies somewhere between TD and TI.

2.3.1 Text Dependent Verification

If the system demands that the speaker uses the same utterance during enrollment and testing it's called a TD system. The utterance could be a fixed phrase for all speakers or an individual phrase. In TD systems, the speaker model will cover specific characteristics from both the speaker and the text. Such a detailed model require less training data than the more general model in a TI system and, therefore, TD systems generally achieves good performance. TD

[1]The test ratio is often called the "likelihood ratio". The author consider the use of the term likelihood ratio unnecessary since the introduction of likelihood functions may be confusing.

systems may be quite susceptible to tape recordings which could be used by an impostor to bypass the system.

2.3.2 Text Independent Verification

If the speaker is allowed to use different utterances during enrollment and testing, the system is called TI. A TI system require more training data than a TD system because phrase specific characteristics are not available. Therefore, TI systems often exhibit worse performance than TD systems. Of course, a true TI system may be very susceptible to tape recordings if the system is used in, for example, a telephone bank. The main usage of a true TI speaker verification system lies in bugging technology. Some systems prompt for an unpredictable text to be spoken and checks, by using speech recognition, whether the speaker utterance was the correct one. For example, the system prompts for words during testing which are composed by phonemes used during enrollment. Such a system is very secure to tape recordings but it's also quite difficult to implement.

2.3.3 Digit-based Verification

In digit-based verification, digits are used to assemble an utterance which may be, for example, a password, an account number or a telephone number. If the sequence of digits is prompted and the system checks whether the correct sequence was spoken then the system will be quite secure to tape recordings. Moreover, a digit-based system yields good performance because of the limited set of possible words, i.e. the word specific characteristics is limited to digits. This limitation makes it more easy to build accurate speaker models compared to TI verification. Such a system is also quite easily implemented.

2.4 Performance Measures

There are two possible situations that may occur in single-speaker detection: if the claimed identity is the same as the speaker's true identity then the speaker is known as the *true speaker* or the *client speaker*, or if the speaker tries to fool the system by claiming an existing client speaker identity then the speaker is known as the *impostor* or the *non-client speaker*.

If an impostor is accepted by the system, this is called false acceptance (FA), and if a true speaker is rejected, this is called false rejection (FR). Often there is a tradeoff between FA-rate (E_{FA}) and FR-rate (E_{FR}) that depends explicitly on the decision threshold η. It is common to visualize FR-rate as a function of FA-rate in a Detection Error Trade-off plot (DET plot) [16].

There are also several scalar performance measures. The perhaps most common is called equal error rate (EER) [20]. EER is received by adjusting η until $E_{FA} = E_{FR} = ERR$. Operational systems usually don't have equal E_{FA} and E_{FR} since η is fixed, and may have been set to favor either E_{FA} or E_{FR}. In the fixed threshold case, performance can be measured by the geometric mean error defined as

$$E_{GM} = \sqrt{E_{FA} \cdot E_{FR}}.$$

However, the E_{GM} is quite rough. Another performance measure is formulated as a detection cost function. The detection cost, C_{det}, is defined as [9]

$$C_{det} = C_{FR}E_{FR}P_{true} + C_{FA}E_{FA}(1 - P_{true})$$

where C_{FR} and C_{FA} are the costs of a false rejection and false acceptance and P_{true} is the *a priori* probability of a true speaker. The C_{det} measure has the advantage of modeling the application, where perhaps low E_{FA} is more important than E_{FR}, and, hence, produces a more meaningful measure.

2.5 Setting the Threshold

For N speakers $n = 1, \ldots, N$ each speaker can have a speaker dependent threshold η_n or a common speaker independent threshold η. The conventional approach is to use a speaker independent threshold because the result can easily be presented as a DET-plot. However, a speaker independent threshold will produce worse performance compared to speaker dependent thresholds if the score value distributions for each speaker differ too much. Whether a speaker independent or speaker dependent threshold is chosen, the setting of the threshold is not a trivial problem. Actually, there is currently no good way to set the threshold a priori. However, if the system is run against a large speaker database the threshold can be set *a posteriori*, i.e. by calculating FA/FR-rates for a given threshold. Then the trade-off between FA-rate and FR-rate must be taken into consideration. If low FA-rate is crucial then a high FR-rate must be accepted which can be annoying for the user. On the other hand, if pleased users are more important then a more insecure verification must be accepted.

2.6 Speaker Variability

Speaker verification makes use of the fact that speakers' voices sound differently from each other. The variation in voices between people is called inter-speaker variability. If an impostor's voice is similar to a client speaker then the FA-rate may raise and, therefore, inter-speaker variability is closely related to FA-rate. The variation of one person's voice from time to time is called intra-speaker variability. This variation could depend on several things, for example if the person has a cold. The FR-rate depends mainly on intra-speaker variability.

Empirical tests have shown that most systems behave well for a majority of a target population but not for a minority [23]. This minority may be divided into subpopulations with animal names [9]. Speakers that contribute to a minority of all FR-errors and dominate the population are termed *sheep*. Speakers that have trouble with the system are termed *goats*. They tend to contribute to most of the FR-errors while in minority. Target speakers that are unusually susceptible to many different impostors are called *lambs*. If the impostor population have some speakers that have unusually good success to mimic many different target speakers, then these are called *wolves*. The reason for dividing a population into these categories[2] is to study and understand these speaker inhomogeneities. For example, if goats and lambs are detected during enrollment [24], then the system can take an appropriate action such as demanding more training.

[2]The categories are not necessarily disjunct sets.

2.7 Channel Distortion

A major challenge in speaker verification is the fact that different microphones, noise and channel transmission *color* the speech. The problem arises when one speaker uses one handset in enrollment and another in verification. Then the test utterance will be scored against a model that is trained with a different color and FA/FR-rate will increase. From the system point of view this is the same as increased inter- and intra-speaker variability. If speaker verification is used over a telephone network then this is a difficult problem. There exists various methods for channel normalization and most of these operate in the spectral domain. A different approach used by Reynolds *et al.* [1] is handset score normalization (*hnorm*). Since hnorm works in a different domain than spectral methods, these techniques can be combined. Of all possible distortions the handset often contributes the most and therefore the total channel distortion is denoted as just *handset*.

2.8 System Components

A speaker verification system generally consists of four modules:

1 An analysis module which extracts speaker dependent features from a speech signal. A standard method is to compute spectral parameters such as mel-frequency cepstral coefficients (MFCC) or linear prediction cepstral coefficients (LPCC) in a window for every 10 ms of speech which result in a stream of feature vectors.

2 A modeling module which builds a model from the feature output of the analysis module. Common models are based on HMM, GMM or ANN.

3 A scoring module which computes how well a feature output from a utterance fits the model in the modeling module.

4 A decision module which, from the output of the scoring module, accepts or rejects the speaker.

Chapter 3

The UBM-GMM system

3.1 System Environment

GIVES is a software package for research in automatic speaker verification and it is mainly developed by Melin at the Centre for Speech Technology (CTT) and the Department of Speech, Music and Hearing, KTH. The UBM-GMM system is implemented with Blitz++ library [22] as a module in GIVES. Blitz++ is a C++ library which supports dense vectors and multidimensional arrays.

3.2 An Overview of System Components

GIVES provides a framework for various components. Figure 3.1 shows an overview of the main system components. Gender detection and signal analysis components are available in GIVES. In the following Sections, each component is described in detail.

3.3 Signal Processing

In order to extract relevant feature vectors from a signal, several standard methods available in GIVES are used. The choice of signal processing is based on previously good results [1, 19]. First, the signal is segmented into speech and silence. Then, silence segments are thrown away and the speech segments are pre-emphasized with a coefficient 0.97. A 12-element mel-frequency cepstral coefficient (MFCC) vector is computed from the frequency interval 300-3400 Hz every 10 ms over a Hamming-window of length 25.6 ms. The zeroth element, which is a measure of energy, is excluded and the MFCCs are liftered with a constant 22. Delta and acceleration coefficients of the MFCC vectors are computed and appended to the feature vectors so that the resulting vector length is 36.

Since the speech signal is often transmitted through different telephone channels or microphones, the MFCC will include other characteristics than those of the speaker. Therefore, the MFCC must be channel normalized. One simple way to do so is cepstral mean subtraction [29] (CMS), where the mean, computed over each utterance, of the MFCCs is subtracted from each MFCC vector.

