IDIAP RESEARCH REPORT

Joint Speech and Speaker Recognition

Mohamed Faouzi BenZeghiba[a]

IDIAP RR 05-28

February 2005

TO APPEAR AS

BenZeghiba, M. F. (2005). *Joint speech and speaker recognition.* Docteur ès Sciences thesis, Swiss Federal Institute of Technology Lausanne (EPFL), Lausanne, Switzerland.

Dalle Molle Institute for Perceptual Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 – 27 – 721 77 11
fax +41 – 27 – 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

[a] with the Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP), CH-1920 Martigny, Switzerand, and with the Swiss Federal Institute of Technology Lausanne (EPFL), CH-1015 Lausanne, Switzerland

Abstract

The goal of the thesis is to investigate different approaches that combine and integrate Automatic Speech Recognition (ASR) and Speaker Recognition (SR) systems, with applications to (1) User-Customized Password Speaker Verification (UCP-SV) systems, and, (2) joint speech and speaker recognition.

Unlike text-dependent speaker verification systems, in UCP-SV systems, customers can choose easily their own password, which has to be pronounced a few times during enrollment to create a customer specific model that will be subsequently used for verification. The main assumption in such systems is that no *a priori* knowledge about the password (such as its phonetic transcription) is available. However, although more user-friendly and more secure, UCP-SV systems are less understood and actually exhibit several new challenges, including: automatic inference of Hidden Markov Model (HMM) password (using a speaker-independent ASR system), fast speaker adaptation of the resulting acoustic models, score normalization, and verification of both lexical and speaker characteristics. Development and evaluation of such systems are then based on their ability to jointly verify: (1) the identity of a claimed speaker, (2) pronouncing the correct password, and thus rejecting all other possible alternatives.

In this thesis, two different speaker acoustic modeling approaches are investigated: HMM/GMM approach (based on Gaussian Mixture Model, GMM) and hybrid HMM/MLP approach (based on Multi-Layer Perceptron, MLP). In the case of HMM/GMM approach, the background model used for likelihood normalization was the main difficulty, and several solutions were investigated to improve the baseline system. In the HMM/MLP approach, MLP adaptation was also a problem. In that context, we found that the modeling capability of the adapted MLP was more towards learning the lexical content of the password than the customer's voice characteristics. Therefore, a probabilistic framework that combines the hybrid HMM/MLP systems and GMM is proposed and extensively investigated. In this case, the HMM/MLP system is used for utterance verification, while GMM is used for speaker verification.

Since UCP-SV involves both speech recognition (ASR) and speaker verification (SV), a natural extension of our work was to also investigate new approaches towards using ASR together with Speaker Recognition (SR) to improve both ASR and SR systems. In this framework, we show in this thesis that optimization and recognition based on a joint ASR-SR posterior probability criterion yields better ASR and SR performance, beyond what could be achieved from the two systems independently, as well as from a "sequential" approach (e.g., first performing speaker identification/clustering, followed by speech recognition).

This work resulted in a PC-based real time implementation of an HMM based UCP-SV system available for demonstration.

Version abrégée

L'objectif de cette thèse est d'explorer differentes approches qui combinent et intègrent les systèmes de la reconnaissance automatic de la parole (ASR) et la reconnaissance du locuteur (SR), avec comme applications la vérification du locuteur basée sur un mot de passe personnalisé (UCP-SV) et la reconnaissance jointe de la parole et du locuteur.

Contrairement aux systèmes de vérification du locuteur dépendant du texte, dans les systèmes UCP-SV, les utilisateurs peuvent choisir facilement leur propre mot de passe qui va être prononcé un petit nombre de fois pendant l'enregistrement du client pour créer un modèle qui va être utilisé pendant l'accès au système. L'hypothèse principale dans tels systèmes est qu'aucune connaissance *a priori* sur le mot de passe n'est disponible (comme example la transcription phonétique). Cependant, bien qu'ils soient facile d'utilisation et plus sécurisés, les systèmes UCP-SV posent des nouveaux défis, comme : l'inférence automatique du modèle HMM du mot de passe, l'adaptation rapide du modèle inferé aux caractéristiques du client, la normalisation des scores et la vérification. Le développement et l'évaluation d'un tel système reposent sur son aptitude à vérifier conjointement : (1) l'identité proclamée par le locuteur (2) qu'il prononce le mot correct et pas un autre. Cela demande l'exploration de différentes approches utilisant conjointement la vérification du locuteur et la reconnaissance automatique de la parole.

Dans cette thèse, deux approches différentes pour la modélisation acoustique du locuteur sont étudiées : les systèmes HMM/GMM et les systèmes HMM/MLP. Dans le cas des systèmes HMM/GMM, la difficulté principale réside dans le modèle de normalisation utilisé pour normaliser la vraisemblance des données telle qu'estimée par le modèle client. Plusieurs solutions sont explorées pour améliorer la compétitivité du modèle de normalisation et ainsi d'améliorer les performances du système de base. Dans le cas des systèmes HMM/MLP, l'adaptation du MLP est le problème principal. On a observé qu'en pratique le processus d'adaptation modélise plus le contenu lexical du mot de passe que les caractéristiques vocales du client. Pour surmonter ce problème, un cadre probabilistique qui combine les avantages des systèmes hybrides HMM/MLP et des GMMs est proposé. Dans ce cadre, le système HMM/MLP est utilisé pour la vérification de l'énoncé et le GMM pour la vérification du locuteur.

Puisque le UCP-SV comprend à la fois la reconnaissance de la parole et la vérification du locuteur, une extension naturelle de l'étude précédente est d' explorer une nouvelle approche où les systèmes de reconnaissance de la parole sont utilisés avec les systèmes de reconnaissance du locuteur, dont le but d'améliorer les performances des deux systèmes. Dans ce cadre, on a montré que l'optimisation et la reconnaissance basées sur le critère de la probabilité a posteriori conjointe du ASR-SR produit des performances meilleures que celles obtenues par une approche "séquentielle".

Ce travail a abouti à une implémentation d'un système UCP-SV temps réel basé sur les modèles HMM et disponible pour une démonstration.

Contents

List of Figures

List of Tables

Chapter 1

Thesis Overview

1.1 Introduction

Speech recognition aims to extract the lexical information from a speech signal. Speaker recognition aims to recognize (identify or verify) the speaker's identity from a speech signal. Both tasks take the same signal as input but for two different purposes. As a results, and ideally, speech recognizers are designed to ignore or remove the information in the signal that may be useful for speaker recognition, while speaker recognition should mainly extract features that are characteristics to the speaker (Heck, 2002). However, the recognition outputs of both recognizers might contain complementary information that can be combined to improve the performance of each of the systems used independently or to improve the joint speech and speaker recognition rate.

This thesis investigates two complementary areas involving both Automatic Speech Recognition (ASR) and Speaker Recognition (SR), i.e., text-dependent speaker verification systems, with the assumption that users can choose their own passwords and the combination of speech and speaker recognition systems with the goal of improving the performance of both systems.

Speaker Verification (SV) is the task of automatically accepting or rejecting a claimed identity based on the voice characteristics of a speaker (Furui, 1994). In Text-Dependent Speaker Verification (TD-SV) systems, and to get enrolled into the system, the user is prompted to pronounce a predefined password selected from a limited vocabulary (e.g., a digit string corresponding to an account number). A model which represents both the lexical content of the password as well as the speaker voice characteristics is then created from this enrollment data. In this case, the system knows in advance the transcription (in terms of phonemes or digits) of the password. Because the user has no freedom to choose the predefined password, TD-SV systems are usually not fully appreciated by the users.

In this thesis, a particular type of text-dependent speaker verification system, referred to as *User-Customized Password Speaker Verification* (UCP-SV), is investigated. The main assumption in this system is that no *a priori* information about the password is available to the system at the time of enrollment. The customer can choose any password, on which the identity verification will be done later, without any constraint. In this case, the lexical content of the password should be extracted automatically (in terms of subword units such as phonemes) from the enrollment data. Furthermore, the decision to accept or reject a speaker will involve the joint verification of (1) the pronounced password and, (2) the claimed identity.

As a natural extension to our work in UCP-SV systems, we have also investigated a new probabilistic framework towards using ASR systems together with SR systems. In this framework, the goal is to maximize the joint posteriori probability of the pronounced word and the speaker identity given the observed data.

1.2 Topics of the thesis

1.2.1 User-Customized Password Speaker Verification (UCP-SV)

In this thesis, a language dependent UCP-SV system is studied, where speakers (customers) have the possibility to choose their own password on which identity verification can be performed without any lexical constraint. Consequently, it will be more difficult for an impostor to guess the customer's password. For increased security, customers can also change their passwords easily at any time. However, to make the system practical, the enrollment session should be short, simply requiring a few repetitions of the password. Consequently, UCP-SV systems should exhibit some properties, such as:

1. *Password length:* For practical reasons, customers have the tendency to choose a short passwords. This might affect the accuracy and the robustness of the system, since the customer acoustic modeling and the decision making will be then based on a very limited speech data which usually yielded lower performance.

2. *Multiple references modeling:* Unlike most TD-SV systems, where each customer is modeled by one reference model (corresponding to the correct phonetic transcription of the password), in UCP-SV systems each repetition of the password can be used as an alternative pronunciation model for the customer's password. This allows customer to be represented by multiple reference models. The few repetitions of the password can be considered as a high-level information about how the customer pronounces his/her password. This information can be useful in reducing the mismatch between train and test data due to the pronunciation variability which increases the robustness of the UCP-SV system[1].

The decision to accept or reject a speaker then involves two hypothesis tests (see Section 2.4): (1) hypothesis testing of the customer password where the null hypothesis is the pronounced word corresponds to the customer's password and (2) hypothesis testing of speaker identity where the null hypothesis is the speaker identity corresponds to the customer.

However, UCP-SV systems present new challenges. First the system has to automatically infer the topology of the Hidden Markov Model (HMM) (see Section 3.2.2) associated with the password simply based on a few utterances. To achieve this, we need a well trained speaker-independent phone recognition system. Once the topology of the password has been extracted, we then have to parametrize the inferred model in terms of speaker-independent parameters that can be easily adapted to the customer.

HMM inference

The HMM model associated with the password should be inferred automatically in terms of phone-like sub-units from a few utterances of the customer password. The inferred phonetic transcriptions (PT) should be representative of the lexical content of the password. The accuracy of the inferred phonetic transcriptions depends on the accuracy and the consistency of the speech recognizer. Ideally, the inferred phonetic transcriptions should be almost the same for all utterances of the same password, but in practice, this is never the case. Therefore, we have to find a way to consolidate all resulting models or to pick the one that best represents the enrollment data.

[1] SuperSID project at the JHU summer workshop, 2002, http://www.clsp.jhu.edu/ws2002/groups/supersid

In this thesis, the HMM inference step is performed using a speaker independent phoneme based speech recognizer derived from a standard large vocabulary continuous speech recognition system and an ergodic lexical model to map each of the enrollment utterances into a phonetic sequence. We have compared the use of a speaker-independent hybrid Hidden Markov Model/Artificial Neural Network (HMM/ANN) system (see Section 3.4) and a speaker-independent Hidden Markov Model/Gaussian Mixture Models (HMM/GMM) system. In HMM/GMM systems, a GMM is used to estimate HMM state emission probabilities, while in HMM/ANN systems a ANN is used to estimate HMM state posterior probabilities. The hybrid HMM/ANN systems have been successfully used in speech recognition and are known to consistently yield better accuracy at the frame level compared to HMM/GMM systems, thus being better suited to recognize utterances in terms of phone sequences.

Speaker adaptation

Having inferred the HMM topology associated with the customer password, the next step is to create the customer specific acoustic model. Since in UCP-SV systems, the amount of enrollment data is very limited, a solution is to start from a speaker-independent speech recognizer that will be used as *a priori* information about the acoustic space, and adapt its parameters to the speaker specific characteristics. The adaptation process requires us to consider an appropriate parameter adaptation scheme as well as the number of parameters that will be updated.

In this thesis, two approaches were investigated: hidden Markov models and hybrid HMM/ANN models.

1. Hidden Markov models are the current state-of-the-art in text-dependent speaker verification systems. In these models, Gaussians Mixture Models (GMM) (see Section 3.2.1) are used to estimate the HMM state emission probabilities. The speech data provided by the customer is used to adapt the parameters of these GMMs to create a customer-dependent HMM model. Within this approach, two UCP-SV systems were compared. In the first system, the best inferred phonetic transcription was used to create the customer HMM model. Consequently, each customer was modeled by one HMM model. In the second system, each inferred phonetic transcription was used to create a customer HMM model. Consequently, each customer was represented by multiple HMMs. The performances of these two systems are compared with a TD-SV system where the correct phonetic transcription of the password is known. This comparison allowed us to evaluate the effect of the accuracy of the inferred phonetic transcription on the performance of the UCP-SV system and to identify some problems of UCP-SV systems.

2. In the hybrid HMM/ANN approach, a supervised adaptation technique was used to adapt the parameters of a ANN model for each customer. The adapted ANN is then used to estimate the inferred HMM state posterior probabilities instead of GMMs. Due to the amount and the nature of the adaptation data, different MLP adaptation techniques were tested. Because the ANN was adapted to discriminate between phone classes and not between customer and impostors, most of the information used to make the decision to accept or reject a speaker was based on the lexical content of the test access. Consequently, the performance of the UCP-SV was lower from what we have expected.

1.2.2 Combination of speech and speaker recognition

Speech signal conveys different kind of information, including the spoken language, the dialect, the lexical content of the utterance, the transmission channel and information about the speaker (such as age and gender). Ideally, the processing and the analysis of the speech signal should be different depending on the final application. Only information that is useful for the application should be extracted while the other information will be discarded. For example, in speaker-independent speech recognizers, we are more interested in information that is useful to characterize the lexical content of the speech segment, independently of the speaker. So, the effect of inter-speaker variability is often reduced through speaker normalization techniques. Similarly, in speaker recognition, the lexical content of the speech segment is usually considered as unnecessary information (Heck, 2002). Because the type of information extracted from the signal for speech and speaker recognizers is different, they might contain complementary information that can be beneficial to improve the performance of each system individually or to perform joint speech and speaker recognition. The combination of speech and speaker recognition systems could also be beneficial when both systems are trained with two different criteria, even if they use the same feature set, as illustrated in Figure 1.1.

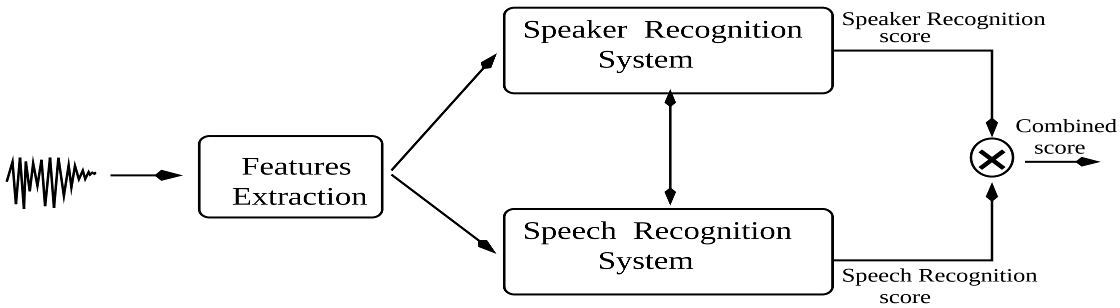

Figure 1.1. *A block diagram for speech and speaker scores combination*

Finally, joint speech and speaker recognition is important for many applications, such as interaction with an automated voice systems, verbal information verification (Li *et al.*, 2000), and information retrieval from audio data. In general, joint speech and speaker recognition can be useful in any application where we are interested in *who is speaking and what was said*. For this purpose, new probabilistic approach combining in a principle way the speech recognition score and the speaker recognition score was investigated. The speech recognition score is *posteriori probability* based and, is estimated through a speaker-dependent hybrid HMM/ANN system. The speaker recognition score is *likelihood* based and, is estimated through a text-independent speaker recognition system. This approach was first used to improve the performance of the hybrid HMM/ANN based UCP-SV system described above, and then tested for different applications, including: closed-set speaker identification, open-set speaker identification [2], and speaker clustering [3]. For closed-set and open-set speaker identification, the goal was to enhance the speaker identification performance, so the joint speech and speaker recognition performance will improve. In speaker clustering, the goal was to find from a set of reference speakers the one whose voice characteristics are the closest to the actual speaker in order to perform a speaker-independent speech recognition task.

1.3 Contributions

In the following, I briefly discuss what I believe are the main contributions resulting from the present thesis.

[2] See the next Chapter, for a definition of closed-set and open-set speaker identification
[3] See Chapter 6 for a definition of speaker clustering

User-customized password speaker verification

- *TD-SV systems:* To verify the lexical content of the test utterance, most TD-SV systems use an HMM word model with speaker-independent parameters. In this thesis, the speaker-dependent password HMM was used. This improved the discriminative capabilities of the utterance verification part between customer and impostors.
- *Baseline HMM/ GMM based UCP-SV system:* A baseline UCP-SV system based on HMM/GMM was developed. A comparison with a reference TD-SV system allowed us to draw the main problems of this baseline system.
- *Multiple reference models:* Based on the previous comparison, the use of multiple reference models to reduce the pronunciations variability effect and increase the discriminant capabilities of the system between the customer and impostors was investigated and several speaker verification score estimation criteria were proposed and compared. These proposed criteria included: dynamic model selection techniques, score fusion techniques and decision fusion techniques.

  The resulting best system was implemented on a PC-based real time version available for demo.
- *Hybrid HMM/ ANN based UCP-SV systems:* The use of hybrid HMM/ANN for UCP-SV systems was also investigated. This investigation included: ANN adaptation and posteriori probability based score estimation. Our conclusion was that the posteriori probabilities estimated at the outputs of the speaker-dependent ANN are more dependent on the lexical content of the pronounced password than the speaker characteristics. Consequently, the use of hybrid HMM/ANN for speaker verification task has appeared to be not effective.
- *Probabilistic framework for utterance and speaker verification:* Based on the above conclusion, a probabilistic framework was developed, where both the password content and the speaker voice characteristics are modeled separately. In this framework, the hybrid HMM/ANN was used only for modeling the customer password, while a Gaussian mixture model was used to capture the speaker voice characteristics.

Combination of speech and speaker recognition

- *Joint speech and speaker recognition:* The above probabilistic framework was extended to perform joint speech and speaker recognition. By combining the speaker recognizer and speech recognizer scores, it was shown that it is possible to simultaneously improve both the speech and speaker recognition performance.
- *Speaker identification:* Applications to closed-set and open-set speaker identification were shown that the proposed framework outperform the standard approaches.
- *Speaker clustering:* This framework was also tested to perform speaker clustering (to select the closest speaker) for speaker-independent speech recognition task in a open set application. The obtained word recognition rate was the best among others obtained using standard approaches.

1.4 Thesis outlines

Chapter 2 will review the basis of speech recognition and speaker recognition systems, with strong emphasis on speaker recognition systems. A short review of the statistical hypothesis tests and some basic speaker verification tools such as performance evaluation and thresholding will be given. A description of the cepstral feature extraction process, typically used for speech and speaker recognition systems as well as the databases used to curry out our experiments will be given.

Chapter 3 will discuss some of the speaker acoustic modeling approaches (such as Gaussian mixture models, hidden Markov models and artificial neural network) and how they are used in a speaker recognition task. Some of the model-based adaptation techniques and likelihood ratio based score normalization techniques will also be described. This chapter will describe the use of the hybrid HMM/ANN models for speech recognition task. This includes the description of ANN training procedure and the decoding procedure within HMM framework using state posterior probabilities estimated by the artificial neural network.

Chapter 4 will discuss the development of a baseline HMM based UCP-SV system, including a description of the HMM inference procedure and speaker adaptation. This system uses the best phonetic transcription to create the customer HMM password model. A comparison with a reference TD-SV system where the correct phonetic transcription of the password is known shows that the main issue in HMM based UCP-SV system stem in the background model. That is a model used to normalize the likelihood estimated by the customer model. This chapter will then, investigate the use of multiple reference models to improve the performance of the baseline system. This investigation includes mainly the description of some speaker verification scoring criteria. This will be the main contribution of this chapter.

Chapter 5 will discuss, investigate and analyze the use of the hybrid HMM/ANN systems for UCP-SV. These systems will be used in the same way they are used for speech recognition. The investigation will focus on the MLP adaptation and the estimation of the verification score. The conclusion of this investigation is that the hybrid HMM/ANN are not effective for a speaker verification task as they are for a speech recognition. Their drawback is that they do not capture properly the speaker characteristics. This chapter will end up with a novel probabilistic framework that combines the advantages of the hybrid HMM/ANN system and GMMs. The use of this framework for UCP-SV as well as the obtained results will be discussed and analyzed. This framework is the most important contribution in this chapter.

Chapter 6 will extend the framework developed in chapter 5 to allow performing joint speech and speaker recognition. Three different task on which this framework is tested and compared with standard approaches will be described. These tasks are closed-set speaker identification, open set speaker identification and speaker clustering for speaker-independent speech recognition.