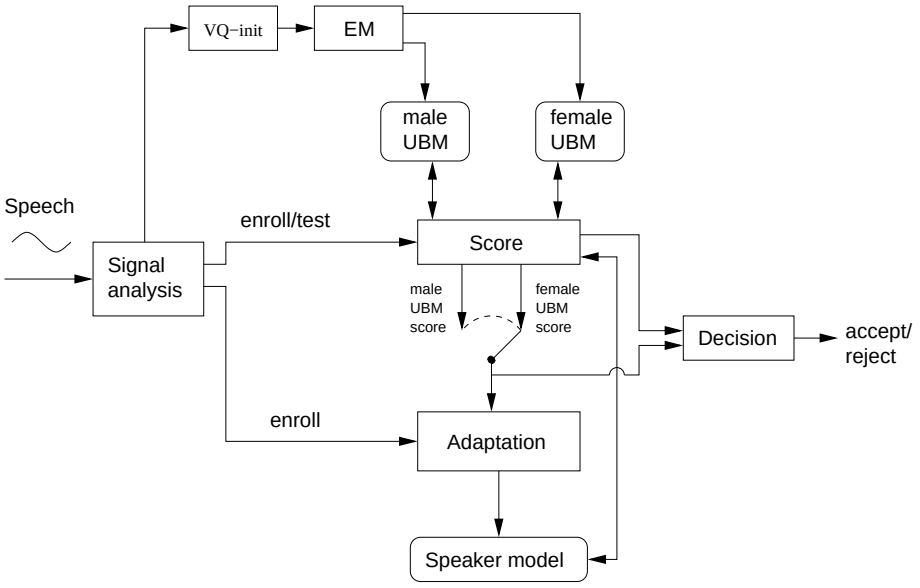

Figure 3.1: The UBM-GMM system components.

CMS is a quite rough method and doesn't remove channel characteristics completely. Therefore a score normalization method is also used, which is described in Section 3.9.

3.4 Gender Detection

To evaluate the test ratio in Section 2.2, at least two models are needed. One model that represents the speaker and one Universal Background Model (UBM) that represents all possible speakers. It is important that training data for a UBM has the same set of subpopulations as the population that is meant to use the system. However, a single UBM is not always the best choice. Male and female speech differ significantly in pitch and also in vocal tract length, linguistic and stylistic use. Therefore, gender dependent UBMs yield better performance than a gender-independent UBM [17].

When using gender dependent UBMs, there are at least two ways to choose a UBM during testing if there is no prior knowledge of the speaker gender:

1 The system detects which gender the enrolled speaker has by scoring the enrollment utterances against the male and female UBM. During testing, the test ratio is computed between the claimed speaker and its associated UBM, as determined at enrollment time.

2 The speaker model doesn't have an associated UBM and gender detection is performed both during enrollment and testing.

The advantage of the first alternative is simplicity. However, if a female impostor claims to be a male client, the impostor test utterance is scored against

the male UBM. It is not unlikely that the female impostor's voice is more equal the male client's voice compared to the voice of an average male (represented by the male UBM). This makes it easier for an impostor to fool the system if an impostor claims to have the opposite gender than the impostor's actual one. The second alternative doesn't have this disadvantage but if the gender detection fails, performance is again poor. Fortunately, gender detection works quite well (1% error per utterance for a typical evaluation) and in this system, alternative two is chosen. This method is known as dynamic cohort and the female and male subpopulations are called *cohort speakers* [15]. This method can also be expanded to include more subpopulations, for example, age or dialect subpopulations.

3.5 Gaussian Mixture Models

To implement the hypothesis test in Section 2.2 the probability function $f(\lambda|\mathbf{x})$ must be chosen. For text dependent applications a HMM (Hidden Markov Model) is preferable and gives good performance but for text independent client verification GMMs (Gaussian Mixture Models) have been the most successful so far [26].

The GMM density is given by

$$f(x_t|\lambda) = \sum_{k=1}^{K} w_k \mathcal{N}(x_t|m_k, r_k) \tag{3.1}$$

where

$$\mathcal{N}(x|m_k, r_k) \propto |r_k|^{1/2} \exp\left[-\frac{1}{2}(x-m_k)^t r_k (x-m_k)\right] \tag{3.2}$$

$$\lambda = (w_1, \ldots, w_K, \theta_1, \ldots, \theta_K) \quad \theta = (m_1, \ldots, m_K, r_1, \ldots, r_K).$$

The GMM density is simply a weighted summation of K unimodal Gaussian densities where $\sum_{k=1}^{K} w_k = 1$. m_k is a D x 1 vector and r_k is the inverse of a D x D covariance matrix. Figure 3.2 shows a simple GMM density. In this thesis report only diagonal covariances are used. The reasons for this are: A low order full-covariance GMM can also be well approximated by a high order diagonal GMM [1]. Also, a diagonal GMM requires less computational effort since repeated inversions of r are not required.

The test ratio may be expanded by using Bayes rule

$$T(\mathbf{x}) = \frac{f(\lambda_{client}|\mathbf{x}))}{f(\lambda_{UBM}|\mathbf{x})} = \frac{g(\lambda_{UBM}) f(\mathbf{x}|\lambda_{client})}{g(\lambda_{client}) f(\mathbf{x}|\lambda_{UBM})} \tag{3.3}$$

where $g(\lambda)$ is the prior density. In fact, the prior density is assumed to be equal for the UBM and the client model (discussed in appendix B). With this simplification in mind the following is assumed: For a set of $t = 1, \ldots, T$ independent and identically distributed (i.i.d) feature vectors x_t with dimension D and distribution f, the test ratio is

$$T(\mathbf{x}) = \frac{f(\mathbf{x}|\lambda_{client})}{f(\mathbf{x}|\lambda_{UBM})} = \frac{\prod_{t=1}^{T} f(x_t|\lambda_{client})}{\prod_{t=1}^{T} f(x_t|\lambda_{UBM})} \tag{3.4}$$

The set of feature vectors is often very large and, hence, the value of $f(..)$ is often very small. Therefore, it is common to compute the logarithm of the test ratio instead. The log-test ratio is given by

$$\Lambda(\mathbf{x}) = \log f(\mathbf{x}|\lambda_{client}) - \log f(\mathbf{x}|\lambda_{UBM}). \tag{3.5}$$

where $\log f(\mathbf{x}|\lambda)$ is computed as

$$\log f(\mathbf{x}|\lambda) = \sum_{t=1}^{T} \log \left(\sum_{k=1}^{K} w_k \mathcal{N}(x_t|m_k, r_k) \right). \tag{3.6}$$

In the UBM-system a normalized log-test ratio is used. Normalization is given by dividing the log-test ratio by T [25].

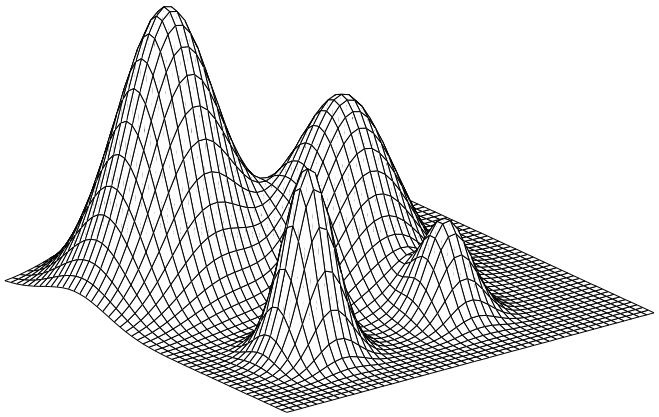

Figure 3.2: A 2-dimensional GMM density with 4 terms.

3.6 The UBM

3.6.1 EM-training

For a set of i.i.d. feature vectors $\mathbf{x} = \{x_1, \ldots, x_T\}$, the *maximum-likelihood* (ML) estimate of the parameters of a GMM, λ^*, is [2]

$$\lambda^*_{ML} = \underset{\lambda}{\operatorname{argmax}}\, f(\lambda|\mathbf{x}) = \underset{\lambda}{\operatorname{argmax}}\, f(\mathbf{x}|\lambda).$$

The expectation-maximization (EM) algorithm [3] finds λ^*_{ML} by iteratively estimating λ so that $f(\mathbf{x}|\lambda^{i+1}) > f(\mathbf{x}|\lambda^i)$ for each iteration i.

The prior probability for a mixture component k in a GMM, given a feature vector x_t is

$$c_{kt} = f(k|x_t, \lambda) = \frac{f(k, x_t|\theta_k)}{f(x_t|\lambda)} = \frac{w_k \mathcal{N}(x_t|\theta_k)}{\sum_{l=1}^{K} w_l \mathcal{N}(x_t|\theta_l)} \quad k = 1, \ldots, K \tag{3.7}$$

and the probabilistic count for new data is

$$c_k = \sum_{t=1}^{T} c_{kt}. \tag{3.8}$$

Then the EM reestimation equations which maximize the log-test for GMM parameters are [2]:

$$\hat{w}_k = \frac{c_k}{T} \tag{3.9}$$

$$\hat{m}_k = \frac{\sum_{t=1}^{T} c_{kt} x_t}{c_k} \tag{3.10}$$

$$\hat{r}_k^{-1} = \frac{\sum_{t=1}^{T} c_{kt}(x_t - \hat{m}_k)(x_t - \hat{m}_k)^t}{c_k} \tag{3.11}$$

The EM algorithm is guaranteed to monotonically converge to a local maximum [3]. In practice, convergence is assumed if $\log(f(\mathbf{x}|\lambda_i)) - \log(f(\mathbf{x}|\lambda_{i-1})) > \epsilon_{EM}$. Reynolds *et al.* [1] claims: "Generally, five iterations are sufficient for parameter convergence", but without motivation. This is probably an empirical result for training large UBMs. However, Xu and Jordan [11] claims that if mixture elements are poorly separated then convergence is slower and this could be the case if too many mixture terms are used. During the development of this system, there was some numerical problems (discussed in Appendix A). To avoid this problem, a variance floor of 0.001 was applied. This means that the elements of r_k^{-1} are not allowed to be smaller than 0.001.

3.6.2 Initialization

Before the UBM can be trained, it is crucial to initialize the UBM: Since the EM-algorithm is quite slow and a UBM must be trained on a huge amount of data, a fast initialization makes the EM-algorithm converge faster. Furthermore, a good initialization makes the EM-algorithm converge close to the global maximum [12, 25]. A good candidate for initialization is vector quantization (VQ) [21] because it encodes feature vectors in regions that correspond to the unimodal Gaussian densities in the GMM (as illustrated in Figure 3.3). Of course, any clustering technique will accomplish the task but with different results.

In this system, the simplest VQ is implemented. That is a linear VQ, trained iteratively via the generalized Lloyd algorithm with the same dimension as the number of mixtures, K, in the GMM. This is done as follows:

Step 1 Deterministically select K feature vectors as initial cluster centers $y_1, \ldots, y_K$ by setting $y_k = x_n$ where $n = 1 + (k-1)\lfloor T/K \rfloor$ and $k = 1, \ldots, K$. Set $i = 1$.

Step 2 Divide all feature vectors into K disjoint regions, $R_1, \ldots, R_K$, such that a feature vector, x_t, is an element of R_k if

$$d(x_t, y_k) \leq d(x_t, y_l) \quad \text{for all} \quad l = 1, \ldots, K \quad l \neq k$$

where $d(.,.)$ is a distance function.

Step 3 Set each y_k to the mean of the feature vectors in the corresponding region R_k.

Step 4 If a region R_k contains less then U feature vectors, move y_k to $y_l + \epsilon$ where l is the region containing the largest number of feature vectors.

Step 5 For each x_t in a region R_k, calculate $d_k = d(x_t, y_k)$ and $d_{avg}^{i+1} = \frac{\sum_{k=1}^{K} d_k}{T \cdot K}$.

Step 6 set $i = i + 1$.

Step 7 Repeat step 2-6 until $d_{avg}^{i} - d_{avg}^{i-1} < \epsilon_{avg}$.

The distance function $d(.,.)$ can be arbitrarily chosen. Squared error, average squared error, Mahanalobis and SNR distance measures have been implemented. In practice, only the Mahanalobis distance was used. The reason for this is that Mahanalobis takes the variance into consideration when the distance is computed which may result in a more accurate GMM-initialization. In order to save time without reducing available data too much, every second feature vector is used in the training set for the VQ and the rest is encoded into regions used for GMM initialization, once the VQ has been trained.

To initialize the GMM, each Gaussian density, k, is set to the mean and variance in the corresponding VQ-region k. The weights, w, are set to T_k/T where T_k is the number of feature vectors in region k. Again, regions containing less then U feature vectors are removed and replaced by the largest region with recalculated weights. U must be larger than one because the variance of a single component is zero.

3.7 The Speaker Model

The speaker model can be estimated in the same way as the UBM. However, there exists a better approach. Assume that the speaker model doesn't differ too much from the UBM, then the speaker model could be adapted from the UBM. Moreover, since speech data in enrollment is sparse, ML estimation tends to overtrain model parameters. A model is overtrained if it fits irrelevant details in training data. If an adaptation approach such as *maximum a posteriori* (MAP) is used instead, then the overtraining problem is reduced by the fact that only Gaussian terms that are close to new data are adapted (discussed later in this Section).

In MAP estimation the parameters, λ, are treated as random variables. For a set of i.i.d. feature vectors $\mathbf{x} = \{x_1, \ldots, x_T\}$, the MAP estimate of the the parameters is [6]

$$\lambda_{MAP}^{*} = \underset{\lambda}{\operatorname{argmax}}\, f(\lambda|\mathbf{x}) = \underset{\lambda}{\operatorname{argmax}}\, f(\mathbf{x}|\lambda)g(\lambda).$$

Since λ are random, they belong to a hyper-density with its own parameters. These hyper-parameters can be estimated (which is a difficult problem) or guessed (which is not realistic). To avoid this overparameterization problem, some constraints are assumed and a relevance factor is introduced. This is the approach adopted by Reynolds *et al.* [1] which they call Bayesian adaptation.

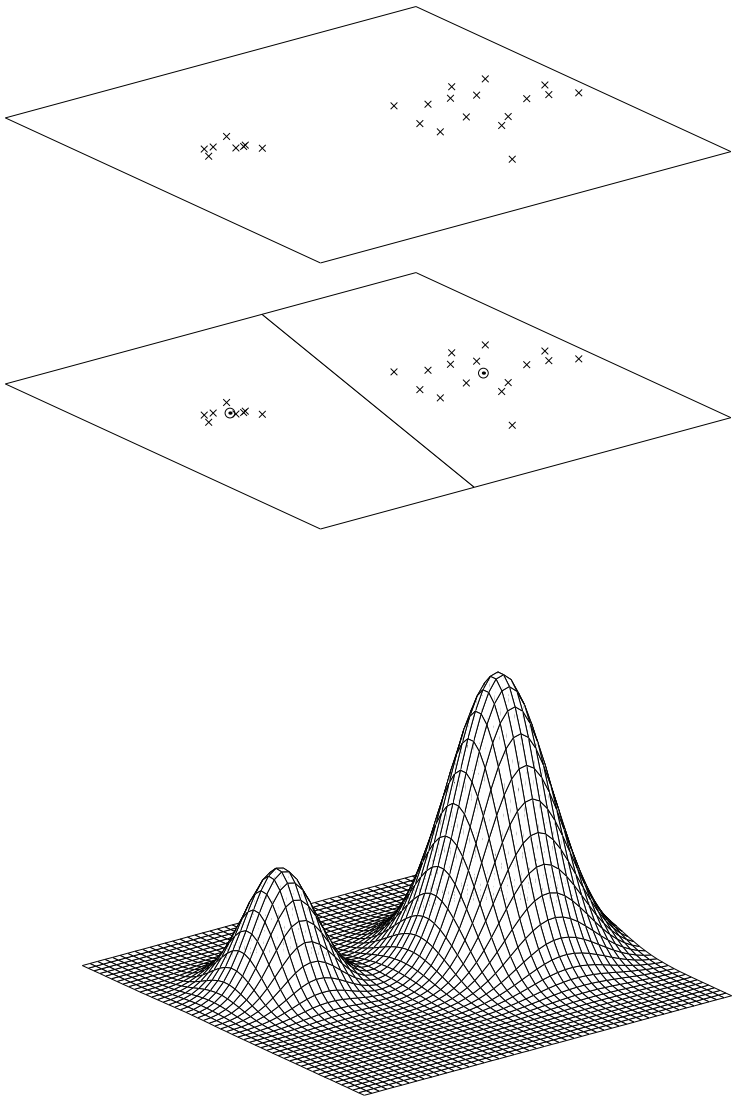

Figure 3.3: Principle of VQ-initialization for 2 dimensions and 2 clusters. The top Figure shows training data. The middle Figure shows VQ-clustering where $\odot$ are cluster centers. The bottom Figure shows a GMM estimated from VQ clusters. This initial GMM is then used as the starting point of the EM-algorithm.

With the same definitions used for ML-EM (3.7), (3.8) and

$$E_k(\mathbf{x}) = \frac{1}{c_k} \sum_{t=1}^{T} c_{kt} x_t \tag{3.13}$$

$$E_k(\mathbf{x}^2) = \frac{1}{c_k} \sum_{t=1}^{T} c_{kt} x_t^2 \tag{3.14}$$

the Bayesian adaptation equations (derived in Appendix B) are

$$\hat{w}_k = [\kappa_k^w c_k/T + (1-\kappa_k^w)w_k]\gamma \tag{3.15}$$
$$\hat{m}_k = \kappa_k^m E_k(\mathbf{x}) + (1-\kappa_k^m)m_k \tag{3.16}$$
$$\hat{\sigma}_k^2 = \kappa_k^v E_k(\mathbf{x}^2) + (1-\kappa_k^v)(\sigma_k^2 + m_k^2) - \hat{m}_k^2 \tag{3.17}$$

where

$$\kappa_k^\rho = \frac{c_k}{c_k + r^\rho} \tag{3.18}$$

for some parameter ρ and

$$\gamma = \frac{1}{\sum_{k=1}^K \hat{w}_k}. \tag{3.19}$$

The speaker model is adapted from the UBM selected by the gender detection component. The relevance factor, r^ρ, can be viewed as an adaptation coefficient. If r^ρ is large, adaptation is slow and if r^ρ is small, adaptation is fast. In this UBM-GMM system, a single relevance factor[1] is used, $r^w = r^m = r^v = 16$. It has been found experimentally that a single relevance factor $r = 16$ perform well for both diagonal covariances [1] and full covariances [25]. Note that equation (3.15) is not the true MAP estimate. It has been found experimentally [1] that the estimate (3.15) gives better performance than the true MAP estimate $\hat{w}_k = \frac{r^w + c_k}{Kr^w + T}$ (B.17). Also note that adaptation is not evaluated iteratively.