Chapter 7 will summarize the work done in this thesis and describe some future works.

Chapter 2

Speaker Recognition Systems

2.1 Introduction

This chapter reviews the principals of both speech and speaker recognition systems with an emphasis on speaker recognition systems. We will describe the functional components of these systems. We will review briefly the notions of hypothesis testing, decision threshold and performance evaluation. We will describe the feature extraction procedure to extract the low-level acoustic parameters from the speech signal that form the basis for speaker specific acoustic modeling. Finally, a description of databases used to carry out the experiments and evaluate our modeling approaches will be given.

2.2 Speech recognition systems

The aim of speech recognition systems is to extract the lexical content from a speech signal. This procedure is performed in several successive steps. More details about this subject can be found in (Rabiner and Juang, 1993; Gold and Morgan, 2000; Huang *et al.*, 2001). A block diagram of a typical speech recognition system is shown in Figure 2.1.

- *Feature extraction:* First, acoustic feature vectors are extracted from the speech signal (waveforms). These features convey the relevant information that is useful to characterize a subword unit (typically phonemes) to be recognized and distinguished from other different subword units. This information should be also robust to different sources of variations such as noise, articulator effects and pronunciation variation. That means, this information should be almost the same for the same sub-word unit even if this subword unit is pronounced by the same speaker or different speakers in different conditions.

- *Acoustic modeling:* The acoustic features are then used to create an acoustic model for each subword unit. This step is known as *training*. The goal of training is to estimate the parameters of the acoustic model that best represent the training data and minimize the recognition error on unseen data. There are many factors that affect the acoustic model and make it less robust when the recognizer is running in real applications. Among them, the amount of training data and the mismatch between train and test data are the important factors. Several acoustic modeling approaches exist, but today's speech recognizers use a HMM model where the state emission probabilities are estimated using either Gaussian mixture models (Rabiner, 1989) or artificial neural networks (ANN) (Bourlard and Morgan, 1994). In this thesis,

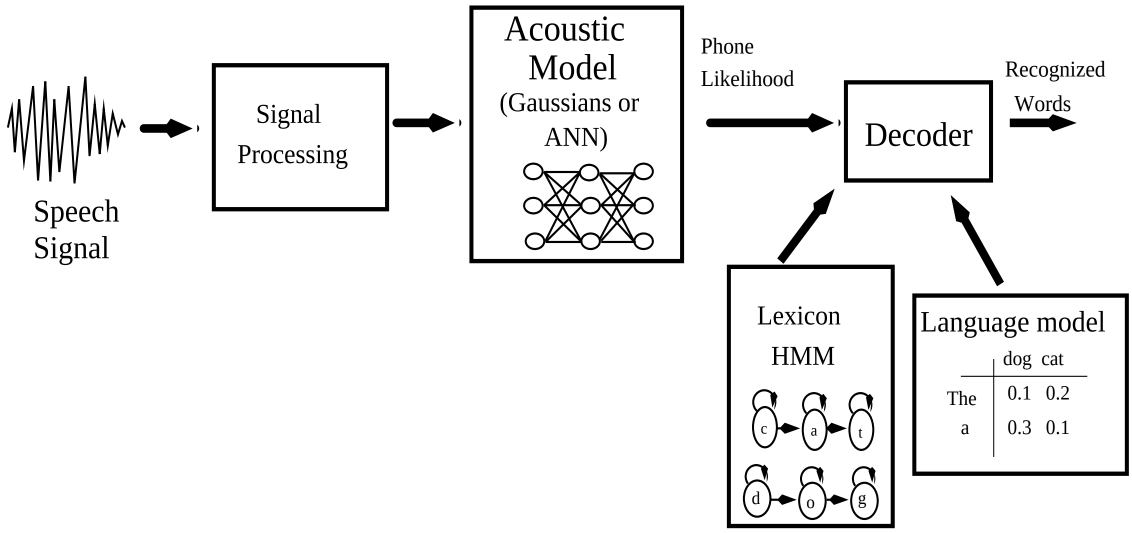

Figure 2.1. *Block-diagram showing the main speech recognizer components: First, the speech signal is analyzed to extract acoustic features which then used in the acoustic model (second block) to estimate the local probabilities of each sub-word unit. These probabilities are then used by the decoder for the temporal alignment using some constraints represented by the lexicon and the language model. The results is a sequence of recognized words.*

the former approach will be referred to as HMM/GMM approach, while the later approach will be referred to as hybrid HMM/ANN approach.

- *Decoding:* The speech recognition task consists of finding the sequence of words that best matches the speech data according to a certain criterion. This task is known as decoding process. The decoding is performed using local similarity (distance) measure (Gold and Morgan, 2000) expressed in terms of likelihoods (HMM/GMM) or posteriori probabilities (HMM/ANN) that can be converted to *scaled likelihood* (see Section 3.4). This similarity measure indicates how good an acoustic vector belongs to each sub-word unit. The decoder uses this similarity measures with the temporal information embedded in the acoustic model (like transition probabilities between different sub-word models) to search through the acoustic space to find the best sequence of sub-word units corresponding to the sequence of recognized words. In the case of HMM/GMM or HMM/ANN approaches, the decoding is performed using Viterbi alignment (Viterbi, 1967).

  To improve the recognition accuracy, some *a priori* linguistic knowledge about the sequences to be recognized can be used during the decoding. This knowledge is usually embedded into a *language model*.

2.3 Speaker recognition systems

Speaker recognition is the process of recognizing people by their voices. The goal is to extract the information in the speech signal which conveys speaker characteristics. Indeed, the speech signal contains many characteristics that are specific to the speaker and are difficult to reproduce by any other speaker. Such characteristics are independent of the linguistic message.

2.3.1 Speaker recognition components

As illustrated in Figure 2.2, the first step in building a speaker recognition system is known as the *enrollment step* which consists of creating an acoustic model for the speaker using some speech data provided by the speaker. The effectiveness of the speaker model depends upon, among other factors, the amount and the quality of the training data available for that speaker, which is usually very limited in real applications.

During testing, the recognition step consists mainly of three modules:

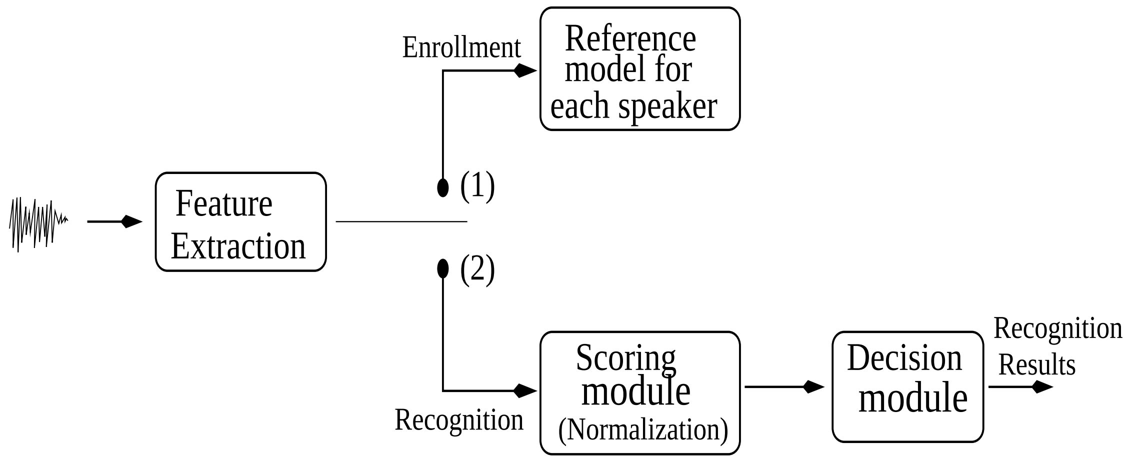

Figure 2.2. *Block diagram showing the main components of speaker recognition systems.*

1. *Feature extraction:* For speaker recognition, feature extraction consists of extracting the information that is useful to characterize a speaker's voice to be recognized and distinguished from other different speakers. It follows that features that exhibit high inter-speaker variability and low intra-speaker variability are desired (Campbell, 1997). Some of feature extraction techniques will be described in Section 2.6.

2. *Scoring module:* This module estimates the recognition score that represents the reliability of the hypothesis that the speech segment comes from the claimed identity. Depending on the task (speaker identification or verification) and the modeling approach, we might need an additional model known as *anti-speaker* model to estimate the score.

3. *Decision module:* Based on the estimated recognition score, this module makes the decision to confirm (in the case of speaker verification) or establish (in case of speaker identification) an individual's identity.

Speaker recognition can be categorized depending on the task (speaker identification or speaker verification) and the text used for verification (text-dependent or text-independent). For more details, some general overview papers on speaker recognition are (Doddington, 1985; Furui, 1994; Campbell, 1997; Bimbot *et al.*, 2004).

2.3.2 Speaker identification versus speaker verification

Depending on the application context, speaker recognition can be divided into speaker identification and speaker verification tasks. They have the same feature extraction module but different scoring and decision modules.

- *Speaker identification systems:*
  In Speaker IDentification (SID) system, no identity claim is provided, the test utterance is scored against a set of known (registered) references for each potential speaker and the one whose model best matches the test utterance is selected. In speaker identification, we have to distinguish between *closed-set* and *open-set* speaker identification. In *closed-set*, the test utterance belongs to one of the registered speakers. During testing, a matching score is estimated for each registered speaker. The speaker corresponding to the model with the best matching score is selected. This requires N comparisons for a population of N speakers. In *open-set*, any speaker can access the system, those that are not registered should be rejected. This requires another model referred to as "garbage model" which is trained with data provided by other speakers different from the registered speakers. During testing, the matching score corresponding to the best speaker model is compared with the matching score estimated using the "garbage model" in order to accept or reject the speaker, making the total number of comparisons equal to $N + 1$. Speaker identification performance tends to decrease as the population size[1] increases.

- *Speaker verification systems:*
  In Speaker Verification (SV) systems , the speaker is classified as having the purported identity or not. That is, the goal is to automatically accept or reject an identity that is claimed by the speaker. During testing, a verification score is estimated using the claimed speaker model (and the *anti-speaker* model in case of using generative models like Gaussian mixture models). This verification score is then compared to a threshold. If the score is higher than the threshold, the speaker is accepted, otherwise, the speaker is rejected. Thus, speaker verification, involves an hypothesis test requiring a simple binary decision: accept or reject the claimed identity regardless of the population size. Hence, the performance is quite independent of the population size, but it depends on the number of test utterances used to evaluate the performance of the system.

2.3.3 Text-dependent versus text-independent

Speaker recognition can be based on text-independent, text-dependent and text-prompted utterances, depending on whether there is a constraint or not on the text during the enrollment and/or recognition phases.

- *Text-dependent:*
  In Text-Dependent (TD) speaker recognition system, the enrollment process requires the password (which is usually a sequence of digits corresponding to the account number) to be predefined or its transcription (in term of phonemes or digit) to be available to the system. In other words, the system has *a priori* knowledge about the password. The training and the test text are the same. For each individual, there is a model that encodes the speaker's voice characteristics associated with the phonemic or syllabic content of the password. Since recognition is based on the speaker characteristics as well as the lexical content of the password, text dependent speaker recognition systems are generally more robust and achieve good performance. The text-dependent system studied in this thesis has the particularity that the user (and not the system) who chooses the password.

- *Text-Independent:*
  In the case of Text-Independent (TI) speaker recognition, the lexical content of the utterance used for recognition can not be predicted. To access the system, the test utterances can be

[1] Population size: number of registered speakers

different from those used for enrollment, hence, text-independent speaker verification needs a large and rich training data set to model the characteristics of the speaker's voice and to cover the phonetic space. Consequently, without a large training set and long test segments, the performance of a text-independent speaker recognition system is usually below that of their text-dependent equivalent (Li *et al.*, 1999).

- *Text-prompted:*
  Both text-dependent and text-independent systems are susceptible to fraud, since for typical applications the voice of a speaker could be captured, recorded, and reproduced. To limit this risk, a particular kind of text-dependent speaker verification systems based on *prompted text* have been developed. In this case, a recorded or synthetic prompt asks the user to utter a different random sentence each time the system is used (Higgins *et al.*, 1991; DeVeth and Bourlard, 1995; Che *et al.*, 96). The underlying lexicon which could either be very large or limited to just 10 digits, would then be used to generate random sentences. During recognition, speaker-dependent phoneme models are concatenated according to the prompted-text and a verification score is then estimated. The advantage of such an approach is that impostors cannot predict the prompted sentence. Consequently, pre-recorded utterances from the customer are of no use to the impostor. As in the case of text-independent systems, the text-prompted systems also need a large and rich training data set for each registered speaker to create a robust speaker-dependent models.

2.4 Hypothesis testing

Speaker verification is a classification problem, in which we have to make a decision whether to accept or reject a speaker based on the verification score obtained by a measurement process. The goal here is to determine if a speech segment X belongs to the claimed speaker or not. There is a choice between two hypotheses. The null hypothesis, denoted by H_0 states that the speech segment belongs to the claimed speaker and the alternative hypothesis, denoted by H_1 states that the speech segment does not belong to the claimed speaker and belongs to somebody else. The choice between H_0 and H_1 is simply made based upon the *posteriori probability* of each hypothesis given the test utterance X. That is:

$$S = S_c \;\text{ if }\; P(H_0|X) \geq P(H_1|X) \tag{2.1}$$

where S and S_c are the test speaker and the claimed speaker identities, respectively. These posteriori probabilities can be rewritten as a function of the conditional Probability Density Function (PDF) or likelihoods defined as $\{p(X|H_i)\}_{i\in\{0,1\}}$. and *a priori* probabilities using Bayes rule as follows [2]:

$$S \;=\; S_c \;\text{ if }\; \frac{p(X|H_0).P(H_0)}{P(X)} \geq \frac{p(X|H_1).P(H_1)}{P(X)} \tag{2.2}$$

Since $P(X)$ is common to both sides of the inequality, the decision rule can be simplified to:

$$S \;=\; S_c \;\text{ if }\; \frac{\mathrm{p(X|H_0)}}{\mathrm{p(X|H_1)}} \geq \frac{\mathrm{P(H_1)}}{\mathrm{P(H_0)}} \tag{2.3}$$

where $\frac{p(X|H_0)}{p(X|H_1)}$ is called the *likelihood ratio*, and $\frac{P(H_1)}{P(H_0)}$ is called the *threshold*. This is the optimum test according to the Bayes decision rule (Fukunaga, 1990). If we assume the two *priori* probabilities $P(H_0)$ and $P(H_1)$ to be equal, then the theoretical value of the threshold is equal to one.

[2] In this thesis, $p(.)$ and $P(.)$ will be used for likelihoods and posteriori probabilities, respectively

The likelihood ratio based decision rule is optimal if we know exactly the PDF for both hypotheses. In practice, this is not the case and the PDFs of both hypotheses are assumed to be a parametric representations estimated from training data. Since the true PDFs $p(X|H_0)$ and $p(X|H_1)$ and the *a priori* probabilities, $P(H_0)$ and $P(H_1)$, are often unknown, generally $\frac{P(H_1)}{P(H_0)}$ is replaced with an experimental threshold T determined to satisfy the application requirements. So, the decision after taking the logarithm of the likelihoods becomes:

$$S = S_c \text{ if } \log p(X|H_0) - \log p(X|H_1) \geq \log T \tag{2.4}$$

The central questions now is how to evaluate the performance of speaker verification systems? and how to determine the threshold?

2.5 Performance evaluation

2.5.1 Error measure

A speaker verification system could make two types of errors; the *false rejection* (FR) occurring when an authorized speaker is classified as an impostor and is rejected, and the *false acceptance* (FA) occurring when an impostor is accepted as a valid target speaker. Consequently, the decision should be made to minimize both FA and FR errors. In real application, we should take into account the application requirements, which is a trade off between these two types of errors. In a more secure system, it may be desired to have a low FA rate at the expense of higher FR rate, and vice-versa for a more convenient system. The various costs and impacts of these two types of errors as well as other factors must be taken into account to evaluate the utility of the system. The system performance can be depicted using a graphical representation such as a Receiver Operating Characteristic (ROC) or Detection Error Trade-off (DET) (Martin *et al.*, 1997) curve. In both representations, the probability of false acceptance (horizontal axis) is plotted versus the probability of false rejection (vertical axis) for varying decision threshold. The better the system, the closer to the origin the curve will be. Figures 2.3 and 2.4 are examples of DET and ROC curves, for the same system performance.

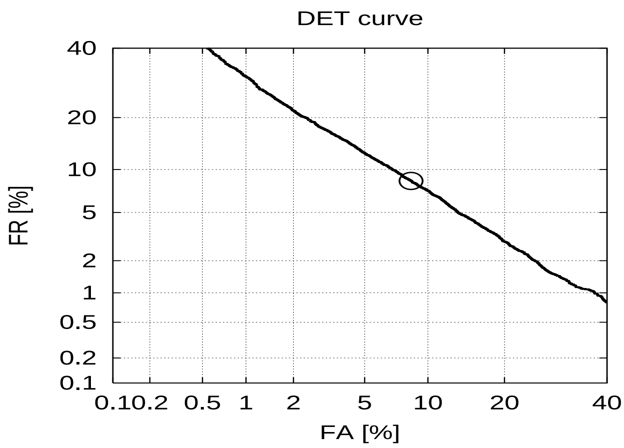

Figure 2.3. *The DET curve: The false acceptance (FA) is plotted against false rejection (FR). The circle corresponds to the EER.*

The difference between the ROC and DET curves is that, the ROC plot is on a linear scale while the DET plot is on a normal deviate scale. That is instead of plotting the percentage of false

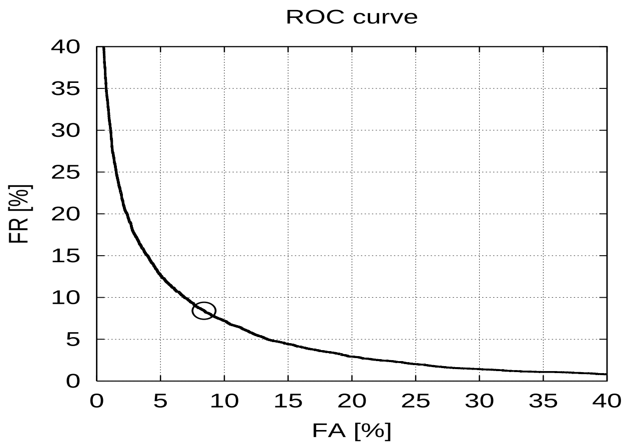

Figure 2.4. *The ROC curve corresponding to the DET curve shown in Figure 2.3, false acceptance (FA) is plotted against false rejection (FR). The circle corresponds to the EER.*

acceptance and false rejection rates, these two values are expressed as the number of standard deviations from the mean of a normal distribution (Wan, 2003). If the scores of impostor and target accesses are Gaussian distributed, then the result is a linear DET curve with a slope equal to -1. For system comparison, the performance is often reported in terms of EER (equal error rate), corresponding to the decision threshold where the False Rejection Rate (FRR) is equal to the False Acceptance Rate (FAR). Another error that is often reported to compare systems performance is known as HTER (Half Total Error Rate), corresponding to the arithmetic mean of FAR and FRR associated with an operating point.

$$HTER = \frac{FA + FR}{2} \tag{2.5}$$

The performance of a speaker verification system depends on the amount of the training data, training and testing conditions and the length of the test segment, hence, it is important to also report such information during the performance evaluation (Kung *et al.*, 2004).

2.5.2 Decision threshold

The robustness of a speaker verification system depends among other factors, on the reliability of the decision threshold which can be speaker-dependent or speaker-independent. The estimation of the optimal threshold is a critical problem and it depends closely on the application. A low threshold will result in a high FAR while a high threshold will result in a high FRR, each of which can pose problems for the service provider and user. Therefore , the threshold should be set according to the importance associated with each type of error. Often the performance of speaker verification system is reported using *a posteriori* EER threshold. That is, the threshold is estimated by assuming that the distributions of the test scores under the null hypothesis and alternate hypothesis are Gaussians and computing the resulting EER point (FRR = FAR).

In real task, the threshold should be determined *a priori*. Many *a priori* threshold setting techniques have been proposed (Pierrot *et al.*, 1998; Zhang *et al.*, 1999; Surendran and Lee, 2000; Chen, 2003). A common method to estimate a *priori* threshold is to conduct an experiment on a development dataset composed of train and test data provided by speakers different from the actual client population. The estimated *a posteriori* threshold is then used as *a priori* threshold for the current application. It is clear that if the development dataset is not representative of the application conditions then the reliability of the estimated threshold can not be ensured.

Another method for threshold estimation is to divide the enrollment data provided by each speaker into two parts. One part is used to create the speaker model and the other part to estimate the threshold. The difficulty here arises from the fact that usually very limited data is available. Consequently, estimating the threshold on a half portion of the data is detrimental to the accuracy of the speaker model.

The *a priori* threshold gives a feasible way to create a real speaker verification system, while the *a posteriori* threshold gives a good way to evaluate the discrimination capabilities of the client speaker model and to compare objectively the modeling performance (Chen, 2003).