It is the data dependent adaptation coefficient κ^ρ that makes the adaptation efficient compared to ML-estimation. For terms with low probabilistic count, c_k, of new data, $\kappa^\rho \to 1$ and the emphasis in adaptation lies in speaker data. If new data doesn't match a specific Gaussian term then, $\kappa^\rho \to 0$ and the emphasis in the adapted term lies in the UBM data. Since terms with low probabilistic count are more likely to be overtrained, the adaptation approach should be robust to limited training data.

3.8 Fast Scoring

The fact that the client model is adapted from a UBM allows a fast scoring method [1]. This method is based on two observations. First, when a large GMM is evaluated only a few terms contribute significantly to the log-test ratio. This is because only a few terms of the GMM will be near the feature vector. Secondly, since the speaker model is adapted there is a correlation between terms in the UBM and in the speaker model. This means that a feature vector that is close to a term in the UBM is probably also close to the corresponding term in the speaker model.

With the discussion above in mind the "best" term, l_1, can be defined as

$$l_1 = \underset{k}{\operatorname{argmax}}\, w_k \mathcal{N}(x_t | m_k, r_k)$$

and the best C terms, $l_1, \ldots, l_C$, in a similar manner. Then a fast scoring method operates as follows: For each x_t compute the best C terms in the UBM and score these terms against the corresponding terms in the speaker model. This method requires only $K + C$ Gaussian computations compared to $2K$ Gaussian computations when the ordinary log-test ratio is used.

[1] In fact, the Bayesian adaptation equations are not true MAP estimates if $r^m \neq r^v$.

3.9 Score Normalization

The threshold η in equation (2.1) can be either speaker dependent or speaker independent. The purpose of speaker dependent thresholds is to reduce negative effects of speaker dependent variability on performance. Another solution is to adopt a reversible transform on score values so that the result is equivalent to using speaker dependent thresholds. For practical reasons the transform is based on impostor scores rather than the true speaker scores. One such method, currently known as *znorm* [17], is to transform the impostor score distribution to zero mean and unit variance, whereas a Gaussian distribution is assumed. For an observation $\mathbf{x}$ and a claimed identity λ, the normalized log-test is given by

$$\Lambda_\lambda^{znorm}(\mathbf{x}) = \frac{\Lambda_\lambda(\mathbf{x}) - \mu_\lambda}{\sigma_\lambda} \tag{3.20}$$

where μ_λ and σ_λ are moment estimates of the impostor score distribution for a speaker λ. In GIVES, a znorm module is available.

If knowledge of different handsets is incorporated, znorm can be extended by using one transformation per handset. This is called handset score normalization or *hnorm* [1].

Chapter 4

Experiment Setup

4.1 Evaluation Strategy

A speaker verification system has a lot of parameters to adjust. Also, evaluation is a slow process since experiments require large speech databases to achieve reliable results. Since time is limited in this thesis project, only a few parameters that affect performance were examined, namely:

- GMM order, i.e. the number of terms in the GMM
- The effect of adapting different sets of GMM-parameters (weights, means and variances)
- The effect of znorm
- The presence of goats, wolves and lambs

It is quite clear the number of terms in the GMM will affect performance. The choice of investigating different sets of GMM-parameters is motivated by the following:

The constraints in the derivation (in appendix B) of the Bayesian adaptation equations indicate that adaptation of some parameters is not optimal. For example, it may be possible that the best performance is achieved by only updating the means. Therefore, adaptation of all possible combinations of weights, means and variances are tested. The use of a global relevance factor is also a strong constraint that could affect some adaptation combinations in some unknown way.

In the signal processing component a simple handset normalization method, cepstral mean subtraction (CMS), was applied. As mentioned in Section 3.3, CMS doesn't compensate for mismatched handset situations completely. Therefore, the effect of znorm is investigated.

In a speaker verification system it is desirable that there are no goats, wolves and lambs i.e. the errors are homogeneously distributed in the speaker population. In practice, the presence of these "animals" are real which motivate an "animal" analysis.

One of the goals of this thesis project was to compare performance to another system. An available system is a digit-based TD system implemented by Melin

[18] but the system used for comparison in this thesis report uses dynamic cohort rather than the cohort based system used in the referred paper. In short, the TD system has one left-to-right HMM for each digit and almost the same preprocessing was used as in the UBM-GMM system. This means that experiments with digits must also be carried out.

In this evaluation, the fast scoring method was not used (see Section 6.2 for motivation).

4.2 Training the UBMs

A UBM must cover all possible speaker variabilities. This means that the database has to be well balanced, i.e. the training data has the same set of sub-populations as the population that is intended to use the system. A database that has these properties is the full Swedish SpeechDat database, FDB5000 [13], which comprises 5000 speakers recorded over the fixed telephone network. Unfortunately, training UBMs on this database would take too much time in this thesis project. Therefore, the smaller FDB1000[1], which is an initial version of FDB5000, was used. The FDB1000 contains 1000 speakers but it is not carefully balanced which may affect performance negatively.

The FDB1000 is composed of many kinds of speech data (Table 4.1). The sentences and words for corpus identifier S and W are different for all speakers. These sub-corpora are used for training UBMs with two different types of speech. First, a subset containing S1-S3 and W1-W2 is labeled "various speech" and a subset containing B1 and C1-C4 is labeled "digit speech". Secondly, both subsets were split into female and male speakers. Finally, each subset was used to train GMMs with 128, 256, 512 and 1024 terms. Various handsets were used during speech recording of the corpora. The amount of speech (with removed silence) used to train each UBM is listed in Table 4.2. The segmentation for the digit UBMs is produced by a speech recognizer working in forced alignment mode given the text actually spoken by the subject [19]. The segmentation for the various speech UBMs is produced by a silence/speech detector.

For initialization (see Section 3.6.2), the minimum cluster size U is set to 5, a Mahanalobis distance function is used, ϵ_{avg} is set to 0.0005 and a maximum of 8 VQ iterations are allowed. With the convergence properties of EM, discussed in Section 3.6.1 in mind, ϵ_{EM} is set to a very small number, 0.01, and a total of 8 iterations are allowed. Generally, all 8 EM iterations were proceeded.

4.3 Enrollment and Testing

For enrollment and testing the GANDALF database [8] was used. GANDALF is a speaker verification database that covers both long-term variations in speaker variability and telephone handset variations. Two subsets are used for enrollment and testing. The first subset is labeled D1H/D4 and contains digits. The second is labeled V1H/V and contain various sentences. These subsets were extracted from the evaluation set [18] in GANDALF. There is also a smaller development set which was used for early experiments and for tuning parameters.

[1] Described in the manual: "FIXED1SV - FDB1000, A 1000 Speaker Swedish Database for the Fixed Telephone Network", Design.doc, v2.1

Table 4.1: FDB1000 corpora used for UBMs.

Corpora identifier	Item identifier	Corpora content
B	1	1 sequence of 10 isolated digits
C	1	1 sheet number (5+ digits)
C	2	1 telephone number (9-11 digits)
C	3	1 credit-card number (16 digits)
C	4	1 PIN code (6 digits) (set of 150 SDB codes)
S	1-9	9 phonetically rich sentences
W	1-4	4 phonetically rich words

Table 4.2: Speech duration for the UBM training material excluding silence.

Speech content	digits		various	
Gender	F	M	F	M
Speech duration	2.6h	1.7h	2.2h	1.6h

The subsets used for evaluation are summarized in Tables 4.3 and 4.4. During the test phase, an average of 21 true speaker tests per speaker are evaluated. In addition to the available impostors, true speakers are also used as simulated impostors. Every impostor is scored against each true speaker and, of course, a true speaker is not used as an impostor to oneself. Only same-sex impostor test are used since this is more likely in an operational system. If cross-sex impostors are included, the EERs drops by 1-2 percent overall. This shows that the system can handle cross-sex impostors quite good, but as mentioned above, only same-sex impostor results are presented.

The segmentation during enrollment and test is the same that is used for the UBM training data, except that the digit segmentation uses forced alignment mode given the expected (not the actual) text of the utterance.

Table 4.3: Enrollment subsets. Speech duration excludes silence.

Subset labels	D1H	V1H
Handsets/speaker	1	
Utterance content	5 digits	1 sentence
Utterances/speaker	25	10
Speech duration/speaker (s)	50	30
Clients (male/female)	24/18	

Table 4.4: Test subsets. Speech duration excludes silence.

Subset labels	D4	V
Handsets/speaker	4-10	
Utterance content	2x4 digits	2 sentences
Speech duration/utterance (s)	3	6
Client speaker tests	886	
Impostors (male/female)	58/32	
Same-Sex impostor tests	1926	

4.4 Statistical Significance

It is obvious that a large test gives FA/FR-rates closer to the true values than a small test. Assuming independent trials and a binomial distribution for an error-rate p, a confidence interval $p \pm e \cdot p$, gives a lower bound of

$$p = \frac{1}{n_{total}(e/\lambda_{\alpha/2})^2 + 1} \tag{4.1}$$

where n_{total} is the total number of trials and $\lambda_{\alpha/2}$ is, for example, 1.65 for a 90 percent confidence interval [27].

Assuming that $e = 0.3$, then 1926 impostor tests gives a FA-rate interval of $1.6 \pm 0.5\%$ and 886 true speaker tests gives a FR-rate interval of $3.3 \pm 1.0\%$, both with 90 percent certainty. This formula assumes that each speaker can expect the same error rate, which is not exactly the case, but this gives a clue to how relevant the experiment results are. This exercise is quite academic since a confidence interval for a known p is a different thing. The assumption that each speaker can expect the same error rate is fairly optimistic and, therefore, no confidence intervals are computed for the experiment results.

Chapter 5

Experiment Results

In this chapter the experiment results are presented and basic observations are made. Initially, znorm was meant to be used with both subsets D1H/D4 and V1H/V, but a software bug stopped this experiment for the D1H/D4 subset.

5.1 Model Mixture Order

Client GMMs with 128, 256, 512 and 1024 terms were evaluated for D1H/D4 and V1H/V. All GMM parameters were updated and znorm was not applied. The results are shown as DET-plots in Figures 5.1 and 5.2. EER values are shown in Table 5.1. As expected, the text dependent experiment with digits performed better than the text independent experiment with various sentences. Note that the EERs are higher in the 1024 term case compared to the 512 term case.

It may be possible that the 1024 terms UBMs have not really converged during training since slower convergence could be expected if mixture elements are poorly separated. In order to test this hypothesis another 4 iterations were performed on the 1024 terms digit UBMs. Then the EER dropped to 4.4 for D1H/D4, which supports the hypothesis. However, performance is still almost the same as for the 512 term GMMs, and the evaluation of 1024 term UBMs is very slow, so subsequent tests were limited to the 512 term UBMs. Unfortunately, the log-test values from UBM training were never saved so a true convergence analysis could not be carried out.

Table 5.1: EER (%) for different GMM orders. All GMM parameters are updated and znorm is not used.

Terms	128	256	512	1024
D1H/D4	5.6	4.7	4.5	4.6
V1H/V	7.7	7.1	7.0	8.0

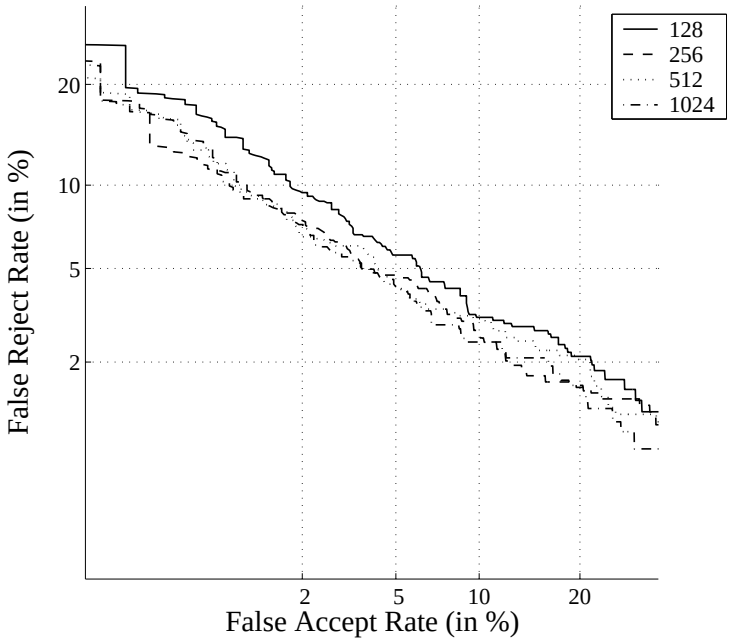

Figure 5.1: DET plots for D1H/D4 with different GMM orders.

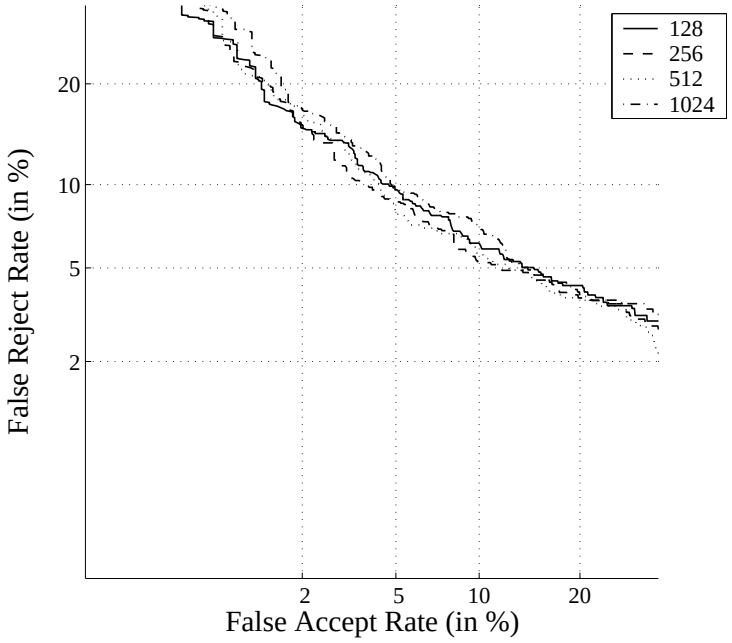

Figure 5.2: DET plots for V1H/V with different GMM orders.

5.2 Parameter Update

In this experiment, all possible combinations of updating weights, means and variances were tested for D1H/D4 and V1H/V with 128 and 512 GMM terms. The results are presented in Figures 5.3, 5.4 and in Table 5.2.

The EERs from these tests show that the best results with 128 GMM terms were achieved by updating the means and variances, while for 512 GMM terms, the best results were achieved by updating all GMM parameters. Note that if only variances are updated, the 512 term GMMs yield higher EER compared to the 128 term GMMs.

Table 5.2: EER (%) for all combinations of GMM parameters (w = weights, m = means and v = variances). The best result for each subset is printed in bold style.

128 Terms							
Subset	w	m	v	wm	wv	mv	wmv
D1H/D4	17.9	6.2	45.4	5.7	43.8	**5.5**	5.6
V1H/V	27.9	8.4	42.9	8.7	41.5	**7.3**	7.7
512 Terms							
Subset	w	m	v	wm	wv	mv	wmv
D1H/D4	12.5	5.0	58.4	4.8	55.9	4.8	**4.5**
V1H/V	21.8	8.5	48.7	8.2	45.7	7.6	**7.0**

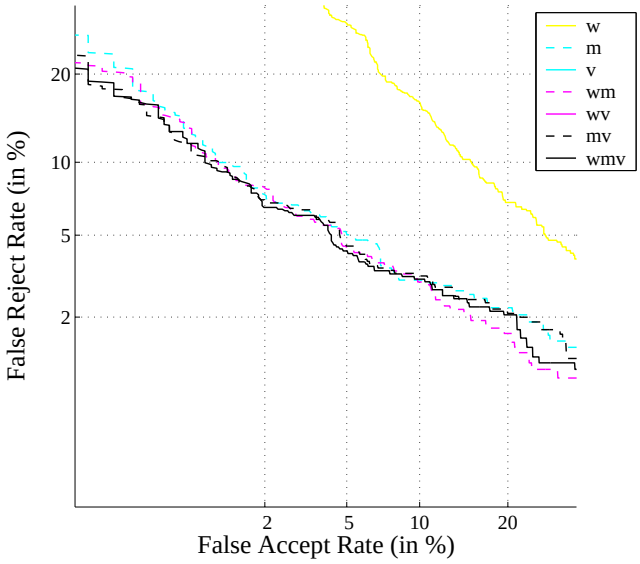

Figure 5.3: DET plots for subset D1H/D4 with all combinations of parameters updated for 512 GMM terms, (w = weights, m = means and v = variances). Some curves are outside the graph.

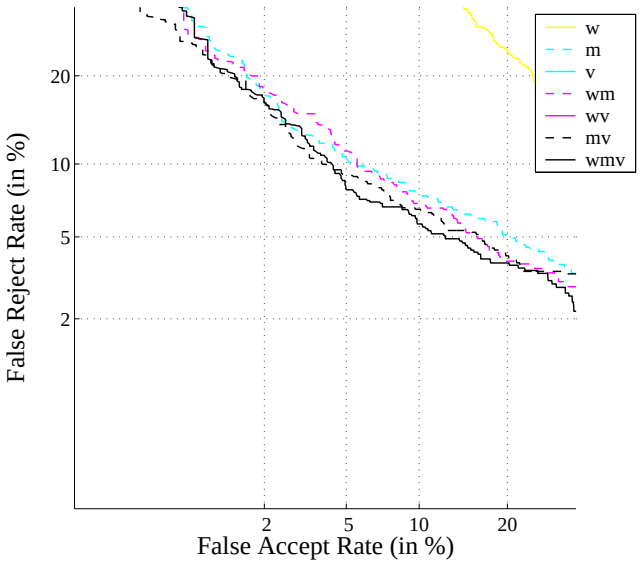

Figure 5.4: DET plots for subset V1H/V with all combinations of parameters updated for 512 GMM terms, (w = weights, m = means and v = variances). Some curves are outside the graph.