2.6 Feature extraction

In speech recognition system, the goal of feature extraction is to extract the acoustic information that is representative of the lexical content and invariant to the speaker. Speaker verification systems require acoustic features that are representative of speaker's voice characteristics and independent (in case of text-independent speaker verification) on the lexical content of a particular word (Gold and Morgan, 2000). We can distinguish between two types of speaker information conveyed in the speech signal (Doddington, 1985), low-level information and high-level information. Low-level information denotes the information such as pitch and magnitude spectrum. Automatic speaker recognition systems typically use low-level information. High-level information denotes information such as dialect, accent, the way the speaker pronounces specific words and speaking style. Such speaker specific traits are often used by human beings to recognize people, but they are less used in speaker recognition systems, because they are very difficult to quantify.

Current speaker recognition systems use acoustic features that have been developed for use in speech recognition. They are based on the short-term spectral analysis. These features do not capture long-term prosodic information such as fundamental frequency $F0$ and which provide more speaker specific information. Experimental results showed that fundamentatl frequency feature $F0$ is robust for speaker recognition in noisy conditions and the performance of a speaker verification system can be improved if it is added to the conventional short-term spectral features (Jankowski *et al.*, 1995; Sarma, 1997; Kajarekar *et al.*, 2003). Unfortunately, this feature is often difficult to estimate reliably, particularly for a short enrollment period such as in UCP-SV systems.

It is worth to note here that the search for the best features is not the subject of this thesis. Therefore, we have used the most current signal representation for speech and speaker recognition, namely MFCC features.

Feature extraction process consists of several consecutive steps and can be summarized as follows: pre-processing, cepstral analysis and post-processing (Rabiner and Juang, 1993; Bimbot *et al.*, 2004).

2.6.1 Pre-processing

The sampled speech (sampling frequency is typically 8 kHz for telephone quality speech) is pre-emphasized to enhance the high frequency components of the spectrum. High frequency formants have low amplitude as compared to low frequency formants. Pre-emphasis of the high frequencies is done to obtain similar amplitude for all formants. This is performed by applying a first order FIR filter to the speech signal:

$$s_p(n) = s(n) - as(n-1) \tag{2.6}$$

where a is the pre-emphasis coefficient, and is kept in the range of $0.9 \leq a \leq 1$. A typical value for a is 0.95.

Most of signal analysis techniques assume that the signal is stationary. For speech signal, this is valid only within a short-time intervals. Therefore a *windowing* of the signal is performed, that is a window whose length is typically between 20 and 30 ms is applied to the beginning of the signal and moved every 10 ms until the end of the signal is reached. This results in a succession of windowed sequences called *frames*:

$$y(n) = w(n).s_p(n) \tag{2.7}$$

where $w(n)$ is the impulse response of the window. The most commonly used window in speech and speaker recognition is *Hamming* window, whose impulse response is defined as follows:

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) & n = 0, 1, ...N-1 \\ 0 & otherwise \end{cases}$$

where N is the total number of samples in the frame. Hamming window is preferred over rectangular window because it reduces the effect of sidelobes.

2.6.2 Cepstral analysis

MFCC parameters

A Discrete Fourier Transform (DFT) of the windowed signal is used to extract the frequency content (the spectrum) of the current frame. To reduce the number of obtained spectral parameters and to get the gross shape of the spectral envelope, a smoothing of the spectrum is performed. This is done by applying a number of triangular filters with different center frequencies and bandwidth. Each of these filters extracts the average of the spectral energy in a particular frequency band. Typically, the Mel scale which is an auditory scale similar to the frequency scale of the human ear is used. The outputs of these filters form the spectral envelope. We take the logarithm of this spectral envelope to obtain the log spectral vectors.

Finally, Discrete Cosine Transform (DCT) is applied to the log spectral vectors to obtain the Mel-Frequency Cepstral Coefficients (MFCC) (Davis and Mermelstein, 1980).

$$C_i = \sum_{k=1}^{K} S_k cos\left[i\left(k - \frac{1}{2}\right)\frac{\pi}{K}\right] \quad i = 1, ..., d \tag{2.8}$$

where K is the number of filter banks (the log spectral vectors dimension), S_k is the logarithm of the k^{th} filter bank's outputs and d is the number of cepstral coefficients ($d \leq K$). The MFCC features are used in (Carey *et al.*, 1991; Reynolds *et al.*, 2000; Auckenthaler *et al.*, 2000) for speaker verification.

LPCC parameters

The Linear Prediction (LP) (Makhoul, 1973) based cepstral parameters extraction can be divided into two steps. First, LP-analysis of the speech is performed to compute a set of predictor coefficients. Second, these predictor coefficients are then transformed into cepstral feature vectors.

For each frame, LP analysis assumes that the current speech sample $y(n)$ can be predicted approximately by a linear combination of the past P samples plus an excitation term:

$$y(n) \approx \tilde{y}(n) = \sum_{k=1}^{P} a_k y(n-k) + Gu(n) \tag{2.9}$$

where $a_k, \quad k = 1, ..., P$ are the LP coefficients, P is the LP analysis order, G is the gain and $u(n)$ is the normalized excitation.

The goal of LP analysis is to find the predictor coefficients a_k that minimize the mean squared error between the actual value $y(n)$ and the predicted value $\tilde{y}(n)$ of the samples in a frame.

These P predictor coefficients represent the spectrum envelope of the speech signal and they are used to compute the Linear Prediction Cepstral Coefficients (LPCC). Thus, the first d LPCCs are obtained as follows:

$$C_i = a_i - \sum_{k=1}^{i-1} \left(\frac{k}{i}\right) C_k a_{i-k} \quad 1 \leq i \leq P \;\; and \;\; 1 \leq k \leq d \tag{2.10}$$

$$C_i = \sum_{k=1}^{i-1} \left(\frac{k}{i}\right) C_k a_{i-k} \quad i \geq P \tag{2.11}$$

Resulting in a d-dimensional cepstral vector:

$$C_i = [C_1, C_2, ..., C_d]$$

The LPCC features are used in (Matsui and Furui, 1992b; Rosenberg and Parthasarathy, 1996) for speaker verification.

The cepstral coefficients LPCC or MFCC are usually augmented by adding the log of the signal energy computed as follows:

$$E = \log \sum_{n=1}^{N} s^2(n) \tag{2.12}$$

where N is the number of speech samples in the frame.

2.6.3 Post-processing

- *Cepstral mean subtraction:* In telephone quality speech applications, which is the case of many speaker recognition systems, the transmission channel variations are considered as a source of signal degradation. To compensate the channel effects, *cepstral mean subtraction* (CMS) technique is applied. In this technique, the mean vector of cepstral coefficients is estimated and subtracted from each cepstral vector.

- *Dynamic information:* The cepstral coefficients do not take into account the spectral variations along time. It is well known that the speech recognition performance can be enhanced by adding information about the temporal dynamism of the signal (Sonng and Rosenberg, 1988; Rabiner and Juang, 1993; Huang *et al.*, 2001). Thus, first and second order time derivatives of the cepstral coefficients are also computed using polynomial approximations (Furui, 1981) and are appended to the cepstral features.

$$\delta C_t = \frac{\sum_{k=-l}^{l} k C_{t+k}}{\sum_{k=-l}^{l} |k|} \tag{2.13}$$

$$\delta\delta C_t = \frac{\sum_{k=-l}^{l} k^2 C_{t+k}}{\sum_{k=-l}^{l} k^2} \tag{2.14}$$

where t is a frame at time instant and $2l$ is the context with t^{th} frame in the center.

- *Speech/ non speech segmentation:* The set of acoustic vectors extracted for each frame, do not carry the same level of information for speech or speaker recognition tasks. Feature vectors corresponding to a speech segment are more important than feature vectors corresponding to a silence segment. These silence segments do not carry any useful information and can degrade the performance of the system. For example, in a Gaussian mixture model based speaker recognition systems, it is important to remove the silence segments from the train and test data. In this thesis, we have used a bi-Gaussian model of the log energy distribution trained in an unsupervised manner. The Gaussian with low log energy will represent the silence segments and the Gaussian with high log energy will represent the speech segments.

2.7 Databases and experimental set-up

In this thesis, two databases were used:

1. The *PolyPhone* (Chollet *et al.*, 1996) database to train different speaker-independent speech recognizers.
2. The Swiss-French *PolyVar* database (Chollet *et al.*, 1996) to carry out the speaker verification experiments.

2.7.1 PolyPhone database

The Swiss-French *PolyPhone* database contains telephone calls from about 4, 500 speakers recorded over the Swiss telephone network. The calling sheets were made up of 38 prompted items and questions and were distributed to people from all over the French speaking part of Switzerland. Among other items, each speaker was invited to:

- Read 10 sentences selected from different corpora to ensure good phonetic coverage for the resulting database.
- Simulate a spontaneous query to telephone directory, given the name and the city of subject.

Different kinds of irregularities (i.e; noise in the recording, strange utterances) were discovered, and the training set was finally reduced to 3, 272 sentences, corresponding to approximately 5 hours of speech.

2.7.2 PolyVar database

For capturing intra-speaker variability and to address inter-speaker variability issues, the *PolyVar* telephone database was also designed and recorded at IDIAP as a complement to the Swiss French *PolyPhone* database. It is particularly relevant for speaker verification research. This database comprises telephone recordings from 143 speakers (85 male speakers and 58 female speakers). Each speaker recorded between 1 and 229 sessions. Several speakers pronounced among other items the same set of 17 touristic application words (InfoMartigny) several times. This makes the database particularly well suited to test user-customized speaker verification systems, i.e., by:

- Assigning each of the words to one specific customer, thus
- Providing enrollment utterances of that word, and test utterances, as well as many impostor utterances pronouncing the right password.
- Providing several utterances associated with words different than the chosen password, from both the customer and potential impostors.

2.7.3 Experimental protocols

A set of 38 speakers (24 males and 14 females) who have more than 26 sessions were selected. The set of 17 words (see appendix A) is divided into *word1* and *word2* with 14 and 3 words, respectively.

The enrollment step consists of creating the customer-dependent model for each speaker and each word in *word1* using the first 5 utterances corresponding to the first 5 sessions. For testing, we have defined two protocols, protocol $P1$ and protocol $P2$.

1. *Protocol $P1$:*

   In this protocol, all possible situations are considered and it is defined as follows:

   - Between 18 and 22 utterances of the same word are used as a customer accesses with the expected password.
   - Each customer has a subset of 18 speakers as impostors (11 males and 7 females if the customer is a male and 6 females and 12 males if the customer is a female).
   - Each impostor has two accesses with the expected password.
   - Each customer and each impostor has 3 accesses with 3 different words taken from *data2* to simulate the case where speaker pronounces wrong password.

   This protocol will be referred to as first protocol $P1$ which is summarized in Table 4.1.

TYPE OF ACCESS	NUMBER OF ACCESSES
TRAINING	5
TESTING: C-EP	18 − 22
TESTING: C-IP	3
TESTING: I-EP	36
TESTING: I-IP	54

Table 2.1. *Number (for each speaker and each password) of customer (C) and impostor (I) accesses with expected password (EP) and invalid passwords (IP)*

2. *Protocol $P2$:*

   To evaluate our approaches on more difficult conditions, a subset of the protocol $P1$ where customers and impostors test accesses are made only with the expected password is defined. This results in 10,632 customer accesses and 19,152 impostor accesses.

2.8 Conclusion

This chapter has presented an introduction to the speech and speaker recognition systems with more emphasis on speaker recognition systems. Typically, they both use the same low-level acoustic features based on short term spectral analysis to model the speech content and the speaker voice characteristics, respectively.

The components, functional modes and performance evaluation of a speaker recognition systems are described. For operational speaker verification systems, some important factors such as the decision threshold or operating point have to be determined and fixed according to the application requirements.

Having extracted the speaker specific acoustic parameters, the goal is to create a speaker-dependent model that best captures the speaker characteristics as well as the lexical content of the password in text-dependent speaker verification systems. Modeling this speaker specific information will be the topic of the next chapter.

Chapter 3

Speaker Acoustic Modeling

3.1 Introduction

The first step in building a speaker verification system is to create for each new client an acoustic model that best characterizes his/her voice, using some enrollment data. This step is known as *enrollment step*. One of the most important factors that can affect the robustness of a speaker verification system is the *quality* of the client model, which is highly dependent on the amount and the quality of training data available for that client.

In practice, the amount of training data is usually very limited. Hence, we are unable to create a robust client model, i.e., estimate the parameters of the model reliably. To counter this problem, speaker verification systems use *a priori* acoustic knowledge extracted from a large database (many speakers) together with acoustic knowledge extracted from the client training data to create a client acoustic model that has good generalization properties on the unseen data. In speech/speaker recognition literature this is known as *speaker adaptation*. Some speaker adaptation techniques will be described in this chapter.

The quality of the recorded speech depends on the recording conditions such as the background noise and the quality of the microphone used for speech acquisition. A mismatch between recording and testing conditions can cause significant degradation in the system performance. To reduce the effect of these aspects, compensation techniques are introduced. These techniques can be applied at feature extraction level such as cepstral mean subtraction or at the scoring level such as score normalization. Some score normalization techniques will be discussed in this thesis.

Speaker acoustic modeling approaches can be divided into generative and discriminative models. In generative models such as Gaussian mixture models or hidden Markov models, the client model is trained to maximize the likelihood of the data provided by the client without taking into account the impostor's information. In discriminative models such as and artificial neural networks, the client model is trained to minimize the classification error between client and impostors, hence they need some impostor data to create the client model.

To increase the discriminative capabilities of generative models, some discriminative criteria have been proposed. The most widely used criteria in speaker recognition are *minimum classification error* MCE (Chou *et al.*, 1995; Rosenberg *et al.*, 1998; Siohan *et al.*, 1998) and the *maximum mutual information* MMI (Bahl *et al.*, 1986). These criteria will not be described here as they are beyond the scope of the thesis. However, an approach that make generative models discriminative will be described. In this approach, ANN models are used to estimated phone posterior probabilities instead of phone likelihood estimated by a GMM.

This chapter will discuss the most well known generative and discriminative models used in our work.

3.2 Generative models

3.2.1 Gaussian mixture models

A Gaussian mixture model (GMM) is a parametric probability density function comprising the weighted linear combination of several unimodal Gaussian probability density function that is used to estimate the likelihood of the data. Given a GMM model parameterized with a set of parameter Λ, the likelihood of an observation vector x_t is estimated as follows:

$$p(x_t|\Lambda) = \sum_{i=1}^{M} w_i p_i(x_t) \tag{3.1}$$

where $\Lambda = \{w_i, \Sigma_i, \mu_i\}$, M is the number of Gaussian member components and, w_i is the weight of the i^{th} Gaussian component subject to constraints

$$0 \leq w_i \leq 1 \qquad and \qquad \sum_{i=1}^{M} w_i = 1$$

$p_i(x_t)$ is the likelihood of the observation vector x_n given the i^{th} component and is estimated as follows:

$$p_i(x_t) = N(\Sigma_i, \mu_i, x_t) = \frac{1}{(2\pi)^{(d/2)} \mid \Sigma \mid^{(1/2)}} exp\left[-\frac{1}{2}(x_t - \mu_i)^T \Sigma_i^{-1}(x_t - \mu_i)\right] \tag{3.2}$$

where d is the dimensionality of x_t and μ_i and Σ_i are, respectively, the mean vector and the covariance matrix of the i^{th} Gaussian. Because the cepstral feature vectors (LPCC or MFCC) are generally uncorrelated, diagonal covariance matrix are practically used. In this case $p_i(x_t)$ is estimated as follows:

$$p_i(x_t) = N(\Sigma_i, \mu_i, x_t) \approx \Pi_{i=1}^{d} \frac{1}{(2\pi)^{(d/2)}\sigma_i} exp\left[-\frac{1}{2}\left(\frac{x_{t,i} - \mu_i}{\sigma_i}\right)^2\right] \tag{3.3}$$

Given a sequence X of T observation vectors that are independent and identically distributed (i.i.d), the *log likelihood* of X is estimated as follows:

$$\log p(X|\Lambda) = \sum_{t=1}^{T} \log p(x_t|\Lambda) \tag{3.4}$$

The set of parameters Λ can be estimated according to the Maximum Likelihood (ML) using the iterative Expectation-Maximization (EM) algorithm (Dempster *et al.*, 1977). This algorithm maximizes iteratively the likelihood function for all the set of observation vectors conditioned on the set of parameters Λ. That is, for two iterations, k and $k+1$, $p(X|\Lambda^k) \leq p(X|\Lambda^{k+1})$.

GMMs are successfully applied to text-independent speaker recognition systems (Gish and Schmidt, 1994; Reynolds and Rose, 1995; Reynolds, 1995). In these systems each speaker, S_c, is represented by a GMM with sufficient number of Gaussian mixture components M trained with features extracted from the speech data provided by the speaker.

1. *Speaker identification task:*
In speaker identification task, the goal is to select from a set of registered speakers, the one whose model gives the maximum *posteriori probability*:

$$\widehat{S}_c = \underset{S_c}{\operatorname{argmax}} \; P(S_c|X) \tag{3.5}$$

Using Bayes rule with the assumption that the *a priori* probabilities $P(S_c)$ are equal for all speakers, and using logarithms, the decision rule (3.5) can be expressed as a maximum likelihood decision rule:

$$\widehat{S}_c = \underset{S_c}{\operatorname{argmax}} \; \log p(X|S_c) \tag{3.6}$$

The identified speaker is the one whose model gives the highest *likelihood*.

2. *Speaker verification task:*
In speaker verification, we are interested in estimating $P(S_c|X)$, the posteriori probability that a speaker S_c has pronounced the utterance X. This probability will be compared to $P(\overline{S}_c|X)$, the posteriori probability of the speaker being anyone except the true speaker S_c.

In speaker verification, we make an hypothesis that the speaker S_c is the true speaker who has pronounced the utterance X if:

$$P(S_c|X) \geq P(\overline{S}_c|X) \tag{3.7}$$

where $\overline{S}_c$ is the *anti-speaker* model parameters. Using Bayes theorem (3.7) can be rewritten as follows::

$$\frac{p(X|S_c)P(S_c)}{P(X)} \geq \frac{p(X|\overline{S}_c)P(\overline{S}_c)}{P(X)} \tag{3.8}$$

Which leads to the *log likelihood ratio* decision rule:

$$llr(X) = \log p(X|S_c) - \log p(X|\overline{S}_c) \geq \log \frac{P(\overline{S}_c)}{P(S_c)} = \delta_c \tag{3.9}$$

where δ_c is a speaker dependent threshold. As we can see, the $llr(X)$ does not depend on the client model S_c, but also on the *anti-speaker* model $\overline{S}_c$. The design and the choice of this model is crucial to improve the performance of a speaker verification system. This will be discussed in more detail in Section 3.2.4.

The most important advantages of using GMM models for text-independent speaker recognition are the following:

- They are not sensitive to the temporal information of the utterance, they model only the underlying distribution of acoustic observations from a speaker. Each individual component of the GMM can be interpreted to represent some broad phonetic classes.

- As we will see, GMM based speaker recognition can be combined with a speech recognition to perform text-dependent speaker verification or to recognize simultaneously the pronounced text and the identity of the speaker.

3.2.2 Hidden Markov models

Hidden Markov models (Rabiner, 1989) can be defined as a double stochastic process, described by an underlying Markov chain producing observations that are themselves probabilistic functions of the visited state. The first component (i.e., the state sequence) is hidden but it can be observed through the stochastic process generating the observation sequence.

The main characteristic of HMM models that makes them suitable for use in speech/speaker recognition is their abilities to model the temporal information in the speech signal. The speech signal which is considered as a discrete stationary process and composed of a sequence of subword units (like phonemes) is modeled by a finite number of K states $Q = \{q_1, q_2, ..., q_K\}$.

As illustrated in Figure 3.1, the temporal variations is modeled by a set of stationary (i.e., independent on the time) transition probabilities $(a_{ij} = P(q_j^t|q_i^{t-1}) = P(q_j|q_i))$. To each state q_i in the HMM, a state probability distribution is associated to estimate the emission probability $b_i(x_t)$ of an observation vector x_t at time n. The most state distribution employed with HMM is GMM. In this case $b_i(x_t) = p(x_t|q_i^t)$, representing the likelihood of the vector x_t given the state q_i at time t and it is estimated using (3.2). The initial states (i.e., at time $n = 0$) are specified by the initial set of state probabilities $\{\pi_i\}$ $\left(\pi_i = P(q_i^0)\right)$. The set $\left(\{\pi_i\}_{i=1}^K, \{a_{ij}\}_{i=1,j=1}^K, \{b_i\}_{i=1}^K\right)$ are the parameters of the HMM model.