5.3 Score Normalization

In this experiment znorm was applied to the V1H subset with 512 GMM terms and all GMM parameters were updated. Znorm was trained with 391 pseudo impostors with a 50/50 gender ratio from the S1 corpora in FDB1000 scored against the gender UBM detected for the client speaker. Since there are no cross-sex trials and the pseudo impostor set contains both sexes, 50 percent of the best scores was used for score normalization. This will correspond to using prior knowledge of a pseudo impostor gender. The result is plotted in Figure 5.5 and show that there is no advantage of using *znorm* in this experiment.

5.4 Performance Comparison

In this experiment the text independent GMM system performance is compared to the text dependent HMM-based system. The same enrollment and test set, D1H/D4, was used for both systems. In the GMM system 512 GMM terms were used and all GMM parameters were updated. The HMM system generated an EER of 4.8 compared to 4.5 for the GMM system. The result is shown as a DET-plot in Figure 5.6.

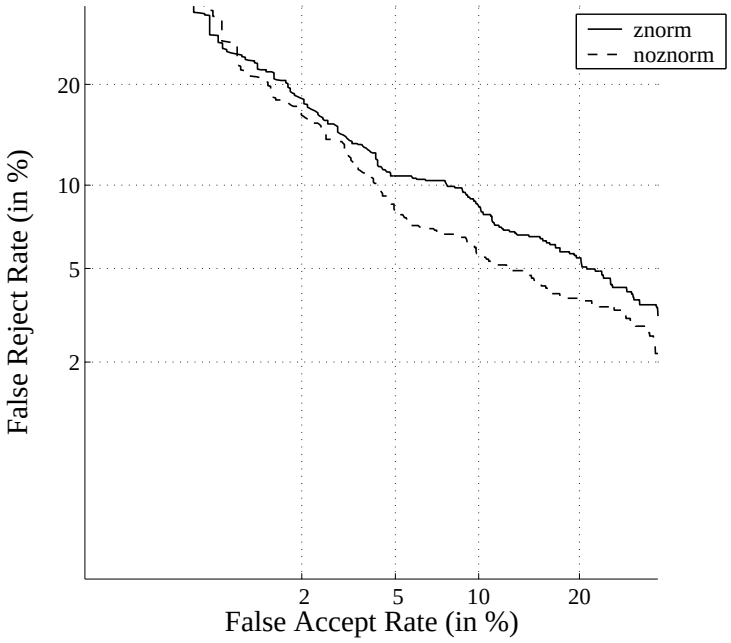

Figure 5.5: DET plots with and without znorm for subset V1H/V with all combinations of parameters updated for 512 GMM terms.

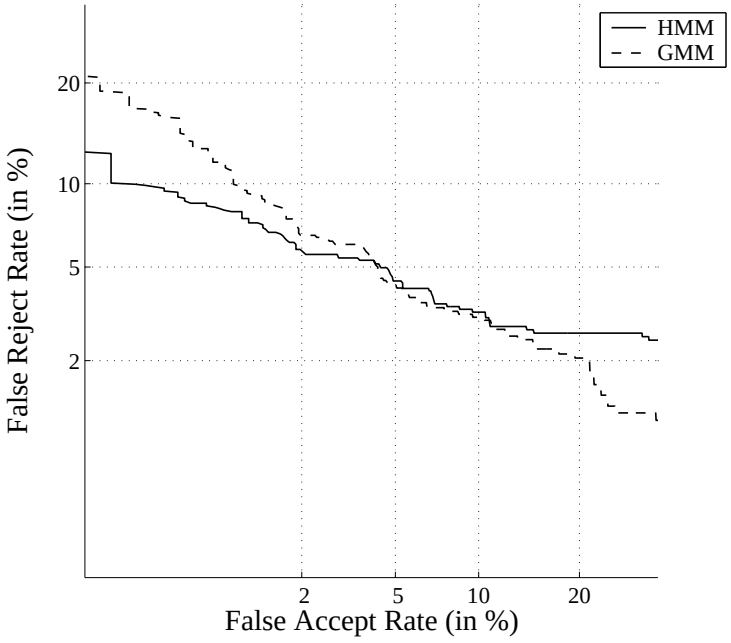

Figure 5.6: DET plots for the TI GMM system and for the TD HMM system.

5.5 Goats, Wolves and Lambs

In Section 2.6 it was mentioned that "goats", "wolves" and "lambs", while in minority, tend to contribute to a majority of all errors. In this thesis report the goats, wolves and lambs are defined as those clients/impostors who contribute to 50 percent of all errors of the respective types. With this definition, the largest possible value is 50 percent for goats, wolves and lambs respectively, in which case no true speaker will expect any disadvantage and no impostor will expect any advantage. The presence of these "animals" were examined for both the HMM system and the GMM system by using the threshold that correspond to a EER. For the GMM system 512 GMM terms was used and all Gaussian parameters were updated. The result is shown in Table 5.3. A more detailed analysis shows that there are 3 goats in the GMM system and 1 goat in the HMM system. The "worst" goat is the same for both systems. Roughly half of the male lambs and wolves are the same in both systems.

Table 5.3: Animals for different systems.

System	Subset	Goats (%)	Wolves (%)	Lambs (%)
HMM	D1H/D4	2	14	18
GMM	D1H/D4	7	19	23
GMM	V1H/V	11	17	24

Chapter 6

Discussion and Conclusions

6.1 Evaluation Results

The most obvious result was that compromising on text dependence by using digits rather than sentences greatly improves performance. This result was also expected (discussed in Section 2.3). It was also found that higher order GMMs seems to perform better than lower order GMMs if convergence is ensured. Moreover, the best results were achieved by updating only means and variances for 128 GMM terms. For 512 GMM terms, the best result was achieved by updating all GMM parameters. This result holds for both the D1H/D4 subset and the V1H/V subset. It seems that means contain most of the information of the enrolled speaker's voice and the combination of means and variances contain even more information. This conclusion is different from Reynolds *et al.* [1] who found that the best performance was achieved by only updating the means. The explanation for this difference is unclear. The use of znorm didn't improve performance for the V1H/V subset. The explanation can be that *znorm* doesn't necessarily improve results if short test segments are used [17]. It was found that a small subpopulation termed "goats", "wolves" and "lambs" contributed to a majority of all errors. The error counts of these subpopulations match the result of Doddington *et al.* [23]. It was also found that there is no benefit from combining the text dependent HMM system and the text independent GMM system since there was a correlation between the "animals" in both systems. An interesting result was that the text independent GMM system and the text dependent HMM system had almost equal performance. The DET-plot (Figure 5.6) shows that the GMM system is slightly better if a low FR-rate is preferred, but on the other hand, the HMM-system is better if the priority is low FA-rate. Overall, the results are statistically significant but with some degree of uncertainty due to the limited size of test data.

If larger UBMs are used, that are well balanced with guaranteed convergence, even better performance may be expected.

6.2 Goals

The main goal of this project was to implement a text independent speaker verification module for GIVES using adapted GMM. This was accomplished and

the system generally performed well compared to a text dependent HMM-based system. However, the fast scoring component didn't work together with some other components of GIVES. Furthermore, the evaluation time was underestimated so the project was delayed with 4 weeks.

6.3 Improvements

Although the number of possible speaker verification techniques are numerous and various methods are suitable for different tasks, some improvements are suggested:

- More advanced channel normalization methods
- Detecting goats and lambs during enrollment [24] and take an appropriate action
- Better initialization methods
- Alternatives to the EM-algorithm [28]
- Study various score normalization methods
- More advanced variance flooring [18]
- Introduce parameter, mixture and/or speaker dependent relevance factors
- Take advantage of higher order language information (a quite difficult task)

It is unclear if all mentioned improvements actually give better performance, but that is a question for future research.

Bibliography

[1] Reynold D. A., Quatieri T. F., Dunn R. B., (2000), "Speaker Verification Using Adapted Gaussian Mixture Models", *Digital Signal Processing*, vol. 10, pp. 19-41.

[2] Blimes J. A., (1998), "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models", International Computer Science Institute, Berkeley, California, U.S.A., Technical Report: TR-97-021, April.

[3] Dempster A.P., Laird N.M., Rubin D.B, (1977), "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society*, Series B (Methodological), vol 39, no. 1, pp. 1-38.

[4] Lee C.-H., Gauvain J.-L., (1996), "Bayesian adaptive learning and MAP estimation of HMM", In Lee C.-H., Soong F.K., and Paliwal K.K., editors, *Automatic Speech and Speaker Recognition - Advanced Topics*, Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 83-107.

[5] Melin H., (1996), "Speaker Verification in Telecommunication", Department of Speech, Music and Hearing, KTH, Available from: http://www.speech.kth.se/~melin/publications.html.

[6] Gauvain J.-L., Lee C.-H., (1994), "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains", *IEEE Trans. Speech Audio Process*, vol. 2, pp. 291-298.

[7] Lee, C.-H., Gauvain, J.-L., (1991), "Bayesian learning of gaussian mixture densities for hidden Markov models", *Proc. DARPA Speech natural language Workshop* (Pacific Grove), California, U.S.A., Feb. 19-22, pp. 272-277.

[8] Melin H., (1996), "The Gandalf speaker verification database", Fonetik-96, TMH-QPSR 2/1996, Department of Speech, Music and Hearing, KTH, Stockholm, pp. 117-120, Available from: http://www.speech.kth.se/~melin/publications.html.

[9] Doddington G. R., (1998), "Speaker Recognition Evaluation Methodology - An Overview and Perspective", *Proceedings of Speaker Recognition and its Commercial and Forensic Applications* (RLA2C), Avignon, France, April 20-23, pp. 60-66.

[10] Huo Q., Chan C., Lee C., (1995), "Bayesian Adaptive Learning of the Parameters of Hidden Markov Model for Speech Recognition", *IEEE Trans. on Speech and Audio Processing*, vol. 3, no. 5, pp. 334-345, Available from: http://citeseer.nj.nec.com/huo95bayesian.html

[11] Xu L., Jordan M.I., (1996), "On Convergence Properties of the EM Algorithm for Gaussian Mixtures", *Neural Computation*, vol. 8, pp. 129-151, Available from: http://citeseer.nj.nec.com/xu95convergence.html.

[12] McKenzie P., Alder M., (1994), "Initializing the EM algorithm for use in Gaussian mixture modeling", Technical Report: TR93-14, The University of Western Australia, Center for Intelligent Information Processing Systems, Crawley, Australia, Available from: http://ciips.ee.uwa.edu.au/Papers/

[13] Elenius, K., (2000), "Experiences from collecting two Swedish telephone speech databases", International Journal of Speech Technology, vol. 3, pp. 119-127.

[14] Campbell, J.P., (1997), "Speaker Recognition: A Tutorial", *Proceedings of IEEE* vol. 85, no 9., pp. 1437-1462.

[15] Furui, S. (1997), "Recent Advances in Speaker Recognition", In Springer, editor, *Audio- and Video-based Biometric Person Authentication*, Crans-Montana, Switzerland, March 12-14, pp. 237-251.

[16] Martin A., Doddington K. G., Ordowski M., Przybocki M., (1997), "The DET curve in assessment of detection task performance", *Proceedings of EuroSpeech '97*, Rhodes, Greece, 22-35 September, vol. 4, pp. 1895–1898.

[17] Gravier G., Kharroubi J., Chollet G., (2000), "On the Use of Prior Knowledge in Normalization Schemes for Speaker Verification", *Digital Signal Processing*, vol. 10, no. 1/2/3, Jan., pp. 213-225.

[18] Melin H., Lindberg J., (1999), "Variance Flooring, Scaling and Tying for Text Dependent Speaker Verification", *Proc. 6th European Conference on Speech Communication and Technology (EUROSPEECH)*, Budapest, Hungary, September 5-9, pp 1975-1978.

[19] Melin, H., (1998), "On Word Boundary Detection in Digit-Based Speaker Verification", Workshop on Speaker Recognition and its Commercial and Forensic Applications (RLA2C), Avignon, France, April 20-23, pp. 46-49.

[20] Gibbon D., Moore R., Winski R., (1997), editors, "Handbook of standards and resources for spoken language systems", Walter de Gruyter, ISBN 3-11-015366-1

[21] Gersho A., Gray M. R., (1991), "Vector Quantization and Signal Processing", Kluwer Academic Publishers, ISBN 0-7923-9181-0.

[22] Veldhuizen T., (2001), "Blitz++ User's Guide", version 1.2, Available from: http://oonumerics.org/blitz/ (November 29, 2001).

[23] Doddington G., et al., (1998), "Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation", *International Conference on Spoken Language Processing*, Sydney, Australia, 30 Nov. - 4 Dec., vol. 4, pp. 1351-1354

[24] Thompson J., Mason J. S., (1994), "The pre-detection of error-prone class members at the enrollment stage of speaker recognition systems", *Proc. ESCA workshop on automatic speaker recognition, identification and verification*, Martigny, Switzerland, April 5-7, pp. 127-130.

[25] Vuuren S., (1999), "Speaker Verification in a Time-Feature Space", Ph.D. thesis, Oregon Graduate Institute, March.

[26] Reynolds D., Rose R., (1995), "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models", *IEEE Transactions on Speech and Audio Processing*, Vol. 3, No. 1, pp. 72-83.

[27] Råde L., Westergren B., (1991), "Mathematics Handbook for Science and Engineering", Studentlitteratur, Fourth edition, ISBN 91-44-00839-2.

[28] Helmbold D., Schapire R., Singer Y., Warmuth M., (1995), "A comparison of new and old algorithms for a mixture estimation problem", *Journal of Machine Learning*, vol. 27, no. 1, pp. 97-119, Available from: http://citeseer.nj.nec.com/helmbold97comparison.html.

[29] Westphal M., (1997), "The Use of Cepstral Means in Conversational Speech Recognition", *Proceedings of Eurospeech Conference*, Rhodes, Greece, 22-35 September, pp. 1143-1146, Available from: http://citeseer.nj.nec.com/104309.html

Appendix A

Numerical Properties

When implementing various algorithms, often numerical problems arises. In the case of the GMM system, division by zero sometimes occur when c_{kt} is computed and sometimes the log-test value gives minus infinity as result. The origin of this is the exponential term in $\mathcal{N}(x_t|m_k, r_k)$. If $\exp(-x^2)$ is evaluated by a computer, then the result can be zero if x is large enough but this can't be allowed. The author solved this problem by a trick called "Gaussian vector normalization" which principle is shown in Figure A.1. So, $\acute{\mathcal{N}}(x_t|m_k, r_k)$ is computed instead:

$$\acute{\mathcal{N}}(x_t|m_k, r_k) = \frac{1}{(2\pi)^{D/2}|r_k^{-1}|^{1/2}} \exp\left(-\frac{1}{2}(x_t - m_k)' r_k (x_t - m_k) - \beta_t\right) \quad \text{(A.1)}$$

where

$$\beta_t = \max_k \left(-\frac{1}{2}(x_t - m_k)' r_k (x_t - m_k)\right) \quad \text{(A.2)}$$

Then the log-test ratio is

$$\log(f(\mathbf{x}|\lambda)) = \sum_{t=1}^{T} \left(\log\left(\sum_{k=1}^{K} w_k \acute{\mathcal{N}}(x_t|m_k, r_k)\right) + \beta_t\right) \quad \text{(A.3)}$$

Now the vector $-\frac{1}{2}(x_t - m_k)' r_k (x_t - m_k)$ can take any possible value allowed by the computer hardware without any numerical problems if r_k^{-1} is positive definite. Using $\acute{\mathcal{N}}$ for c_{kt}, calculation is straight forward

$$c_{kt} = \frac{w_k \acute{\mathcal{N}}(x_t|\theta_k)}{\sum_{l=1}^{K} w_l \acute{\mathcal{N}}(x_t|\theta_l)} \quad \text{(A.4)}$$

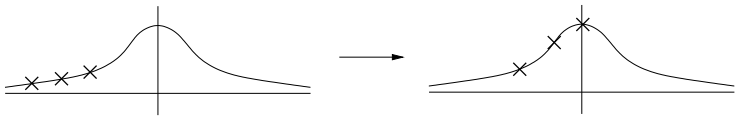

Figure A.1: Principle of Gaussian vector normalization for a 3 dimensional vector.

where β_t is computed for the denominator. This is not optimal because the numerator could be zero. It will ensure that the denominator is larger then zero (if all $w_k > 0$) and this is the most important issue. However, the exponential term in the numerator could overrun the computers internal representation if $-\beta_t$ is very large. This can happen if a very small training set is used on a large number of mixtures, but it is not likely.

Note that the numerical values of c_{kt} and the modified log-test won't differ if $\acute{\mathcal{N}}(x_t|m_k, r_k)$ is used instead of $\mathcal{N}(x_t|m_k, r_k)$. Furthermore, variance flooring was applied to ensure that r_k^{-1} is always positive definite. A simple flat floor is used which means that the elements of r_k^{-1} are not allowed to be smaller than a preassigned value ϵ_r. In this system ϵ_r is set to 0.001.

Appendix B

Maximum A Posteriori Estimates for Gaussian Mixture Models

B.1 An overview of MAP Estimates for Gaussian Mixture Models

The most important ideas of MAP Estimates for GMM, presented by Gauvain and Lee [4, 6, 7], summarized and extended by Hou, Chan and Lee [10], are described here.