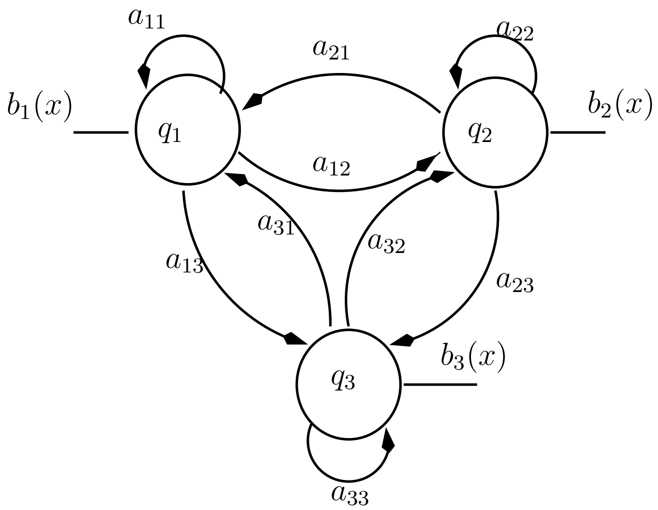

Figure 3.1. *A 3-states ergodic HMM model*

In HMM-based speech recognizer, the training step consists of determining the parameters of the HMM that maximize the global likelihood of the enrollment data. This is usually performed by applying several iterations of the EM algorithm. During recognition, the likelihood of the test utterance is estimated using Viterbi decoding (or alignment), which consists of finding, among all possible state sequences, the best one that has the highest likelihood. If states are associated with phonemes, the generated best sequence will correspond to the phonetic segmentation of the test utterance. To improve the recognition performance, in practice, for each phoneme, a minimum duration of, typically 3 frames is imposed. This is done by modeling each phoneme as a strictly left-to-right 3-states HMM.

For a speaker identification/verification task, HMMs are used in the same way as GMMs, i.e., an HMM is created for each client S_c. During the test, a maximum likelihood (in speaker identification task) or likelihood ratio (in speaker verification task) based decision is applied.

In text-independent SV, the temporal information of the speech signal is not important. So, an ergodic HMM model where all states are fully connected between them have been used. Comparison experiments with other text-independent SV modeling approaches such as vector quantization (Matsui and Furui, 1992a) and GMM (Przybocki and Martin, 1998; Auckenthaler *et al.*, 1999) showed that ergodic HMMs are less robust. Further investigation for speaker identification task, showed that the recognition rates are correlated with the number of mixtures irrespectively with the number of states. Indicating that the transition probabilities in text-independent speaker recognition task are ineffective (Auckenthaler *et al.*, 1999).

However, in text-dependent and text-prompted speaker recognition systems, the temporal information is important. To capture this information, some constraint should be made on the topology of the HMM model. Therefore, left-to-right HMM models also known as Bakis model (Bakis, 1976) are often used. In this model, only, the transition from left to right are allowed reflecting the speech production.

3.2.3 Speaker adaptation

The best way to cope with the individual speaker acoustic properties and hence to improve the robustness of a speaker recognition system is to train from scratch a speaker-dependent model with data provided by the speaker himself. Because of the high variability in the speech signal, this requires a large amount of training data, so, the parameters can be estimated reliably. Unfortunately, in practice, the amount of training data is very limited and we have to exploit it to create a speaker model whose performance is close to the speaker-dependent model. To circumvent this problem, a *model-based adaptation* techniques are proposed. The goal of speaker adaptation, is to adjust the parameters of a speaker-independent acoustic model to match the current speaker as closely as possible using the available adaptation data (Hazen, 1998). The use of *a priori* knowledge about the statistical properties of the parameters to be adapted make the adaptation process very fast. The adaptation technique as well as the number of parameters to be adapted depend on the amount of training data available. In speaker recognition literature, the most popular adaptation techniques are known as *maximum likelihood linear regression* (Leggetter and Woodland, 1995) and *maximum a posteriori* (Gauvain and Lee, 1994) adaptation techniques. Many variants of these techniques are also proposed. A comparative studies between these techniques can be found in (Ahn *et al.*, 2000) for text-dependent speaker verification and in (Mariéthoz and Bengio, 2002) for text-independent speaker verification. In this section, a description of maximum a posteriori and maximum likelihood linear regression is given.

Maximum a posteriori

Given observation data, X, the *Maximum A Posterior* (MAP) adaptation technique consists of finding the set of parameters $\widehat{\Lambda}$ that maximizes $P(\Lambda|X)$:

$$\widehat{\Lambda} = \underset{\Lambda}{\operatorname{argmax}} \; P(\Lambda|X) \tag{3.10}$$

where $P(\Lambda|X)$ is the posterior probability of the model parameters given the data, which can be rewritten according to Bayes rule as follows:

$$P(\Lambda|X) = \frac{p(X|\Lambda)P(\Lambda)}{P(X)} \tag{3.11}$$

where $p(X|\Lambda)$ is the likelihood of the observation data, $P(\Lambda)$ is the priori probability of the set of parameter Λ and $P(X)$ is the priori probability of the data which is independent of the model.

Hence, (3.10) can be rewritten as:

$$\widehat{\Lambda} = \underset{\Lambda}{\operatorname{argmax}}\ p(X|\Lambda)P(\Lambda) \tag{3.12}$$

If no *a priori* knowledge about Λ is available, $P(\Lambda)$ is assumed to be uniform distribution, and (3.12) reduces to the *maximum likelihood* formulation. The difference between ML and MAP training lies in the incorporation of the prior knowledge into the model. The MAP adaptation procedure is a two-step estimation like the EM algorithm. In the *E-step* we compute a *new* sufficient statistics of the adaptation data. In the *M-step* these statistics are combined with *old* statistics from the prior distribution. The outline of the MAP adaptation procedure for a GMM is as follows (Bimbot *et al.*, 2004):

Given a GMM with M mixture components parameterized with w, μ, σ and an observation sequence X, we compute for each mixture component i in the GMM:

$$P(i|x_t) = \frac{w_i.p_i(x_t)}{\sum_{j=1}^{M} w_j.p_j(x_t)} \tag{3.13}$$

where $p_i(x_t)$ is the likelihood of the observation x_t given the i^{th} mixture components and is estimated using (3.2). The "new" sufficient statistics for the weight of mixture i is then defined as follows:

$$n_i = \sum_{t=1}^{T} P(i|x_t) \tag{3.14}$$

where T is the number of frames in the sequence X, and n_i is the occupation likelihood of the observation data X.

Using n_i, $P(i|x_t)$ and x_t, the mean and variance of the adaptation data are then estimated as follows:

$$E_i(x_t) = \frac{1}{n_i}\sum_{t=1}^{T} P(i|x_t)x_t \tag{3.15}$$

$$E_i(x_t^2) = \frac{1}{n_i}\sum_{t=1}^{T} P(i|x_t)x_t^2 \tag{3.16}$$

The estimation of the adapted parameters of the i^{th} mixture, $(\hat{w}_i, \hat{\mu}_i, \hat{\sigma}_i)$ is a combination of the new statistics $n_i, E_i(x_t), E_i(x_t^2)$ and the old statistics w_i, μ_i, σ_i (from the *a priori* distribution), and they are estimated as follows:

$$\hat{w}_i = \gamma[\alpha_i \frac{n_i}{N} + (1-\alpha_i)w_i] \tag{3.17}$$

$$\hat{\mu}_i = \alpha_i E_i(x_n) + (1-\alpha_i)\mu_i \tag{3.18}$$

$$\hat{\sigma}_i^2 = [\alpha_i E_i(x_t^2) + (1-\alpha_i)(\mu_i^2 + \sigma_i^2)] - \hat{\mu}_i^2 \tag{3.19}$$

The scale factor γ is computed over all mixture weights to ensure that they sum to 1. The adaptation factor α_i is defined as follows

$$\alpha_i = \frac{n_i}{n_i + r} \tag{3.20}$$

where r is the relevance factor and is used to control how much new data should influence the estimation of the new model's parameters.

Empirical results show that the best verification performance is generally obtained when only the mean vectors of Gaussians are adapted. A possible reason is that due to the limited amount of training data the variance parameters and the mixture weights may be underestimated.

Maximum likelihood linear regression

Maximum likelihood linear regression (MLLR) (Leggetter and Woodland, 1995) is a model based adaptation technique that estimates the parameters of a set of linear transformations used to update the parameters of a given distribution, like GMM or HMM, using a small amount of adaptation data. Like MAP adaptation, MLLR is usually applied to update the means only.

For a GMM, the new mean vector for a particular mixture component i is estimated as follows:

$$\hat{\mu} = W_i \xi_i \tag{3.21}$$

where W_i is an $d * (d+1)$ transformation matrix defined as:

$$\xi_i = [w, \mu_1, ..., \mu_d] \tag{3.22}$$

where w is the offset term of the regression. The parameters of W_i are estimated to maximize the likelihood of the adaptation data given the transformed model. The statistics used to estimate the transformation matrix are computed using a forward-backward alignment of the adaptation data. It is evident that having a separate transform W_i for each Gaussian, will increase the number of parameters to be estimated, therefore, an approach in which several Gaussians share the same transformation matrix is adopted. In this case, all the data seen by these Gaussians is used to estimate the transformation matrix. If the amount of adaptation data is very small, a single global transformation can be used. Before starting the adaptation process, a Gaussian clustering technique is performed to cluster the Gaussians into regression classes, that can be determined dynamically using regression class tree (Gales, 1996).

3.2.4 Score normalization

The operation of a speaker verification system, requires a decision to be made whether to accept or reject the identity claim. In a generative approach, this process consists of estimating $p(X|S_c)$, the likelihood of the test segment X given the client model S_c, and is compared to a (usually speaker-independent) threshold. Typically, the threshold is adjusted to satisfy the security level of the application. The estimated likelihood may vary considerably due to inter and intra speaker variability. This makes the use of raw likelihood $p(X|S_c)$ in real applications less effective and the estimated threshold less reliable even if the threshold is speaker-dependent. To circumvent this problem a *log likelihood ratio* based score normalization technique derived from the use of Bayes rule is proposed ((3.7), (3.8), (3.9)). The *log likelihood ratio* is estimated as follows:

$$llr(X) = \log p(X|S_c) - \log p(X|\overline{S}_c) \tag{3.23}$$

Using an "anti-speaker" model to represent $\overline{S}_c$, the normalization score $\log p(X|\overline{S}_c)$ will be affected in a similar manner to the client speaker model S_c, because the client speaker model is usually adapted from the "anti-speaker" model. Consequently, the likelihood ratio remains relatively unaffected, making the decision threshold more stable and reducing the need for a speaker-dependent threshold.

The choice of the "anti-speaker" model in a SV system is as important as the threshold determination. A poor choice of the "anti-speaker" model can significantly degrade the performance of the system. This model should be determined in such a way that the selectivity or the discriminant capability of the system against impostor accesses improves. There is no theoretical approach for "anti-speaker" model design, but empirical studies suggest that an anti-speaker model which is close to the client model is a reasonable choice (Rosenberg and Parthasarathy, 1996). We should mention here that the use of score normalization is useful in reducing the false acceptance rate.

There are three approaches commonly used to design the "anti-speaker" model, known as the *cohort* model for both text-dependent and text-independent, the *background* model for text-dependent and the *world* model or *universal background model* for text-independent speaker verification systems.

- In the *cohort* model (Rosenberg *et al.*, 1992), we assume that a set of speakers that are close to the client model are representative of all impostors. The log likelihood $\log p(X|\overline{S}_c)$ is approximated by the average of the log likelihood of the cohort models. The $llr(X)$ in (3.23) is then estimated as follows:

$$llr(X) \approx \log p(X|S_c) - \left[\frac{1}{N} \sum_{S_i \in C, S_i \neq S_c} \log p(X|S_i) \right] \tag{3.24}$$

  where N is the number of speakers in the cohort set C. The selection of the cohort can be done off-line during the enrollment of the speaker (Rosenberg and Parthasarathy, 1996; Isob and Takahashi, 1999) making the cohort selection speaker-dependent. This method has a serious problem, in that it is vulnerable to attack by acoustically dissimilar speakers. Therefore, the cohort should be large enough to cover the distribution of the general population. To overcome this problem,(Reynolds, 1995) has shown that a randomly selected (close and far) speakers to form the cohort outperforms this method. Another solution is to select the cohort model on-line (during the use of the system), making the cohort selection test-dependent. This is referred to as *unconstrained cohort* (Ariyaeeinia and Sivakuaran, 1997). Experiments have shown that on-line cohort selection outperforms the off-line cohort selection. An alternative method to chose the *cohort* is to keep the speaker model that has the maximum likelihood among all speakers model excluding the client model (Higgins *et al.*, 1991).

$$llr(X) \approx \log p(X|S_c) - \max_{S_i} \log p(X|S_i) \tag{3.25}$$

  In (Matsui and Furui, 1994), a *posteriori probability* based cohort model selection is proposed. In this technique, the client model is included in the cohort set. Experiments have shown no difference between the *posteriori probability* and *likelihood* based cohort selection techniques.

- In the *background* model, the $\log p(X|\overline{S}_c)$ is estimated using an acoustic model trained with data provided by other speakers whose selection might depend on the lexical content of the client's password in the case of text-dependent speaker verification. The background model is usually an HMM model that can be a whole phrase or a concatenation of speaker-independent phone models (Rosenberg and Parthasarathy, 1996).

- The world model or Universal Background Model (UBM), firstly, proposed by (Carey *et al.*, 1991) is a single model representing all the population. It is used in text-independent speaker verification. In these systems, the world model is a GMM whose order is usually between 512 and 2048 mixtures (Reynolds *et al.*, 2000; Bimbot *et al.*, 2004), depending on the application (constrained or unconstrained speech). Compared to the cohort models, the use of world or background model reduces the computational requirement of the system as the log likelihood $\log p(X|\overline{S}_c)$ is estimated using only one model.

3.3 Discriminative models

In generative approaches the parameters of the client model are estimated separately without taking into account the data from other speakers. This makes the *likelihood ratio* based decision suboptimal for classification and the performance of the speaker verification system depends closely on the design of the "anti-speaker" model (Rosenberg *et al.*, 1998). Discriminative approaches do not suffer from this problem by training to learn the boundaries between clients and impostors to minimize the classification/verification error. This requires the availability of client and impostor data to learn the decision boundaries.

Multi-layer perceptron

Artificial Neural Networks (ANN) are a parametric discriminative classifier. They are used in various classification tasks including speaker recognition. An overview on the use of ANN for speaker recognition can be found in (Bennani and Gallinari, 1995). ANNs have shown good performance, comparable to the use of GMM or HMM models. Although different ANN models have been used for speaker recognition, such as predictive neural network (Hattori, 1994), Recurrent Neural Networks (RNN) (Rudasi and Zahorian, 1990), the use of Multi-Layer Perceptron (MLP) still remains the most common model for this task.

An MLP is a feed-forward connection network composed of several layers (input layer, hidden layers and output layer) of nodes. for speech/speaker recognition, a 3-layers MLP is generally used. Such an MLP can be viewed as a universal approximation function. Each node (except input nodes) computes a linear weighted sum over its input nodes. A nonlinear transfer function (generally a sigmoid or a tanh function) is then applied to compute the output of that node. It has been shown that for a classification task, an MLP can be trained to estimate the posterior probabilities of classes given an input acoustic vectors.

Depending on the task (speaker identification/verification), MLPs can be used in different ways. For closed-set speaker identification (Rudasi and Zahorian, 1991), the common approach is to use a 3-layer MLP with the number of outputs equal to the number of registered speakers. Each output corresponds to one speaker. The MLP is trained to give a posterior probability of one to the output corresponding to the speaker from whom the speech data belongs (target speaker) and zero for the other outputs. During testing, the MLP outputs estimate the *posteriori probability* that the input vector belongs to each speaker. The speaker with the highest *posteriori probability* will be selected. The main limitations of this modeling approach is that the training time increases considerably as the number of speakers increases and the addition of a new speaker requires the training of the entire MLP.

For a speaker verification task usually one 3-layer MLP is used for each speaker. It has two outputs, one for the client and one for the other speakers (impostors) and is trained in the same manner as the speaker identification MLP. The problem with such approach is that: (1) if the impostor training data is not representative of the impostor testing data, the performance of the MLP will be poor, and (2) there should be a way to take into account the amount of training data available for each class (*a priori information*), otherwise the MLP decision will be biased to the class that has occurred more often during training.

3.4 Hybrid HMM/MLP systems

A hybrid HMM/MLP system combines the advantages of the hidden Markov model and artificial neural network. In this system, an MLP is used to estimate the HMM state posterior probabilities instead of using Gaussian mixtures, while the HMM is used to model the temporal dynamics of the speech signal. Some advantages of using the hybrid HMM/MLP systems are listed bellow:

1. The assumption that the emission probabilities can be modeled by a GMM might not be true in practice. In MLP there is no such strong assumption. The MLP can model any arbitrary probability distribution.

2. Under certain conditions (as discussed below), the outputs of the MLP trained with standard back-propagation can be interpreted as *a posteriori probabilities*.

3. It is easy to use the temporal information (context) by providing several acoustic vectors to the inputs of the MLP, easily incorporating the correlation between successive frames.

The discussion in this section is based on that of (Richard and Lippman, 1991; Bourlard and Morgan, 1994; Renals *et al.*, 1994; Gold and Morgan, 2000).

3.4.1 MLP as posteriori probabilities estimator

The outputs of an MLP can be interpreted as estimates of posterior probabilities of output classes conditioned on the input if certain conditions for the MLP training are taken into account:

1. The MLP should be trained in the classification mode, that is, for K classes, the target is one for the correct class and zero for all the others.

2. The MLP must be sufficiently large (i.e, contains enough parameters)

3. The MLP must be trained to a global minimum of the error function. In practice, this is hard to achieve, however, empirical results showed that even with local minimum, a good estimates of the posterior probabilities can still be achieved.

4. The cost function (error function) is either the Mean Squared Error (MSE) or the cross-entropy between the outputs and targets. Short proofs can be found in (Richard and Lippman, 1991) which demonstrate that when an MLP is trained for a classification task, these two cost functions are minimized when MLP outputs are posterior probabilities.

3.4.2 Hybrid HMM/MLP for speech recognition

Hybrid HMM/MLP systems have been successfully used in many automatic speech recognition applications including large vocabulary speaker-independent speech recognition. In state-of-the-art HMM/MLP speech recognizer, a 3-layer MLP is used. A temporal context of 9 consecutive acoustic vectors are fed to the input layer of the MLP. A sigmoid activation function is often used in the hidden layer with a large number of nodes. The nodes in the output layer correspond to the HMM states and are often associated with context-independent acoustic classes such as phones. The first step in building a hybrid HMM/MLP system is to train the parameters of the MLP.

MLP training

MLP training consists of estimating the set of parameters Θ (weights and biases) that minimizes a cost function E given a set of training examples. For a speech recognition task, the cost function commonly used is either the MSE or the cross-entropy. In this thesis, our MLP is trained using cross-entropy as it has been shown to lead to better performance than MSE (Bourlard and Morgan, 1994). The average cross-entropy is defined as follows:

$$E = \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{K} d_k(x_t)\log\frac{d_k(x_t)}{g_k(x_t,\Theta)} \tag{3.26}$$

where

- K is the number of outputs and corresponds to the number of phones.

- x_t is the acoustic vector at time t and T is the total number of acoustic vectors in the training data.
- $d_k(x_t)$ is the desired value of the k^{th} output given x_t as input. This value is known for each output and for each training example x_t and it is defined as follows:

$$d_k(x_t) = \begin{cases} 1 & \text{if } x_n \text{ belongs to the } k^{th} \text{ phone} \\ 0 & \text{otherwise} \end{cases}$$

- $g_k(x_t, \Theta)$ is the observed (estimated) value of the k^{th} output, given x_t and the set of MLP parameters Θ.

An iterative gradient descent algorithm (back-propagation) is used to evaluate the partial derivatives of the cost function with respect to each parameter (Rumelhart *et al.*, 1986):

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \tag{3.27}$$

where w_{ij} is the weight associated to the connection between the nodes i and j, and η is the learning rate. Finally the weights are adjusted as follows:

$$w_{ij}^{new} = w_{ij}^{old} + \Delta w_{ij}^{new} \tag{3.28}$$

where "old" and "new" denotes values before and after the adjustment.

Characteristics of MLP training procedure

The training step includes a number of choices that can affect the modeling capability of the MLP. Some of these choices are: the architecture of the MLP (the number of nodes in the hidden layer), the training method (stochastic or batch),the cost function or training criterion (MSE or cross-entropy), and stopping criterion (number of iterations or cross-validation). In this section, the characteristics of the training procedure used to train our MLP are described.

1. *Supervised training:* This means that the desired output (target phone class) should be known for each input frame x_n. This requires an initial labeling or segmentation (linear segmentation for example) of the training data.
2. *Stochastic training:* The adjustment using (3.28) of the parameters Θ is done after every training example, contrary to the batch training where the adjustment is done after passing all the training examples. The stochastic training has the advantages that converges much faster than batch training and results in better performance.
3. *Cross-validation:* When we train an MLP for speech recognition task, our aim is not to minimize the cost function on the training data which can lead to the *over-training* (i.e., the MLP is over-fitted to the training data). Our aim is to maximize the MLP's ability to generalize. That is, the MLP performs well even for new unseen data. It is desirable to stop training when the generalization ability starts to degrade. In our MLP training procedure we have used a cross-validation technique. The data is split into a training set on which the parameters are trained and a validation set on which the generalization abilities are evaluated. As described in (Bourlard and Morgan, 1994), after each iteration (presentation of all train data) we test the frame level performance of the resulting MLP on the cross-validation set and continue training only while the performance on the cross-validation set improves.

4. *Learning rate:* The learning rate η in (3.27) determine the stepsize or the speed of the learning process. if η is small then the learning process will take a long time to converge, and if it is large, the learning process might diverge from the solution. In this thesis, an adaptive learning rate in combination with cross-validation is used. As in (Bourlard and Morgan, 1994), we start with a learning rate, η, equal to 0.1. Each time the frame level performance on the cross-validation data degraded, we divide this learning rate by a factor 2 for the next iteration. This process is iterated until the learning rate is below 0.0001, at which point training process is considered to be complete.

5. *Embedded training:* Once the MLP is trained, a forced Viterbi alignment using the MLP output estimates on the train data is performed to generate a new segmentation. This segmentation is then used to retrain the MLP as described in the previous points. Experiments for a speech recognition task, showed that this procedure can improve the performance of the neural network (Bourlard and Morgan, 1994).

Recognition

During recognition the values of the MLP outputs estimate the phone posterior probabilities $P(q_k|x_t)$ for each state q_k conditioned on the acoustic vector x_t given as input. These probabilities should satisfy:

$$0 \leq P(q_k|x_t) \leq 1 \qquad and \qquad \sum_{k=1}^{K} P(q_k|x_t) = 1 \tag{3.29}$$

To ensure (3.29), a *softmax* output function (Bridle, 1990) is used instead of sigmoid function, and defined as:

$$g_k(x_t) = \frac{exp(f_k(x_t)}{\sum_{\ell=1}^{K} exp(f_\ell(x_t))} \tag{3.30}$$

where $g_k(x_t)$ is an estimate of $P(q_k|x_t)$ and $f_k(x_t)$ is the value of the output unit associated with the phone q_k prior to the nonlinearity for an input vector x_t.

These posteriori probabilities can be used directly for recognition, they are usually converted to *scaled likelihoods* using Bayes rule:

$$\frac{P(q_k|x_t)}{P(q_k)} = \frac{p(x_t|q_k)}{P(x_t)} \tag{3.31}$$

where $P(q_k)$ is the a *priori* probability of phoneme q_k estimated from relative frequencies in the training data. The *scaled likelihood*, $\frac{p(x_t|q_k)}{P(x_t)}$, is estimated from the posteriori probability divided by the prior. As shown in figure (3.2), these scaled likelihoods are used as emission probabilities for HMM states. Viterbi decoding with combination with the transition probabilities is then performed to find the most likely HMM state sequence that could have generated the sequence of acoustic vectors. It is worth mentioning here, that the performance of the hybrid HMM/MLP systems is much more dependent on the emission probabilities than the transition probabilities.

3.4.3 Speaker MLP adaptation

MLP adaptation techniques can be divided into retraining the speaker-independent ANN and training a transformation network usually a Linear Input Network (LIN).

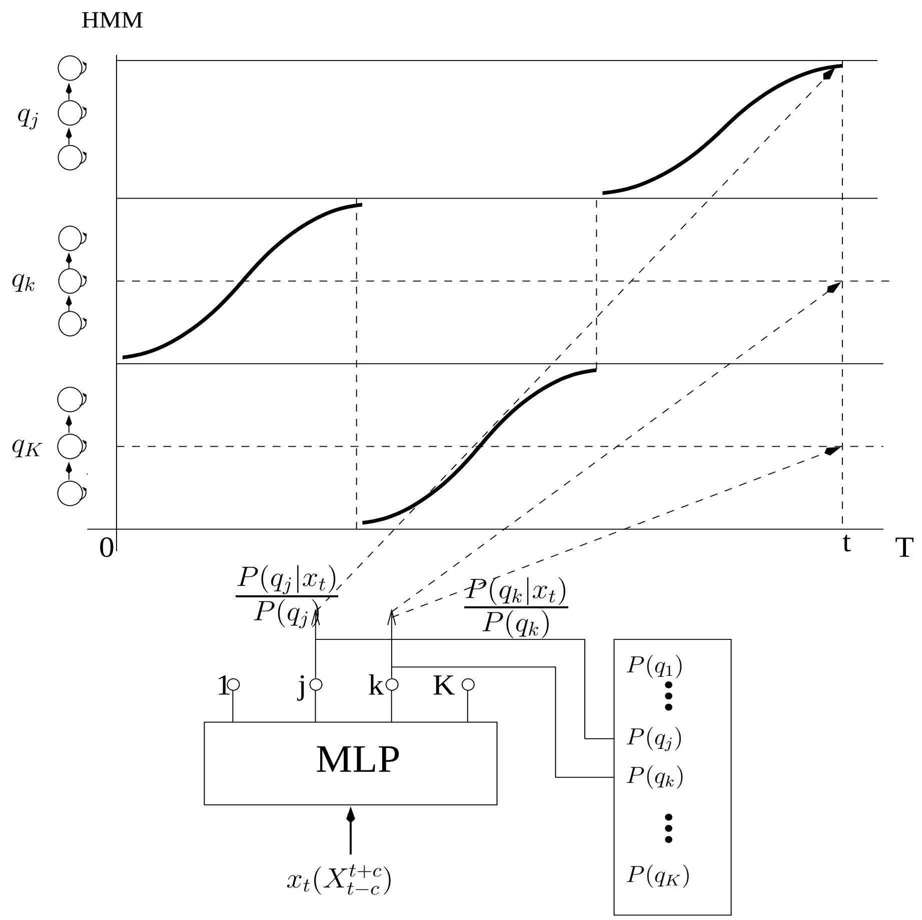

Figure 3.2. *Block-diagram for speech recognition uses a hybrid HMM/ANN system: for each frame x_t (with c frames on the left and right), the MLP estimates the posteriori probabilities $P(q_k^t|x_t)$ for each state q_k, which will be divided by the priori $P(q_k)$ to obtain the scaled likelihood. These scaled likelihoods are then used as HMM states emission probabilities for Viterbi decoding.*

Retrained speaker-independent ANN

In this approach, the speaker adaptation data is used to retrain all the parameters of the speaker-independent MLP. The advantage of this technique is that the initial condition is good (i.e., the weights are initialized to a good values) which is important for MLP training and adaptation. However, given the large number of parameters to be adapted and the small amount of adaptation data, generalization of the resulting MLP on unseen data can not be guaranteed and is difficult to achieve.

Linear input network

In this approach, a trainable speaker-dependent transformation layer is added to the input of the speaker-independent MLP (SI-MLP) to map the speaker specific acoustic space to the speaker-independent acoustic space[1] (Neto *et al.*, 1995; Abrash *et al.*, 1995). In this manner, the ability of the MLP to estimate the *posteriori* class probabilities is enhanced. For each acoustic vector x_n in the adaptation data, a new acoustic vector $\hat{x}_t$ is computed using an input linear transformation:

$$\hat{x}_t = W x_t + b \tag{3.32}$$

where W and b are the weights and the bias parameters, respectively, to be adapted. In practice, the bias vector b is often set to zero. The frame, $\hat{x}_t$, now corresponds to the input vector of the SI-MLP. The parameters of the additional layer are trained to minimize the cross-entropy error at the output of the SI-MLP whose parameters remain frozen. Only the parameters of LIN are updated. Before adaptation, the LIN is initialized to the identity matrix I ($W = I$). This guarantees that the initial condition is the SI-MLP, that is, the performance of the adapted MLP is at least equivalent to the SI-MLP. The major advantage of this approach is the significant reduction in the number of parameters to be adapted. This reduction can be more important by defining different connection architectures between the LIN layer and the input layer of the SI-MLP. Some of these architectures will be described in Chapter 5. Since the additional layer is linear and $b = 0$, those adaptation parameters can be included in the main network after training, by multiplying the adaptation matrix with the input-hidden matrix.

An extension to this technique known as Mixtures of Transformation Networks (MTN) is presented in (Abrash, 1997). Instead of using a global transformation, the acoustic features are divided into several regions and an input transformation network or LIN is adapted for each region. The transformations are then combined probabilistically with weights derived from a separate Acoustic Gating Network (AGN).

Both adaptation techniques can be performed in supervised or unsupervised paradigm. In supervised paradigm, the SI-MLP is used to estimate the emission probabilities, then, a forced Viterbi alignment on the phonetic transcription of the adaptation sentences is performed to obtain the initial segmentation that will be used as a desired outputs for MLP adaptation. In unsupervised adaptation the phonetic transcription is inferred from a decoding carried only by the speaker independent model. That is for each input vector x_n, the desired outputs is the one with the highest posterior probability conditioned on x_t

3.5 Speaker acoustic modeling in UCP-SV: related works

In this section, a description of published works related to UCP-SV are given. They are ordered according to their publication year.

[1]In (Abrash *et al.*, 1995), this approach is used to perform speaker normalization.

- **Speaker identification with user-selected password phrases** (Rosenberg and Parthasarathy, 1997): This paper describes an open set speaker identification system where customers are allowed to choose their own passwords. A speaker-independent speech recognizer is used to transcribe each utterance in a sequence of phonemes. These phonetic transcriptions constitute the customer password lexicon. The best inferred phonetic transcription is used to build the background model and to specify the number of states in the whole-password customer dependent HMM (the number of states is taken to be 1.5 times the number of phones in the best phonetic transcription).

  During identification, the speaker-independent speech recognizer scores the test utterance against the alternate phonetic transcriptions for each customer lexicon. The 5 highest scoring phonetic transcriptions are selected. For each of them, the score of the associated best phonetic transcription referred to as background score "B" as well as the score of the whole-password HMM model referred to as reference score "R" are reported. The sum of "B" and "R" scores is used to identify the customer and the difference between "R" and "B" scores is then compared with a customer dependent threshold to verify the identified customer. The verification performance is compared with a system where the phonetic transcription of the password is given by a dictionary (known). The authors found that the use of the sum score increased the performance of identification and the use of dictionary transcriptions to create the background model worked better than the use of the inferred best phonetic transcription.

- **Background model Design for flexible and portable speaker verification systems** (Siohan *et al.*, 1999): The goal of this paper is to build a flexible and portable speaker verification system. No databases and no speaker-independent speech recognizer are available. The issue addressed here is how to design a background model when only the customer enrollment data is available. Two different techniques are proposed. In the first technique, the enrollment data is used to train a background model with fewer number of states (5 states) compared to the customer model (25 states). In the second technique, the background model is derived by perturbing the temporal information of the previously trained background model in the first technique. This is achieved by reversing the states order. Results showed that the use of the normalized score improves the performance compared to the use of raw likelihood. But the these results were far from the performance that can be obtained when a speaker-independent background model is used.

- **Improved normalization without recourse to an impostor database for speaker verification** (Hebert and Peters, 2001): The goal of this paper is to improve the selectivity of the background model so the performance of the speaker verification system for casual impostor accesses will improve. In this work, the speaker-independent speech recognizer has 800 allophone models composed of 3 states, each with 5 GMM components. The best inferred allophonic transcription is used to create the background model. The mean parameters of this model are then adapted using MAP with a small *priori* weight (0.01) to create the customer dependent password model. A likelihood ratio based speaker verification score is used to make the decision.

  The examination of score distributions showed that using the speaker-independent background model is not optimal for casual/close impostor. The authors proposed the use of a modified normalizing model (MNM) to reflect the lexical content of the password. Several techniques were proposed. The best performance is obtained by the *unconstrained* MAP technique. In this case, first, a new model with mean parameters is generated using MAP adaptation with large weight on the *prior*. Then a single MNM model which is constructed by having in the same mixture all of the original means (speaker-independent background) and their (new) modified counterparts is created. The performance on the casual/close impostor was improved by 15% without degrading the performance on the other type of impostor accesses.

- **Password-dependent speaker verification using quantized acoustic trajectories** (Gagnon *et al.*, 2001): This paper describes a new single step speaker verification approach. That is the password itself will be used to identify and verify the speaker. This approach is based on GMMs and Quantized Acoustic Trajectories (QAT) and is text and language independent. In this approach, a GMM is used as a seed acoustic model (1) to determine a QAT which model the acoustic path of the user password and (2) as *a priori* distribution for MAP adaptation. The QAT is the most likely sequence of GMM components.

  During Verification, the QAT for the test utterance is obtained. A fast Dynamic Time Warping (DTW) based algorithm is used to select the best-matching set of customer QATs. The adapted GMMs corresponding to the best set are scored using the test utterance and the most likely model is selected. A log likelihood based score verification is then estimated and compared to a speaker-independent threshold. The results showed that the two-step verification (perfect speaker identification) performed significantly better.

3.6 Conclusion

This chapter has discussed the most common and most successful generative (Gaussian mixture models and hidden Markov models) and discriminative (neural network) models currently used for speaker recognition. The main difference between these two models lies in the training criterion. Generative models are trained to maximize the likelihood of the client data, i.e., estimating the set of parameters that best describes the speaker characteristics. Discriminative models are trained to minimize the classification error between client and impostors, i.e; estimating the set of parameters that provides better discrimination between client and impostors. To minimize the classification error of generative models, score normalization techniques using an “anti-speaker” model are introduced. The design and the selection of the “anti-speaker” model can be as important as the training of the client model.

This chapter has also described hybrid HMM/MLP modeling approach where the outputs of an MLP are used to estimate HMM states posterior probabilities instead of GMMs. As we will show later, while HMM/MLP systems provides interesting properties, their use in speaker recognition may result in non-obvious difficulties.

Chapter 4

HMM/GMM based UCP-SV

4.1 Introduction

This chapter discusses and optimizes an HMM-based UCP-SV system. The main assumption of the UCP-SV systems is that no *a priori* knowledge about the password is available. The development of such systems within HMM/GMM framework raises some practical issues, including: HMM inference, speaker adaptation and score normalization. We have compared two speaker-independent speech recognizers to infer the phonetic transcriptions associated with the repetitions of the customer's password, i.e., HMM/GMM system and hybrid HMM/MLP system. Both systems are trained using the same training data and the same initial segmentation. In this chapter, the parameters of the inferred HMM as well as their adaptation will use GMMs.

As we have seen in Section 3.4, the MLP has several advantages, including : powerful (non-linear) modeling capabilities, possibility of capturing time correlation (by including the input of the MLP with contextual inputs), discriminant training, and estimation of posterior probabilities. These advantages make the hybrid HMM/MLP particularly good at recognizing phonetic strings which has been shown to be efficient, e.g., in a voice dialing system (Fontain and Bourlard, 1997).

This chapter starts with the description of the Bayesian framework of the decision rules in TD-SV systems, which are also valid for UCP-SV systems.

For comparison purposes, an HMM-based TD-SV system will first be described. This system uses the correct phonetic transcription of the password (i.e., given by a dictionary) to create the customer HMM model. This system will be used as a reference system.

Second, a baseline UCP-SV system will be described. This system uses the best inferred phonetic transcription of the password to create the customer HMM model. A comparison with the reference system shows that the performance of the baseline UCP-SV system is lower. The analysis of the results shows that the background model is the main problem.

Third, an optimized UCP-SV system will be investigated. This system uses multiple reference models for customer acoustic modeling and multiple background models for score normalization. For each inferred phonetic transcription, a customer HMM password model and a background model are created. The use of multiple reference models allows us to select the customer HMM model that best matches the test utterance, while the use of multiple background models allows us to select the background model that best competes with the selected customer model. To estimate the verification score, different techniques are compared such as dynamic model selection techniques, score fusion techniques and decision fusion techniques.

4.2 Decision rules

In TD-SV system and from the statistical hypothesis testing point of view, the decision to accept or reject a speaker involves two hypotheses tests: (1) hypothesis testing of the customer password where the null hypothesis is the pronounced word corresponds to the customer's password (expected password) and (2) hypothesis testing of speaker identity where the null hypothesis is the claimed identity corresponds to the customer.

Formally, we are interested in estimating $P(M_c, S_c|X)$ representing the joint posteriori probability that the customer S_c has pronounced the expected password M_c given the observed acoustic vector X. During verification, this probability is compared to (1) $P(M_c, \overline{S_c}|X)$, representing the joint posterior probability that any other speaker (impostor) $\overline{S_c}$ may have pronounced the expected password M_c, and (2) $P(\overline{M_c}, S|X)$, representing the joint posterior probability that any speaker S (i.e., impostor $\overline{S}_c$ or customer S_c) may have pronounced any other password $\overline{M_c}$. Hence, the decision rules can be formulated in two necessary conditions:

$$S = S_c \quad \text{if} \quad P(M_c, S_c|X) \geq P(M_c, \overline{S_c}|X) \tag{4.1}$$

$$\text{and} \quad P(M_c, S_c|X) \geq P(\overline{M_c}, S|X) \tag{4.2}$$

Using Bayes rule, and assuming that the joint probability of any speaker and any word is equal for all combinations of speakers and words, decision rules (4.1) and (4.2) can be rewritten as follows:

$$\frac{p(X|M_c, S_c)}{p(X|M_c, \overline{S}_c)} \geq \frac{P(M_c, \overline{S}_c)}{P(M_c, S_c)} = \Delta_1 \tag{4.3}$$

$$\frac{p(X|M_c, S_c)}{p(X|\overline{M}_c, S)} \geq \frac{P(\overline{M}_c, S)}{P(M_c, S_c)} = \Delta_2 \tag{4.4}$$

where Δ_1 and Δ_2 are the decision thresholds, respectively, for the correct speaker and the expected password. The normalization models $(M_c, \overline{S}_c)$ and $(\overline{M}_c, S)$ in (4.3) and (4.4) are used to estimate the normalization scores $p(X|M_c, \overline{S}_c)$ and $p(X|\overline{M}_c, S)$, respectively, and they have two different roles. The first normalization model is speaker competitiveness based. That is the speech content represented by this model is supposed to be close (or the same) to the speech content represented by the customer model. So, it is used to discriminate between the customer and impostors pronouncing the expected password. The decision using (4.3) will be referred to as *speaker verification* decision.

If the speech content of the test utterance is different from the expected password, both customer and score normalization models in (4.3) will have a poor individual likelihood which might result in a good likelihood ratio and lead to the acceptance of an impostor. A solution to this problem is to perform a speech recognition or utterance verification step to recognize or to verify that the pronounced word corresponds to the expected password. This is the role of the second normalization model $(\overline{M}_c, S)$ in (4.4). This model is supposed to represent the incorrect (invalid) password. The decision using (4.4) will be referred to as *utterance verification* decision.

Usually, the decision to accept or reject a speaker is made in two steps. First, we perform an utterance verification step, and if the score exceeds the threshold Δ_2, then we perform a speaker verification step. So, the speaker is accepted if the two scores exceed their respective thresholds simultaneously. Experimental results (Rodriguez-Linares *et al.*, 2003) have been shown that the combination of these two scores can significantly improve the performance of the system. In this paper, among different combination techniques, we have used a weighted sum combination technique. This will help us to analyze and give some possible explanation of the obtained results. The verification score VS on which the decision will be made is estimated as follows:

$$VS = \alpha \, LLR_s + (1 - \alpha) \, LLR_u \tag{4.5}$$

where $0 \leq \alpha \leq 1$ and LLR_s is the log likelihood ratio estimated using the *speaker verification* part as follows:

$$LLR_s = \frac{1}{T}\left[\log p(X|M_c, S_c) - \log p(X|M_c, \overline{S}_c)\right] \tag{4.6}$$

and LLR_u is the log likelihood ratio estimated using the *utterance verification* part as follows:

$$LLR_u = \frac{1}{T}\left[\log p(X|M_c, S_c) - \log p(X|\overline{M}_c, S)\right] \tag{4.7}$$

We use $\frac{1}{T}$ to normalize the two *log likelihood ratio* for test utterance duration and T is the length of the test utterance after having removed the silence frames.

An alternative way to estimate LLR_u is to use the speaker-independent HMM model (M_c, S) (Reynolds *et al.*, 2000), instead of speaker-dependent HMM model, yielding the following LLR_u estimation:

$$LLR_u = \frac{1}{T}\left[\log p(X|M_c, S) - \log p(X|\overline{M}_c, S)\right] \tag{4.8}$$

4.3 Methodology

Feature extraction

In all experiments carried out in this thesis, the acoustic feature vector consists of 12 MFCC coefficients with energy complemented by their first derivatives. These coefficients are calculated every 10 ms over 30 ms window, resulting in 26 coefficients.

System development

The different UCP-SV modeling approaches studied here, assume the availability of some *a priori* acoustic models, to perform HMM inference, speaker acoustic modeling and score normalization. Several speaker-independent acoustic models are trained using *PolyPhone* database (see Section 2.7).

1. *HMM inference:* For comparison purposes, two speaker-independent speech recognizers are trained.