Remember the GMM joint p.d.f.

$$f(\mathbf{x}|\lambda) = \prod_{t=1}^{T} \sum_{k=1}^{K} w_k \mathcal{N}(x_t|m_k, r_k) \tag{B.1}$$

$$\lambda = (w_1, \ldots, w_K, \theta_1, \ldots, \theta_K) \quad \theta = (m_1, \ldots, m_K, r_1, \ldots, r_K)$$

where

$$\mathcal{N}(x|m_k, r_k) \propto |r_k|^{1/2} \exp\left[-\frac{1}{2}(x - m_k)^t r_k (x - m_k)\right] \tag{B.2}$$

The MAP estimate is defined as

$$\lambda^*_{MAP} = \underset{\lambda}{\operatorname{argmax}}\, f(\lambda|\mathbf{x}) = \underset{\lambda}{\operatorname{argmax}}\, f(\mathbf{x}|\lambda) g(\lambda)$$

where $g(\lambda)$ is the prior p.d.f.

Finding $g(.)$ is not a trivial problem, mostly due to that the dimension of $\mathbf{x}$ is fixed and therefore sufficient data for estimation of $g(\lambda)$ is not available for GMMs. However, if $g(\lambda)$ is chosen carefully, then it can be shown [6] that the EM algorithm can be applied and incomplete data is no longer a problem.

The GMM weights could be modeled as a Dirichlet density

$$g(w_1, \ldots, w_K, \nu_1, \ldots, \nu_K) \propto \prod_{k=1}^{K} w_k^{\nu_k - 1} \tag{B.3}$$

where $\nu_k > 0$ are the density parameters.

If full covariance matrices are assumed, then the Gaussian parameters (m_k, r_k) are modeled as a normal-Wishart density

$$g(m_k, r_k|\tau_k, \mu_k, \alpha_k, u_k) \propto$$
$$|r_k|^{(\alpha_k - p)/2} \exp\left[-\frac{\tau_k}{2}(m_k - \mu_k)^t r_k (m_k - \mu_k)\right] \exp\left[-\frac{1}{2} tr(u_k r_k)\right] \tag{B.4}$$

where $(\tau_k, \mu_k, \alpha_k, u_k)$ are the prior density parameters such that $\alpha_k > p-1, \tau_k > 0, \mu_k$ is a vector of dimension p, and u_k is a $p \times p$ positive definite matrix.

In the diagonal covariance case, a normal-gamma density is assumed:

$$g(m_k, r_k|\tau_{kd}, \mu_{kd}, \alpha_{kd}, \beta_{kd}) \propto$$
$$\prod_{d=1}^{D} r_{kd}^{\alpha_{kd} - 1/2} \exp\left[-\frac{1}{2}\tau_{kd} r_{kd} (m_{kd} - \mu_{kd})^2\right] \exp\left[-\beta_{kd} r_{kd}\right] \tag{B.5}$$

where $\tau_{kd}, \alpha_{kd}, \beta_{kd} > 0, d = 1, \ldots D$. Note that a normal-gamma density is just a one-dimensional case of a normal-Wishart density.

Assuming independence between the parameters of the individual mixture components and the set of mixture weights, the joint prior density is

$$g(\lambda) = g(w_1, \ldots, w_k) \prod_{k=1}^{K} g(m_k, r_k) \tag{B.6}$$

Now the EM algorithm can be applied to MAP estimation. Define:

$$c_{kt} = \frac{w_k \mathcal{N}(x_t|\theta_k)}{\sum_{l=1}^{K} w_l \mathcal{N}(x_t|\theta_l)} \tag{B.7}$$

$$c_k = \sum_{t=1}^{T} c_{kt} \tag{B.8}$$

$$E_k(\mathbf{x}) = \frac{1}{c_k} \sum_{t=1}^{T} c_{kt} x_t \tag{B.9}$$

Now the EM reestimation formulæs (for full covariances) are:

$$\hat{w}_k = \frac{(\nu_k - 1) + c_k}{\sum_{k=1}^{K}(\nu_k - 1) + T} \tag{B.10}$$

$$\hat{m}_k = \frac{\tau_k \mu_k + c_k E_k(\mathbf{x})}{\tau_k + c_k} \tag{B.11}$$

$$\hat{r}_k^{-1} = \frac{u_k + \sum_{t=1}^{T} c_{kt}(x_t - \hat{m}_k)(x_t - \hat{m}_k)^t + \tau_k(\mu_k - \hat{m}_k)(\mu_k - \hat{m}_k)^t}{(\alpha_k - p) + c_k} \tag{B.12}$$

B.2 Bayesian Adaptation

In this Section the Bayesian adaptation equations presented by Reynolds *et al.* [1] are derived.

The initial estimate may be chosen as the mode of the prior density [10]

$$m_k = \mu_k \tag{B.13}$$
$$r_k = (\alpha_k - p)u_k^{-1} \tag{B.14}$$

However, there is still a huge number of parameters that cannot be estimated but just guessed. Therefore, some assumptions must be made to avoid over-parametrization. If no prior information is available, it is possible to show [6] that the following constraints on the prior parameters hold

$$\nu_k = \tau_k + 1 \tag{B.15}$$
$$\alpha_k = \tau_k + p \tag{B.16}$$

These constraints leaves τ_k left. Now let $\tau_k = r^\rho$ for some parameter ρ. Equation (B.10) and (B.15) gives the MAP estimate:

$$\hat{w}_k = \frac{r^w + c_k}{Kr^w + T} \tag{B.17}$$

Define:

$$\kappa_k^\rho = \frac{c_k}{c_k + r^\rho} \tag{B.18}$$

Then equation (B.11) and (B.13) gives a MAP estimate:

$$\begin{aligned}\hat{m}_k =& \frac{\tau_k m_k + c_k E_k(\mathbf{x})}{\tau_k + c_k} = \left(1 - \frac{c_k}{\tau_k + c_k}\right) m_k + \frac{c_k}{\tau_k + c_k} E_k(\mathbf{x}) \\ =& \kappa_k^m E_k(\mathbf{x}) + (1 - \kappa_k^m) m_k \end{aligned} \tag{B.19}$$

Observe that

$$\kappa_k^m E_k(\mathbf{x}) = \hat{m}_k - (1 - \kappa_k^m) m_k \tag{B.20}$$

and define

$$E_k(\mathbf{x}^2) = \frac{1}{c_k} \sum_{t=1}^{T} c_{kt} x_t^2 \tag{B.21}$$

If diagonal covariances are assumed, then equation (B.12), (B.14), (B.16) and (B.18) gives

$$\begin{aligned}\hat{r}_k^{-1} =& \frac{\tau_k r_k^{-1} + \sum_{t=1}^{T} c_{kt}(x_t - \hat{m}_k)(x_t - \hat{m}_k)^t + \tau_k (m_k - \hat{m}_k)(m_k - \hat{m}_k)^t}{\tau_k + c_k} \\ =& \frac{c_k}{\tau_k + c_k} \frac{\sum_{t=1}^{T} c_{kt}(x_t - \hat{m}_k)^2}{c_k} + \left(1 - \frac{c_k}{\tau_k + c_k}\right) \left(\sigma_k^2 + (m_k - \hat{m}_k)^2\right) \\ =& \kappa_k^v (E_k(\mathbf{x}^2) + \hat{m}_k^2 - 2E_k(\mathbf{x})\hat{m}_k) + \\ & (1 - \kappa_k^v)(\sigma_k^2 + m_k^2 + \hat{m}_k^2 - 2m_k \hat{m}_k) \\ =& \kappa_k^v E_k(\mathbf{x}^2) + (1 - \kappa_k^v)(\sigma_k^2 + m_k^2) + \hat{m}_k^2 - \\ & - 2(\kappa_k^v E_k(\mathbf{x})\hat{m}_k + (1 - \kappa_k^v) m_k \hat{m}_k) \qquad \text{(B.20) and } \kappa_k^m = \kappa_k^v \Rightarrow \\ =& \kappa_k^v E_k(\mathbf{x}^2) + (1 - \kappa_k^v)(\sigma_k^2 + m_k^2) + \hat{m}_k^2 \\ & - 2(\hat{m}_k(\hat{m}_k - (1 - \kappa_k^m) m_k) + (1 - \kappa_k^v) m_k \hat{m}_k) \\ =& \kappa_k^v E_k(\mathbf{x}^2) + (1 - \kappa_k^v)(\sigma_k^2 + m_k^2) - \hat{m}_k^2 \end{aligned} \tag{B.22}$$

Note that the constraints (B.15) and (B.16) doesn't affect equation (B.11).

Appendix C

A List of Abbreviations

ANN Artificial Neural Networks
CMS Cepstral Mean Subtraction
DET plot Detection Error Trade-off plot
EER Equal Error Rate
EM Expectation Maximization
FA False Acceptance
FR False Rejection
GIVES General Identity VErification System
GMM Gaussian Mixture Models
HMM Hidden Markov Models
MAP Maximum A Posteriori
MFCC Mel-Frequency Cepstral Coefficients
ML Maximum Likelihood
TD Text Dependent
TI Text Independent
UBM Universal Background Model
VQ Vector Quantization