   - *Hybrid HMM/MLP system:* The speaker independent MLP [1] of parameters Θ consists of 234 input units with 9 consecutive 26 dimensional acoustic vectors, 600 hidden units and 36 outputs, such that each output is associated with a specific phone.
   - *HMM/GMM system:* The speaker-independent HMM of parameters Ω consists of 36 context-independent phone models. The phone model consists of 3 states left-to-right HMM with 24 mixtures/state. Diagonal covariance matrix is used for each mixture. This HMM (Ω) is trained using segmental K-means algorithm (Rabiner, 1989) followed by several iterations of the EM algorithm.

   The phone level accuracy obtained by these two systems on *PolyPhone* (i.e., train and test data) and *PolyVar* (i.e., customer enrollment data) are reported in Table 4.1. Within HMM/MLP system, both posterior probability and scaled likelihood (3.31) are used. From Table 4.1, and confirming previous speech recognition experiments, it is clear that the hybrid HMM/MLP system yields better phone accuracy than HMM/GMM system.

[1] For more details about this MLP training procedure, see Section 3.4.2

SYSTEM	CRITERION	POLYPHONE TRAIN	POLYPHONE TEST	POLYVAR TRAIN
HMM/MLP	POSTERIOR	58.6%	57.5%	57.5%
	SCALED LIKE.	54.2%	49.4%	56.6%
HMM/GMM	LIKELIHOOD	49.8%	48.4%	55.9%

Table 4.1. *Phone accuracy obtained by the hybrid HMM/ MLP system (Θ) and HMM/ GMM system (Ω) on PolyPhone and PolyVar databases.*

2. *Speaker acoustic modeling:* An HMM of parameters λ is trained using the segmental K-means algorithm (Rabiner, 1989) followed by the EM algorithm. This HMM consists of 36 context-independent phone models. The phone model consists of 3 states left-to-right HMM with 3 mixtures/state. This HMM will be used as *a priori* distribution for *maximum a posteriori* adaptation to create the customer dependent password HMM.

3. *Score normalization for utterance verification:* A GMM of parameters Λ modeled by 240 (diagonal covariance) Gaussians is trained using the segmental K-means algorithm followed by EM algorithm. As we will explain in Section 4.4.1, this GMM will be used for utterance verification score normalization.

Performance evaluation

Normally, the performance should be evaluated using *a priori* threshold. A common way to do that is to divide the database into development and evaluation sets. A *posteriori* threshold will be estimated on the development set to optimize the EER or HTER. This threshold is then used as *a priori* threshold to evaluate the performance of the system on the evaluation set. In our case, both development and evaluation sets should contain different customer with different passwords and different test accesses with invalid password. It is clear that such a division reduces the test set and makes the confidence in the results less reasonable. For this reason, we have used *a posteriori* threshold determined to optimize the EER.

We should note here that in this thesis, all experiments are conducted using Torch library (Collobert *et al.*, 2002). Among other characteristics, Torch contains most of the state-of-the-art machine learning algorithms.

4.4 Text-dependent speaker verification (reference system)

For the sake of comparison, a reference TD-SV system was developed to diagnose and analyze the performance of the baseline UCP-SV system. In fact, our aim was to develop a UCP-SV system whose performance is as close as possible to that of the reference system under the same conditions.

4.4.1 Speaker enrollment

For each customer, a customer-dependent word HMM model is created in two steps:

- First, a speaker-independent word HMM (SI-HMM) model is created. This SI-HMM consisted of a set of context-independent phone HMM models concatenated strictly left-to-right according to the correct phonetic transcription of the password. This SI-HMM will be referred to as (M_c, λ). The phone HMM models are taken from the speaker-independent HMM model λ (see Section 4.3).

- Second, starting from (M_c, λ), a MAP adaptation technique (Gauvain and Lee, 1994) using the speaker enrollment data (5 repetitions) is then performed to create (M_c, λ_c). where λ_c represents the set of customer adapted phone HMM parameters. This procedure consists of adapting only the mean of Gaussians of (M_c, λ) model. In the present work, a simplified version of the adaptation formula is used:

$$\hat{\mu}_{j_{\lambda_c}}^{q_i} = \alpha \mu_{j_\lambda}^{q_i} + (1-\alpha)\frac{\sum_{n=1}^{N} P(j, q_i|x_n)x_n}{\sum_{n=1}^{N} P(j, q_i|x_n)} \tag{4.9}$$

where $\hat{\mu}_{j_{\lambda_c}}^{q_i}$ is the new mean of the j-th Gaussian in the state q_i for client S_c, $\mu_{j_\lambda}^{q_i}$ is the corresponding mean in the model (M_c, λ), $P(j, q_i|x_n)$ is the joint posteriori probability of the state q_i and the Gaussian j and α is the adaptation factor and corresponds to the weight given to the *a priori* information.

4.4.2 Speaker verification

As explained in Section 4.2, the decision to accept or reject a speaker is based on the verification score (4.5) resulted from the combination of two *log likelihood ratios*. The first *LLR* corresponds to the *speaker verification* part, and is estimated as follows:

$$LLR_s = \frac{1}{T}\left[\log p(X|M_c, \lambda_c) - \log p(X|M_c, \lambda)\right] \tag{4.10}$$

The second *LLR* corresponds to the *utterance verification* part. The normalization model in (4.7) and (4.8) should represent all the words but the customer's password. Training a model satisfying this characteristic is a very difficult task (actually impossible). Therefore, a Gaussian mixture model of parameters Λ is used (see Section 4.3). The values of LLR_u in (4.7) and (4.8) are then estimated as follows:

$$LLR_u = \frac{1}{T}\left[\log p(X|M_c, \lambda_c) - \log p(X|\Lambda)\right] \tag{4.11}$$

Or

$$LLR_u = \frac{1}{T}\left[\log p(X|M_c, \lambda) - \log p(X|\Lambda)\right] \tag{4.12}$$

The difference between (4.11) and (4.12) is that the numerator in (4.11) uses the speaker-dependent HMM model (M_c, λ_c) to estimate the utterance verification score, while in (4.12), it uses speaker-independent HMM model (M_c, λ).

During the Viterbi decoding, a silence phone model is applied at the beginning and the end of the customer model (M_c,λ_c) to detect the silence frames and discarded them from the log likelihood ratio estimation for both speaker and utterance verification parts. We have found that the use of this silence/speech segmentation gave better performance compared to that obtained by the speaker-independent HMM model (M_c^{ℓ}, λ).

4.4.3 Performance evaluation

The combined parameter α is determined *a posteriori* to optimize the equal error rate (EER). Note that for the second protocol $P2$, where all test accesses correspond to the expected password, the evaluation is made using only the speaker verification part, i.e., the LLR_s in (4.6).

Table 4.2 reports the performance of the reference TD-SV using protocols $P1$ and $P2$ (see Section 2.7.3 for a description of these two protocols). It can be seen that:

PROTOCOL	UV CRITERION	$LLR_u[\%]$	$LLR_s[\%]$	EER[%]
$P1$	SD-HMM	3.2	3.6	3.0 ($\alpha = 0.3$)
	SI-HMM	27.7	3.6	3.1 ($\alpha = 0.6$)
P2	-	-	-	5.2

Table 4.2. *EER of the reference text-dependent speaker verification system using $P1$ and P2*

- Using the speaker-dependent HMM model for utterance verification performs significantly better (3.2%) than the use of speaker-independent HMM model (27.7%). The reason is that the LLR_u estimated using the SI-HMM model reflects only how likely the test utterance corresponds to the expected password, which increases the false acceptance rate. However, the LLR_u estimated using the SD-HMM (4.11) reflects also how likely the test utterance belongs to the customer.
- The benefit from the use of SD-HMM is not reflected in the EER of the TD-SV, when the LLR_u score is combined with the LLR_s score. The results show a slight difference between the two EERs.

In following experiments, we will use (4.11) for LLR_u estimation.

4.5 Phonetic transcription inference

As we have explained in the introduction, the first step in the development of a UCP-SV system is the HMM inference. This step is achieved by first finding the best phonetic transcription of each enrollment utterance. The inference of the Phonetic Transcription (PT) is performed using a speaker-independent automatic speech recognizer to match each of the enrollment utterances on an ergodic phonetic model, yielding a phonetic transcription of every enrollment utterance. We could then merge all the resulting models into a single HMM, or simply choose the one yielding the highest likelihood as the reference model.

In this section, only the use of the hybrid HMM/MLP to perform this task will be described. The use of HMM/GMM system is straightforward [2]. As illustrated in Figure 4.1, for each customer S_c and for each acoustic sequence $X_c^\ell = \left\{x^\ell_{(1,c)}, x^\ell_{(2,c)}, ..., x^\ell_{(T,c)}\right\}$ associated with each repetition of the customer password, the MLP outputs provided, for each acoustic frame $x^\ell_{(t,c)}$ at its input, an estimate of the posterior probabilities $P(q_k^t|x^\ell_{(t,c)}, \Theta)$ of phones q_k^t, with $k = 1, ..., K$, and where K is the total number of phones. Using these phone posterior probabilities and an ergodic HMM model M (containing the set of fully connected phonetic states, each of them being associated with a particular MLP output) with minimum state duration constraints equal to 3 and phone transition probability[3] set to 0.5, a simple dynamic programming algorithm (Bourlard *et al.*, 1985) is applied to estimate the best phonetic sequence. This resulted in L phonetic transcriptions M_c^ℓ, with $1 \leq \ell \leq L$ (L is the number of enrollment utterances).

An example of the results of such a procedure is given below. It represents the inferred phonetic transcriptions of 5 repetitions of the same word *"annulation"* spoken by the same customer.

[sil][ll][aa][ss][yy][on][sil]
[sil][aa][ll][aa][ss][yy][on][sil]

[2] The phone likelihoods estimated by GMM will be used instead of phone posterior probabilities estimated by the MLP.

[3] Several of the values have been tested. We have observed that this probability has no significant effect on the topology of the model. We thus chose 0.5 as a uniform value for transition probabilities.

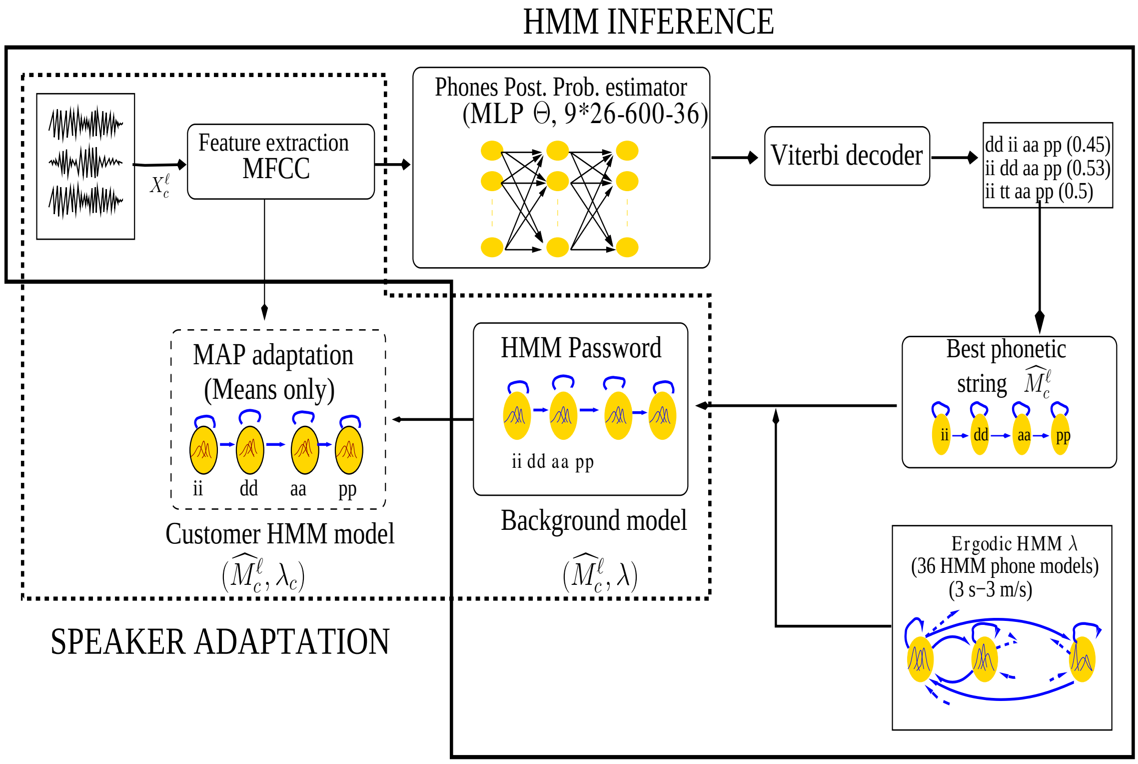

Figure 4.1. *Block-diagram of the enrollment process in the UCP-SV baseline system: For each enrollment utterance X_c^ℓ, we first extract MFCC features which are then fed to the MLP inputs with parameters Θ to estimate phone posterior probabilities, which will be converted to scaled likelihood. A Viterbi decoding through an ergodic HMM/MLP is then performed to search for the most probable phonetic string associated with each utterance. The best phonetic string $\widehat{M}_c^\ell$ is then parameterized by λ, corresponding to speaker-independent GMMs, resulting in a left-to-right speaker-independent HMM/GMM model $(\widehat{M}_c^\ell, \lambda)$, which will be kept and use as background model. Finally, a MAP adaptation procedure is applied to $(\widehat{M}_c^\ell, \lambda)$ to create the speaker-dependent HMM/GMM model $(\widehat{M}_c^\ell, \lambda_c)$.*

[sil][aa][ll][mm][uu][ll][aa][sil][ss][uu][yy][on][an][sil]
[sil][ll][ai][ll][aa][sil][ss][yy][on][sil]
[sil][ll][aa][ff][ss][uu][yy][on][sil]

From this example, it is clear that there is a significant variability between the inferred PTs. This can be attributed to the intra-speaker variability and/or to the inadequacy of the acoustic model (the SI-MLP of parameters Θ for instance).

Once the phonetic transcriptions are inferred, we then aim to create the customer-dependent acoustic model that best represents the lexical content of the password and achieves the best performance. The issue is how do we choose the best phonetic transcriptions to create this model. Two modeling approaches are investigated: single reference model and multiple reference models.

4.6 Single reference HMM based UCP-SV: baseline system

4.6.1 Speaker enrollment

For the baseline system and as illustrated in Figure 4.1, we simply selected the phonetic transcription $\widehat{M_c^\ell}$ yielding the highest normalized (by the number of frames) posterior probability over all the enrollment utterances using forced Viterbi alignment technique,i.e.,

$$\widehat{M_c^\ell} = \underset{1 \le \ell \le L}{\operatorname{argmax}} \left[\sum_{i=1}^{I} \log P(M_c^\ell | X_c^i, \Theta) \right] \tag{4.13}$$

where $L = I$, the number of enrollment utterances (hence phonetic transcriptions). Under the assumption that feature vectors are independent, the $\log P(M_c^\ell|X_c^i, \Theta)$ is defined as the normalized sum of the logarithm of phone posterior probabilities:

$$\log P(M_c^\ell | X_c^i, \Theta) \equiv \frac{1}{T_i} \sum_{t=1}^{T_i} \log P(q_k^{(t,\ell)} | x_{(t,c)}^i, \Theta) \tag{4.14}$$

where $P(q_k^{(t,\ell)}|x_{(t,c)}^\ell, \Theta)$ is the local posterior probability of the decoded phone $q_k^{(t,\ell)}$ using forced Viterbi alignment on the HMM/MLP model (M_c^ℓ, Θ) at time t associated with the frame $x_{(t,c)}^i$ of the i^{th} enrollment utterance, and T_i is the length of the utterance X_c^i after the removal of the silence frames.

As an alternative to *posteriori probabilities* we can also use *scaled likelihoods*. In this case, the best phonetic transcription is generated according to:

$$\widehat{M_c^\ell} = \underset{1 \le \ell \le L}{\operatorname{argmax}} \sum_{i=1}^{I} \left[\frac{1}{T_i} \sum_{t=1}^{T_i} \log \left(\frac{P(q_k^{(t,\ell)} | x_{(t,c)}^i, \Theta)}{P(q_k)} \right) \right] \tag{4.15}$$

where $P(q_k)$ is the *priori* probability of the phone q_k estimated on the *PolyPhone* train data.

The password HMM is then simply built up by concatenating strictly left-to-right (with only loops and skips to the next state) HMM phone models from λ corresponding to each of the phone in the above “optimal” phonetic sequence $\widehat{M_c^\ell}$. The results is an HMM model $(\widehat{M_c^\ell}, \lambda)$ that is acoustically speaker-independent but lexically customer-dependent.

Once the SI-HMM model is created, a MAP adaptation procedure is then performed using (4.9) to estimate the parameters of $(\widehat{M_c^\ell}, \lambda_c)$.

4.6.2 Speaker verification

To verify the claimed identity, we need to define the background and the world models used for score normalization to estimate LLR_s in (4.6) and LLR_u in (4.7). For LLR_u estimation, the GMM model with parameters Λ is used (see Section 4.3).

If some *a priori* knowledge about the content of the password is available, this can help us in designing an effective *background* model to estimate LLR_s. Unfortunately, in UCP-SV system, such information is not available. A straightforward way to define a background model was then to use the inferred SI-HMM password model, which might not be a good model. The two normalized *log likelihood ratios* LLR_s in (4.6) and LLR_u in (4.7) are then estimated as follows:

$$LLR_s = \frac{1}{T}\left[\log p(X|\widehat{M}_c^{\ell}, \lambda_c) - \log p(X|\widehat{M}_c^{\ell}, \lambda)\right] \tag{4.16}$$

$$LLR_u = \frac{1}{T}\left[\log p(X|\widehat{M}_c^{\ell}, \lambda_c) - \log p(X|\Lambda)\right] \tag{4.17}$$

4.6.3 Performance evaluation

The goal of this experiment is to evaluate and analyze the performance of the baseline UCP-SV system by comparing the EER to that obtained by the reference TD-SV system.

Figure 4.2 shows the EER variations of the baseline UCP-SV and the reference TD-SV systems as a function of the combined parameter α using the first protocol[4] $P1$. Figure 4.3 shows the DET curves for different systems using the second protocol $P2$ and Table 4.3 and Table 4.4 report the best EER for different systems using, respectively, $P1$ and $P2$. It is clear that:

- The use of *a priori* information about the lexical content of the password (the phonetic transcription in the case of TD-SV) helps in improving the verification performance of the TD-SV system. In particular when impostor accesses are made with the expected password (i.e., casual impostor).

- The UCP-SV systems using the password HMM inferred by the hybrid HMM/MLP perform significantly better than the UCP-SV system using the password HMM inferred by a HMM/GMM. However, although the phone accuracy on the *PolyVar* customer enrollment data (Table 4.1) using *maximum a posteriori probability* was better, the use of *maximum scaled likelihood* yields better verification performance.

- The performance of the utterance verification part is equivalent in both TD-SV and the best UCP-SV systems.

PROTOCOL	SYSTEMS	LLR_s	LLR_u	EER
$P1$	TD-SV (Dictionary)	**3.6**	**3.2**	**3.0** ($\alpha = 0.3$)
	UCP-SV (Posterior probability)	4.2	3.4	3.2 ($\alpha = 0.2$)
	UCP-SV (Scaled likelihood)	**4.2**	**3.2**	**3.1** ($\alpha = 0.2$)
	UCP-SV (HMM/GMM)	5.6	3.7	3.6 ($\alpha = 0.3$)

Table 4.3. *EER of the reference TD-SV and different baseline UCP-SV systems using the first protocol $P1$ (see Section 2.7.3 for a description of $P1$)*

[4]Both protocols $P1$ and $P2$ are described in Section 2.7.3.

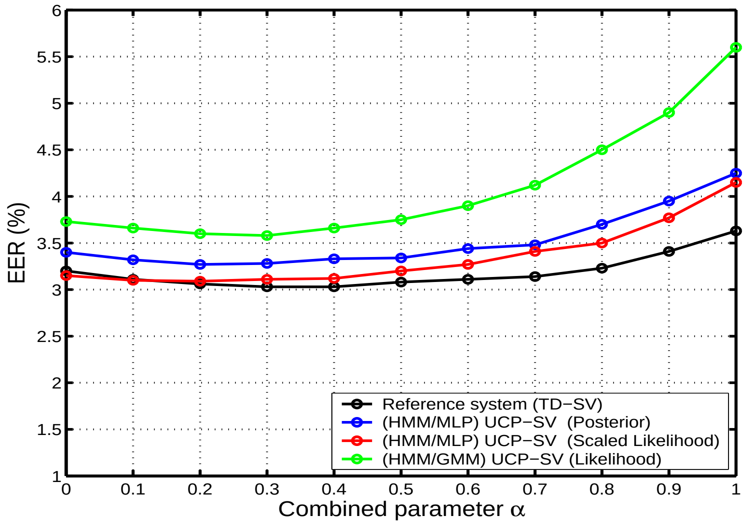

Figure 4.2. *EER variations of the reference TD-SV and different baseline UCP-SV systems as a function of the combined parameter α using the first protocol $P1$*

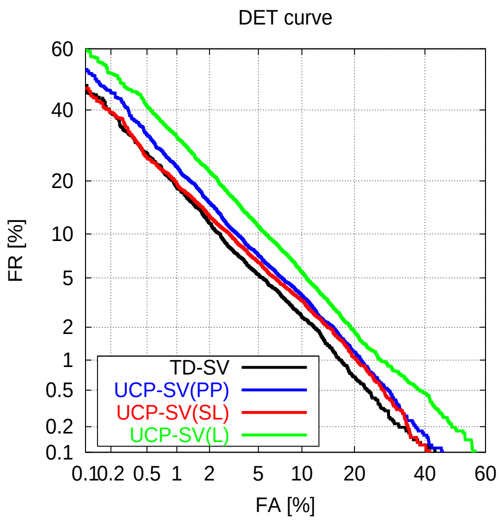

Figure 4.3. *DET curves comparing the performance of the reference TD-SV and different baseline UCP-SV systems using the second protocol $P2$. PP, SL and L mean, respectively, posteriori probability (HMM/MLP), scaled likelihood (HMM/MLP) and likelihood (HMM/GMM)*

PROTOCOL	SYSTEMS	EER
$P2$	TD-SV (Dictionary)	**5.2**
	UCP-SV (Posterior probability)	6.0
	UCP-SV (Scaled likelihood)	**5.7**
	UCP-SV (HMM/GMM)	7.6

Table 4.4. *EER of the reference TD-SV and different baseline UCP-SV systems using the second protocol $P2$ (see Section 2.7.3 for a description of $P2$)*

4.6.4 Analysis

There are two informative values that can help us to analyze and understand these results. These values correspond to the performance of the TD-SV and UCP-SV systems for $\alpha = 0$ and $\alpha = 1$. In the following analysis, we compare the TD-SV system with the UCP-SV system yielding the best speaker verification performance (i.e., UCP-SV (scaled likelihood)).

- $\alpha = 0$:
  The performance of both reference TD-SV and baseline UCP-SV systems using the combined verification score (4.5) becomes equal to the performance using only the *utterance verification* part (LLR_u). In this case, the TD-SV and the UCP-SV systems have the same world model Λ (GMM) for score normalization, but they use an HMM model created from two different phonetic transcriptions. So, if one of these systems (reference or baseline) performs better than the other, this should be attributed to the customer HMM model. The equal error rates associated with $\alpha = 0$ show that the baseline system performs comparably with the reference system (EER = 3.2%). This indicates that the improvement of the reference system cannot be attributed to the fact that this system used the correct phonetic transcription to create the customer-dependent model while the baseline UCP-SV system used the inferred phonetic transcription. It is interesting to note here that in both systems, customer password HMMs are adapted using the same enrollment data but different *a priori* distribution. The fact that both customer password HMMs perform comparably, means that the adaptation process can reduce the effect of errors introduced during the HMM inference process.

- $\alpha = 1$:
  The performance of both reference TD-SV and baseline UCP-SV systems becomes equal to the performance using only the *speaker verification* part (LLR_s). Both systems have two different customer HMM models and two different background models. In this case, if one system performs better than the other, this improvement can be attributed either to the customer HMM model or to the *background* model. As we have seen in the case of $\alpha = 0$, the customer model performs comparably in both reference and baseline systems. Hence, the improvement in the reference system is in great part due to the background model which -in the case of reference system- is more competitive than the one used in the baseline UCP-SV. This explains why the difference between the EERs obtained by the reference and the baseline UCP-SV systems increases as the weight given to the *speaker verification* part increases and why the reference system performs better than the baseline UCP-SV system using the second protocol $P2$.

This is consistent with what has been found in (Rosenberg and Parthasarathy, 1997). One possible explanation is that the *background* model should cover as much as possible the acoustic space of how other speakers pronounce the expected password, and not only how a specific speaker (customer) pronounces it.

To improve the competitiveness of the background model, several techniques have been proposed in (Siohan *et al.*, 1999; Hebert and Peters, 2001) (see Section 3.5).

In this thesis, we propose the use of multiple background models, corresponding to the inferred SI-HMM models, and test different criteria to select the one yielding the best performance.

4.7 Multiple reference models

Using the above criterion (4.15) to select the most likely phonetic transcription during the speaker enrollment step, the associated password HMM model might match well with the training data, but it does not mean that: (1) this model will be *lexically* the most likely during verification, and (2) the associated *background* model will be *lexically* the most competitive to the customer model.

It has been shown in (Jain *et al.*, 1996) that the use of more than one inferred phonetic transcription to represent a word yielded to better recognition performance in voice dialing task. In this thesis, we will demonstrate that the use of more than one model for *background* modeling can improve significantly the verification performance of the UCP-SV system.

4.7.1 Speaker enrollment

In multiple references modeling approach, instead of selecting only one phonetic transcription, we keep all of them and create a customer password HMM model for each one, using the same procedure described above. This results in a set of L customer password HMMs (M_c^ℓ, λ_c) and L *background* models (M_c^ℓ, λ). This approach can be beneficial in selecting: (1) the customer password HMM that best (lexically) matches the test utterance and (2) an appropriate *background* model that is lexically more competitive with the customer password HMM.

4.7.2 Speaker verification

Given a set of customer-dependent HMMs (M_c^ℓ, λ_c) and a set of *background* models (M_c^ℓ, λ), the verification score on which the decision will be made, can be estimated in several ways:

- If we assume that (M_c^ℓ, λ_c) and (M_c^ℓ, λ) are statistically independent, then we can use during the access to the system, some criteria to select separately both customer and background models to optimize the EER. Such techniques will be referred to as *dynamic model selection* (DMS) techniques.
- If we assume that (M_c^ℓ, λ_c) and (M_c^ℓ, λ) are statistically dependent (at least they have the same topology and one is adapted from the other), then we can assume that the background model (M_c^ℓ, λ) is the most competitive to the customer HMM (M_c^ℓ, λ_c). In this case, we will have L UCP-SV subsystems, and we can use certain criteria to chose the best subsystem (with respect to the test utterance) or we can use some *score fusion* or *decision fusion* techniques to estimate the final verification score.

In the rest of this chapter, only experimental results of the UCP-SV system with scaled likelihoods as HMM inference criterion will be reported. Corresponding results with posterior probabilities can be found in (BenZeghiba and Bourlard, 2004a). We should note here that when customers are modeled by multiple reference models, the use of posterior probabilities or scaled likelihoods for HMM inference does not affect the verification performance.

4.7.3 Dynamic model selection techniques

In these techniques, the verification score VS will be estimated as follows:

$$VS = \alpha SVS + (1-\alpha)UVS \tag{4.18}$$

where SVS and UVS are respectively, the speaker verification score and the utterance verification score.

- *Utterance verification score*
  Because the estimation of UVS uses a GMM of parameter Λ for score normalization, the performance of the *utterance verification* part will largely depend on how good the customer model matches the test utterance X. The optimal criterion, with respect to the role of the GMM is probably to select the most likely customer model $(\widehat{M}_c^\ell, \lambda_c)$. That is:

$$(\widehat{M}_c^\ell, \lambda_c) = \underset{1 \le \ell \le L}{\operatorname{argmax}} \; \log p(X|M_c^\ell, \lambda_c) \tag{4.19}$$

  The UVS in (4.18) is then estimated as follows:

$$UVS = \max_{1 \le \ell \le L} LLR_u^{M_c^\ell} = LLR_u^{\widehat{M}_c^\ell} = \frac{1}{T}\left[\log p(X|\widehat{M}_c^\ell, \lambda_c) - \log p(X|\Lambda)\right] \tag{4.20}$$

- *Speaker verification score*
  The performance of the *speaker verification* part does not depend only on how good the customer model matches the test utterance, but also on how well the background model competes with the customer model. Consequently:

  - Both customer and background models may have different model selection criterion to estimate SVS.
  - The selection criterion of the customer and the background models may depend on some statistics applied directly to the LLR_s estimated by each subsystem.

  Three different criteria are tested. They are presented below according to the competitiveness of the background model to the customer model from low to high level.

  1. **Maximizing** $p(X|M_c^\ell, \lambda_c)$:
     Using this criterion, the best customer HMM model $(\widehat{M}_c^\ell, \lambda_c)$ is first selected according to (4.18). Then the associated background model $(\widehat{M}_c^\ell, \lambda)$ is used for score normalization. The SVS in (4.18) is estimated as follows:

$$SVS = LLR_s^{\widehat{M}_c^\ell} = \frac{1}{T}\left[\log p(X|\widehat{M}_c^\ell, \lambda_c) - \log p(X|\widehat{M}_c^\ell, \lambda)\right] \tag{4.21}$$

     However, this criterion might not be a good criterion with respect to the competitiveness constraint of the background model. Indeed, as we will see in the results, a good customer model might have a poor associated background model.

  2. **Maximizing** $p(X|M_c^\ell, \lambda)$:
     While keeping the same customer HMM model selection criterion (4.19) as before, maximizing $p(X|M_c^\ell, \lambda)$ aims to make the background model more competitive by selecting the one that best matches the test utterance.

$$(\widehat{M}_c^{\ell'}, \lambda) = \underset{1 \le \ell \le L}{\operatorname{argmax}} \; p(X|M_c^\ell, \lambda) \tag{4.22}$$

     Thus, the SVS in (4.18) will be estimated as follows:

$$SVS = \frac{1}{T}\left[\log p(X|\widehat{M}_c^\ell, \lambda_c) - \log p(X|\widehat{M}_c^{\ell'}, \lambda)\right] \tag{4.23}$$

     It might happen that both customer and background models will have the same topology (i.e., $\ell = \ell'$, derived from the same phonetic transcription). In this case, this criterion will be equivalent to the previous one.

3. **Minimizing** $LLR_s^{M_c^\ell}$:
Since the *speaker verification* part is based on the estimation of the log likelihood ratio, it may be better if the model selection criterion is applied directly to the LLR_s with respect to the competitiveness constraint. The criterion presented here selects the best phonetic transcription M_c^ℓ that minimize the log likelihood ratio between the customer and its associated background models. Hence, the SVS in (4.18) will be estimated as follows:

$$SVS = \min_{1\leq\ell\leq L} LLR_s^{M_c^\ell} \tag{4.24}$$

The drawback of dynamic model selection criteria though is that there is no guarantee that the selected parameters (customer and background models) will be “optimal” in the sense of yielding the optimal EER.

Performance evaluation

Table 4.5 reports the obtained results, using the first protocol $P1$ and *scaled likelihood* for HMM inference. The second column reports the EER of the *speaker verification* part with different model selection criteria. The third column reports the EER of the *utterance verification* part using (4.20) and the last column reports the EER of the UCP-SV system with the optimal value of the parameter α. Table 4.6 reports EERs using the second protocol $P2$. Performance of both the reference and the UCP-SV baseline systems are also reported (second and third raw, respectively).

DMS criterion	SVS[%]	UVS [%]	EER [%]
REFERENCE SYSTEM	**3.6**	**3.2**	**3.0** ($\alpha = 0.3$)
UCP-SV BASELINE SYSTEM	4.2	3.2	3.1 ($\alpha = 0.2$)
MAX $p(X\|M_c^\ell, \lambda_c)$ (4.21)	5.0	3.3	3.3 ($\alpha = 0.2$)
MAX $p(X\|M_\ell, \lambda)$ (4.23)	4.5	3.3	3.3 ($\alpha = 0.1$)
MIN LLR_s (4.24)	**3.5**	**3.3**	**3.1**($\alpha = 0.5$)

Table 4.5. *EER of the UCP-SV system using $P1$ with different dynamic model selection criteria. The scaled likelihood is used for HMM inference. The second and the third raw report the EER, respectively, of the reference TD-SV and the baseline UCP-SV systems.*

DMS criterion	EER [%]
REFERENCE SYSTEM	**5.2**
UCP-SV BASELINE SYSTEM	5.7
MAX $p(X\|M_c^\ell, \lambda_c)$ (4.21)	6.2
MAX $p(X\|M_\ell, \lambda)$ (4.23)	5.8
MIN LLR_s (4.24)	**5.3**

Table 4.6. *EER of the UCP-SV system using P2 with different dynamic model selection criteria, The second and the third raw report EERs, respectively, of the reference TD-SV and the baseline UCP-SV systems.*

Discussion

Several observations can be made from these results:

1. *Second protocol evaluation:*

   - The performance using the *background* model associated with the best customer model (criterion (4.21)) is worse (6.2%) than that obtained with the baseline system (5.7%). A possible reason is that the $(\widehat{M}_c^\ell, \lambda_c)$ is selected dynamically according to the maximum likelihood criterion. For many impostor accesses, the forced Viterbi alignment against $(\widehat{M}_c^\ell, \lambda_c)$ results in a good likelihood score, and because $(\widehat{M}_c^\ell, \lambda)$ is not necessarily an appropriate *background* model, many impostor accesses will get accepted. It is worth mentioning here that a good background model is useful in reducing the false acceptance rate.
   - The selection of $(\widehat{M}_c^\ell, \lambda_c)$ and $(\widehat{M}_c^{\ell'}, \lambda)$ separately, according to the maximum likelihood criterion (4.23), improved the performance compared to the use of maximum $p(X|M_c^\ell, \lambda_c)$ criterion (4.21). This improvement is 98% significant according to the significant test proposed in (Bengio and Mariéthoz, 2004). But this improvement is not significant compared to the baseline system.
   - Significant improvements (98% of confidence according to (Bengio and Mariéthoz, 2004)) are obtained using the $Minimum\ LLR_s$ as a selection criterion (4.24). The EER dropped from 5.7% to 5.3%. As we can see, the performance of the UCP-SV system is quite competitive with the reference system.

     We should mentioned here, that in previous work (BenZeghiba and Bourlard, 2004b), we have found that this criterion is not *optimal*. If for a given test utterance, the different $LLR_s^{M_c^\ell}$ estimated by each subsystem are sorted from low to high value according to the maximum likelihood ratio criterion, then the best EER was achieved by the use of the second $LLR_s^{M_c^\ell}$ in the list. We have found also that this criterion is better for low false acceptance rate (FAR) applications.

2. *First protocol evaluation:*

   - The use of (4.20) to select the customer HMM model did not improve the performance of the *utterance verification* part. Taking into account, our acoustic modeling approach, it seems that the value of 3.2% is the best we can achieve.
   - Surprisingly, and despite the significant improvement (100% of confidence) in the *speaker verification* part (Table 4.5, column 2), no improvement in the performance of the UCP-SV system is obtained. A possible reason is that the world model (GMM) used for score normalization in the utterance verification part is trained with general speech data from a large set of speakers. It covers the general acoustic space including the customer password. Hence, it has some acoustic characteristics of the *background* model, making the amount of new (complementary) information given by the *speaker verification* part very low.

     We have plotted (Figure 4.4) the UVS score (4.20) against the SVS score (4.24) to see how high is the correlation between these two scores. The figure shows that the UVS and SVS are highly correlated, particularly for the customer accesses. Using all client and impostor accesses, the *correlation coefficient* ρ between UVS and SVS for customer and impostor accesses are found to be 0.90 and 0.80, respectively.

4.7.4 Verification score fusion

In verification score fusion, the inputs to the fusion system are the *individual* verification scores estimated by each subsystem and the outputs are the average of $LLR_s^{M_c^\ell}$ and $LLR_u^{M_c^\ell}$ over all sub-

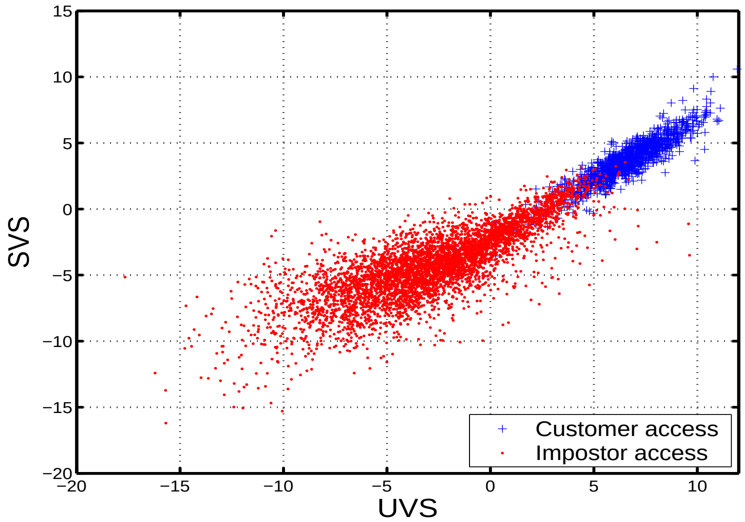

Figure 4.4. *Distribution of utterance verification scores (UVS) according to (4.20) against speaker verification score (SVS) according to (4.24).*

systems. the speaker verification score (SVS) and the utterance verification score (UVS) are then estimated as follows:

$$SVS = \frac{1}{L}\sum_{\ell=1}^{L} LLR_s^{M_c^\ell} \tag{4.25}$$

$$UVS = \frac{1}{L}\left[\sum_{\ell=1}^{L} LLR_u^{M_c^\ell}\right] \tag{4.26}$$

where L is the number of subsystem. The final verification score VS is then a weighted sum of SVS and UVS estimated as follows:

$$VS = \alpha SVS + (1-\alpha)UVS \tag{4.27}$$

It is worth mentioning here that the use of the average of the individual VS prevents us from the use of a poor set of parameters (subsystem) to estimate SVS and UVS.

Performance evaluation

Figure 4.5 shows the EER variations of the UCP-SV using (4.27), the reference TD-SV and the baseline systems, as a function of the combined parameter α. These results are obtained using the first protocol $P1$. Tables 4.7 and 4.8 report the EER of each system obtained using, respectively, $P1$ and $P2$.

Figure 4.5 shows that using the average score criterion (4.27), the UCP-SV system performs comparably with the reference system for all values of the combined parameter α. Tables 4.7 shows a small improvement[5] compared to the baseline UCP-SV system. Tables 4.7 and 4.8 show that the use of average score criterion gives comparable results with those obtained using the best dynamic

[5]This improvement is significant with 79% of confidence.

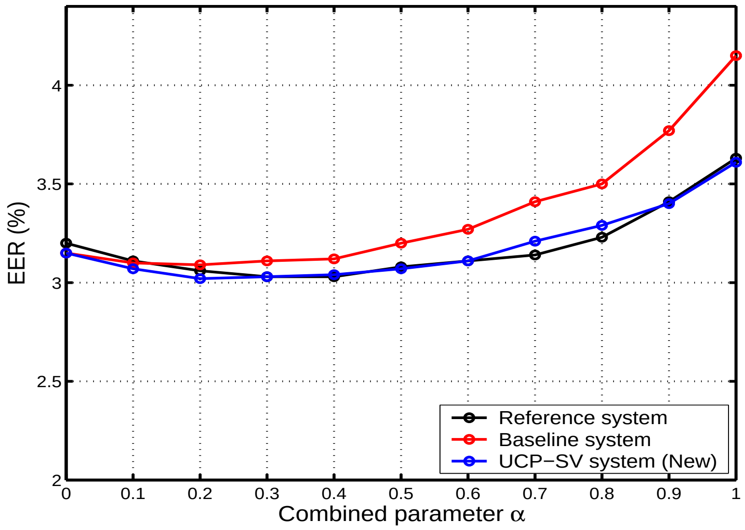

Figure 4.5. *EER variations of the UCP-SV system as a function of the combined parameter α using the verification score fusion technique (4.27) and $P1$*

SYSTEM	SVS[%]	UVS[%]	EER[%]
REFERENCE SYSTEM	**3.6**	**3.2**	**3.0** ($\alpha = 0.3$)
UCP-SV BASELINE SYSTEM	4.2	3.2	3.1 ($\alpha = 0.2$)
UCP-SV (NEW SYSTEM)	**3.6**	**3.2**	**3.0** ($\alpha = 0.2$)

Table 4.7. *EER of the UCP-SV system using verification score fusion technique (4.27), compared to the reference TD-SV and the baseline UCP-SV systems. Evaluation is done using $P1$.*

model selection criteria, i.e., the *min* function (4.24) for SVS estimation and the maximum likelihood criterion (4.20) for UVS estimation. This indicates that the use of (4.24) and (4.20) are a good criteria.

We should note here, that, for a given customer, the verification score estimated by each subsystem are not statistically independent. Indeed, all subsystems are trained using the same adaptation data and the same adaptation procedure, only phonetic transcriptions are different. Consequently, given a test utterance X, there is a set of *optimal* parameters corresponding to only one customer HMM model (M_c^ℓ, λ_c) that gives the best UVS and a set of *optimal* parameters corresponding to only one phonetic transcription M_c^ℓ that gives the best SVS. The combination of these two scores will give the best performance. The use of the other models will be useless as they do not carry any complementary information. Because the search for these *optimal* models is not obvious, using the average score will prevent us from the choose of the poor parameters.

4.7.5 Partial decision fusion

In partial decision fusion, the inputs to the fusion system are the *individual* decisions made by each subsystem and the output is the final verification score. The fusion system uses a *majority voting* technique similar to what suggested in (Li *et al.*, 2000). The verification score is then defined as

SYSTEM	EER[%]
REFERENCE SYSTEM	**5.2**
UCP-SV BASELINE SYSTEM	5.7
UCP-SV (NEW SYSTEM)	**5.4**

Table 4.8. *EER of the UCP-SV system using verification score fusion technique (4.27), compared to the reference TD-SV and the baseline UCP-SV systems. Evaluation is done using P2.*

follows:

$$VS = \frac{1}{L}\sum_{\ell=1}^{L} f(vs_\ell) \tag{4.28}$$

where

$$f(vs_\ell) \quad = \quad \begin{cases} 1, & \text{if } \; vs_\ell \;\geq \delta_{(c,\ell)} \\ 0, & \text{otherwise} \end{cases} \tag{4.29}$$

(4.30)

where vs_ℓ is the combined verification score (4.18) estimated by the UCP-SV subsystem using the phonetic transcription M_c^ℓ, and $\delta_{(c,\ell)}$ is a local customer and model dependent threshold. This VS, which belongs to the $[0,1]$ interval, can be interpreted as a percentage of times that the local verification score vs_ℓ exceeded its local threshold $\delta_{(c,\ell)}$.

One difficulty that can make the use of this technique impractical in real application is the estimation of the local threshold $\delta_{(c,\ell)}$ for each subsystem. Indeed, it is desirable to have a local threshold that:

1. Is customer and model independent ($\delta_{(c,\ell)} = \delta$), hence, it can be determined *a priori* on separate data.

2. Is interpretable and adjustable, hence, it can easily be adjusted according to the application requirements.

3. Allows the parameter α to be optimized independently of the subsystem. That is, all subsystems use the same value of the parameter α for *speaker verification* and *utterance verification* scores combination.

The $LLR_s^{M_c^\ell}$ and $LLR_u^{M_c^\ell}$ have a large dynamic range, theoretically belonging to $]-\infty,+\infty[$ interval. To satisfy the above conditions, we have introduced the *normalized log likelihood ratio* (NLLR) that transforms $LLR_s^{M_c^\ell}$ and $LLR_u^{M_c^\ell}$ into more interpretable scores. The *normalized log likelihood ratio* uses the *log likelihood ratio* of the train data to normalize the *log likelihood ratio* of the test data, and is based on the following assumption:

$$\frac{LLR(test)}{LLR(train)} \leq 1 \tag{4.31}$$

which states that the *log likelihood ratio* estimated using the train data is the best *log likelihood ratio* we can get. We have used this assumption to normalize $LLR_s^{M_c^\ell}$ and $LLR_u^{M_c^\ell}$. Given a customer model (M_c^ℓ, λ_c), the $NLLR_s^{M_c^\ell}$ can be defined as:

$$NLLR_s^{M_c^\ell} = \frac{LLR_s^{M_c^\ell}}{\frac{1}{I}\sum_{i=1}^{I}\left[\log p(X_c^i|M_c^\ell,\lambda_c) - \log p(X_c^i|M_c^\ell,\lambda)\right]} \tag{4.32}$$

and $NLLR_u^{M_c^\ell}$ as:

$$NLLR_u^{M_c^\ell} = \frac{LLR_u^{M_c^\ell}}{\frac{1}{I}\sum_{i=1}^{I}\left[\log p(X_c^i|M_c^\ell,\lambda_c) - \log p(X_c^i|\Lambda)\right]} \tag{4.33}$$

where I is the number of enrollment utterances for the customer S_c. The denominators in (4.32) and (4.33) are the average *log likelihood ratio* over the customer enrollment data.

Using (4.32) and (4.33), the new verification score vs_ℓ in (4.29) will be estimated as follows:

$$vs_\ell = \alpha\ NLLR_s^{M_c^\ell} + (1-\alpha)\ NLLR_u^{M_c^\ell} \tag{4.34}$$

Using these transformations together with assumption (4.31), the $NLLR_u^{M_c^\ell}$ and $NLLR_s^{M_c^\ell}$ will, theoretically, have a limited dynamic range with an upper bound equal to 1. Consequently the vs_ℓ in (4.34) will be bounded by 1. The values $NLLR_s^{M_c^\ell}$ and $NLLR_u^{M_c^\ell}$ indicate how likely the test utterance belongs to the claimed identity. Closer are $NLLR_u^{M_c^\ell}$ and $NLLR_s^{M_c^\ell}$ to 1, more likely the claimed identity is to be valid.

Note that in this approach we now have two thresholds, a local threshold δ to which the local verification score will be compared to make a local decision, and a global threshold Δ to which the final verification score will be compared to make the final decision to accept or reject the speaker.

Performance evaluation

Table 4.9 shows the EER of the UCP-SV system for both protocols.

PROTOCOL	*Local threshold* δ	*Global threshold* Δ	EER[%]
P1	0.28	0.6	3.1 ($\alpha = 0.2$)
P2	0.25	0.6	5.4

Table 4.9. *EER of the UCP-SV system with its optimal local and global thresholds using partial decision fusion technique (4.29) with the first and second protocols.*

This technique performs comparably with the two previous techniques. An additional observation is that, the global threshold Δ is equal to 0.6. This means that the speaker is accepted if 3 among the local verification scores vs_ℓ exceeded the local threshold δ.

To check whether the assumption (4.31) is valid or note, we have computed the number of times that $NLLR_s^{M_c^\ell}$ and $NLLR_u^{M_c^\ell}$ exceeded the value 1.

- For the $NLLR_u^{M_c^\ell}$, we found that over $299,670$ accesses, there were 95 accesses that their $NLLR_u^{M_c^\ell}$ exceeded 1, giving a relative frequency of 0.00032. Moreover, all these accesses were customer's accesses. So, for the $NLLR_u^{M_c^\ell}$ there is no violation of the assumption.

- For the $NLLR_s^{M_c^\ell}$, we found that over $299,670$ accesses, there were 163 accesses that their $NLLR_s^{M_c^\ell}$ exceeded 1, giving a relative frequency of 0.00055. Among these 163 accesses, 153 of them were customer's accesses with the expected password, 1 was impostor's access with the expected password and 9 were impostor's accesses with wrong passwords. This is not surprising as the *speaker verification* part supposes to discriminate between customers and impostors pronouncing the expected password. So, here also, there is no violation of the assumption.

- When we take the optimal combined confidence score cs_ℓ, there were no values higher than 1.

For comparison purposes, we have also used the original $LLR_u^{M_c^\ell}$ and $LLR_s^{M_c^\ell}$ to estimate the combined score. We have got the same performance with $P1$ and $P2$. The advantage of the $NLLR$, however, is that the $NLLR$ can be used as a criterion to select test utterances for incremental customer model adaptation.

4.8 Real time prototype

Based on the work described above, a real time version of the UCP-SV system has been developed and integrated [6] in an existing PC-based biometric authentication application developed at IDIAP. The authentication is carried out using fusion of two modalities: speech and face. The speech modality is a GMM-based text-independent speaker verification. The UCP-SV system uses multiple reference modeling approach with verification score fusion technique. The use of the application consists of two phases (see (Kowalczyk, 2004) for technical details and difficulties encountered during the development of this prototype).

1. *Enrollment phase:* The enrollment phase is performed in one recording session and it is consisted of the following step:

   - Each new customer is prompted (by text) to pronounce 5 times his/her password using a microphone. Before and after each pronunciation, the customer has to click on the begin and the end buttons, so only the signal recorded between this two times will be stored.
   - Once the recording session is finished, the signal processing module analyzes each waveform file and computes MFCC acoustic vectors (frames). Each frame consists of 26 coefficients extracted every 10 ms (see Section 4.3). To alleviate the effect of mismatch between train and test conditions, cepstral mean subtraction technique is applied to the acoustic vectors.
   - These acoustic vectors are then fed to the input layer of an MLP (the same used in this work, see Section 4.3) to perform the HMM inference step.
   - For each inferred phonetic transcription, a customer HMM password model is created as described in Section 4.7.

   Creating 5 customer HMMs found to be time consuming. Therefore, we have reduced the number the iterations during the adaptation process.

2. *Verification phase:* During the access to the system, a speaker pronounces a password and selects from a list of registered customers the claimed identity. The system downloads the claimant password HMMs, the background models and the GMM to estimate the final verification score using the verification score fusion technique described in Section 4.7.4. The final verification score is then compared to a speaker-independent threshold to make the decision to accept or reject the speaker. The verification time depends on the length of the test utterance and the number of parameters in different models used for verification score estimation. With passwords and test accesses corresponding to names, the average verification time was found to be 7 seconds.

[6] This real time version of the HMM/GMM based UCP-SV system was developed in the framework of undergraduate internship student project by Jérôme Kowalczyk.

The values of the hyperparameters such as the MAP adaptation factor, the parameter α used to combine the utterance verification and speaker verification scores as well as the value of the threshold are adjusted on the *PolyVar* database using the first protocol $P1$.

The development and the use of this application allowed us to evaluate the robustness of the UCP-SV system in real conditions and to measure how difficult is to reproduce comparable performance to those obtained in the lab. Although the use of this application gives satisfactory results, several points of weaknesses are observed. Some of them are described below:

- Even if a cepstral mean subtraction technique is applied, the mismatch between train and test conditions still remains the major difficulty. This mismatch can take several forms, such as the level of noise in the room and the distance between the speaker and the microphone.
- The telephone quality of the development data (i.e., *PolyVar* database) used to adjust some functional parameters is different from the microphone quality of the training data provided by each customer. This mismatch makes the adjusted parameters less reliable which affect the performance of the system.

4.9 Conclusion

This chapter has developed and compared HMM/GMM based UCP-SV systems using both single reference model and multiple reference models approaches. A speaker-independent automatic speech recognizer (SI-ASR) is first used to infer the phonetic transcriptions associated with the enrollment utterances, which are then used to create the customer-dependent HMM models.

To evaluate the effect of the SI-ASR performance in terms of phone accuracy on the verification performance of the UCP-SV system, a hybrid HMM/MLP and a HMM/GMM ASR systems are used for HMM inference. Results using single reference model approach showed that the use of HMM/MLP gave significantly better results. This chapter has shown also that with a speech recognizer that has 56.6% phone accuracy, we can develop a UCP-SV system with acceptable performance, but not competitive with a TD-SV system. The analysis of the results has revealed that the main reason of this limitation lies in the background model, which is less competitive to the customer model in the UCP-SV system than in the TD-SV system.

To improve the performance of the UCP-SV system, the use of multiple reference and background models approach has been introduced and investigated. In this context, different scoring criteria have been proposed and tested, including dynamic model selection techniques, verification score fusion, and partial decision fusion techniques.

Results have shown that the use of minimum likelihood ratio estimated by each subsystem as a verification score achieved good verification performance. Similar improvement could be obtained by taking the average log likelihood ratios estimated by each subsystem or using partial decisions fusion.

Finally, this chapter has demonstrated that the availability of the *a priori* knowledge about the password is a sufficient but not a necessary condition to develop a high quality TD-SV system.

Chapter 5

HMM/MLP based UCP-SV

5.1 Introduction

This chapter aims to investigate the use of MLPs instead of GMMs (previous chapter) to estimate the posterior probabilities of the inferred HMM states. The motivation behind this investigation is to exploit the benefits of the use of hybrid HMM/MLP systems exhibited in speech recognition systems for speaker verification. An adapted speaker (customer) dependent MLP is used to estimate HMM state posterior probabilities of the inferred HMM model. Two issues are specifically investigated here, MLP adaptation and verification score estimation.

A speaker-dependent MLP is created for each customer by adapting all the parameters of the MLP (Θ) used for HMM inference using only the adaptation data provided by the customer. The resulting SD-MLP should capture both the lexical content of the password and the speaker characteristics. Thus, we expect that the MLP will give a high verification score to customer access with the expected password and low verification score in all other cases. Given the limited amount and the nature of the adaptation data which consists of only a few examples of a small number of phonemes, adapting a large neural network in this way is unlikely to be effective. Therefore, different MLP adaptation techniques, including the use of a linear input network and adaptation of a small MLP, are investigated. In order to reduce the number of adapted parameters with respect to the amount of adaptation data, various architectures of the linear input network are compared. Significant improvement is reported when some speaker-independent acoustic frames corresponding to the phonemes with zero priori probability, (i.e., phonemes that are not in the inferred phonetic transcription) are added to the adaptation data provided by the customer.

The decision to accept or reject a speaker depends on the reliability of the estimated verification score. In the hybrid HMM/MLP, the MLP is used to estimate the posterior probability that indicates how well the acoustic data matches the customer model. In addition to the use of raw posterior probability scores, score normalization techniques, initially developed for HMM/GMM-based speaker verification are investigated and their performance are analyzed.

The performance of the final HMM/ANN based UCP-SV system was very poor compared to the HMM/GMM based UCP-SV baseline system. The analysis of the results showed that the main problem with these models is their ability to discriminate between customer and impostor accesses is weak. Therefore, new decision rule is introduced where the hybrid HMM/MLP is used for utterance verification and a Gaussian Mixture Model-Universal Background Model (GMM-UBM) is used for text-independent speaker verification.

5.2 Related works

Despite the success of the hybrid HMM/ANN systems for speech recognition tasks, unfortunately, the number of studies which have been devoted to their use for a speaker recognition task are still very limited.

- **A hybrid HMM/ANN speaker verification Algorithm for telephone speech** (Naik and Lubensky, 1994): To our knowledge, this is the first work on HMM/MLP based speaker verification system. In this work, each speaker was represented by an HMM/MLP model. For a fixed-text digit speaker verification task, the authors showed comparable results to HMM/GMM based system. They also suggested that additional gain could be obtained by incorporating more contextual acoustic information in the input layer of the MLP. We must report here that the authors used a sub-optimal decision based on the *likelihood* score instead of *likelihood ratio* (see Section 3.2.4).

- **Combined speech and speaker Recognition with speaker-adapted connectionist Models** (Genoud *et al.*, 1999a): More recently, (Genoud *et al.*, 1999a) proposed a new technique that combines speech and speaker recognition within the target speaker MLP model parameterized by θ_c. They used an MLP with a specific structure called Twin-Outputs MLP (TO-MLP). The target speaker MLP has two sets of K outputs (K is the number of phones). One set for the target speaker and one set for the "anti-speaker". The adaptation process is performed on a mixed (target-speaker and anti-speaker) data. For each output pair, one output is trained to correspond to a particular phone of the target speaker and the second output is trained to correspond to the same phone but for "anti-speaker" data. During verification they first estimate $P(M_c, S_c|X, \theta_c)$, representing the joint *posteriori probability* of the correct password M_c and the correct speaker (target speaker) S_c given X, as obtained at the MLP target speaker specific outputs. Then, they estimate $P(M_c, \overline{S}_c|X, \theta_c)$ representing the joint *posteriori probability* of the correct password M_c and impostor $\overline{S}_c$ given X, as obtained at the MLP anti-speaker specific outputs. The decision to accept or reject a speaker is then made as follows:

$$S = S_c \text{ if } \log P(M_c, S_c|X, \theta_c) - \log P(M_c, \overline{S}_c|X, \theta_c) \geq \delta \tag{5.1}$$

  Evaluation of TO-MLP on the 1997 Broadcast news database showed reasonable results.

5.3 HMM/MLP baseline system for UCP-SV

This section describes the general approach we have followed to implement and evaluate the HMM/MLP based UCP-SV baseline system. It uses a simple MLP adaptation technique. Our goal is to evaluate how well these systems perform for a UCP-SV task.

5.3.1 Speaker enrollment

The enrollment procedure consists of creating a customer-dependent MLP for each new customer registered into the system. The main steps are illustrated in Figure 5.1. Only steps after the HMM inference (described in Section 4.5) are described in the following [1]:

[1]It is worth mentioning here that we are using the best phonetic transcription obtained according to (4.15), i.e., the one that gave the best performance in the previous chapter.

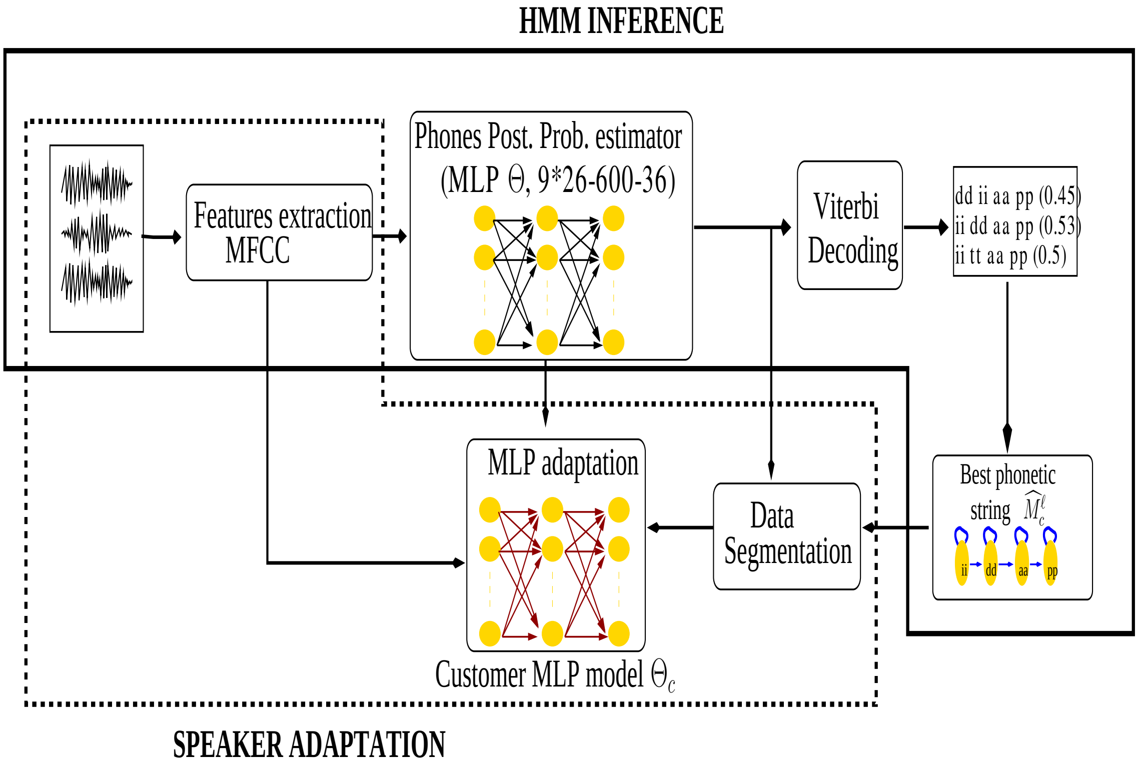

Figure 5.1. *Block-diagram of the enrollment process in the hybrid HMM/MLP based UCP-SV system: After the inference of the best phonetic string $\widehat{M}_c^\ell$ (see Figure 4.1), a forced Viterbi is applied using scaled likelihood derived from the MLP outputs to generate the segmentation (desired outputs). This segmentation is then used to adapt the MLP Θ resulting in a customer dependent MLP parameterized with Θ_c.*

- After HMM inference, we match each of the enrollment utterances X_c^ℓ on the best customer specific model $\widehat{M}_c^\ell$ using posterior probabilities or scaled likelihood estimated by the SI-MLP with parameters Θ (as used for HMM inference) to yield the phonetic segmentation of all the enrollment utterances. That is, each frame in the enrollment utterances will be assigned to one of the phones in the inferred HMM $\widehat{M}_c^\ell$. The supervised adaptation of the SI-MLP parameters to the targeted customer requires this segmentation.
- Adapt the parameters of the SI-MLP Θ using the above segmentation to provide the target outputs and by minimizing the cross-entropy error (3.26) between the observed output vector and the target output vector over all the adaptation data. This results in a speaker-dependent MLP with parameters Θ_c.

Speaker adaptation

For the baseline system, this step consists of retraining all the parameters Θ of the speaker independent MLP to the characteristics of each customer S_c, using -only- the enrollment utterances (five repetitions). The same MLP training characteristics described in Section 3.4.2 are used for MLP adaptation. We started with a learning rate, η equal to 0.01, and the standard error backpropagation algorithm is used to minimize the average cross-entropy:

$$E(X|\Theta,\Theta_c) = \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{K} d_k(x_t,\Theta)\log\frac{d_k(x_t,\Theta)}{g_k(x_t,\Theta_c)} \tag{5.2}$$

where

- $X = \{x_1, x_2, ..., x_T\}$ is the acoustic vector sequence, associated with the adaptation utterances, x_t representing the acoustic vector at time t, T is the total number of training vectors and K is the number of outputs.
- $d(x_t,\Theta)$ represents the target output vector associated with each input vector x_t and corresponding to the phonetic segmentation obtained from the SI-MLP Θ.
- $g(x_t,\Theta_c)$ represents the observed MLP output vector given the current values of the parameters Θ_c:

$$g(x_t,\Theta_c) = \{g_1(x_t,\Theta_c),\ldots,g_k(x_t,\Theta_c),\ldots,g_K(x_t,\Theta_c)\} \tag{5.3}$$

To assess the generalization properties of the speaker-dependent MLP during the adaptation process, the adaptation data (five repetitions) was divided into two parts, the first three repetitions are used to adapt the SI-MLP parameters and the last two repetitions are used to test the generalization properties (as described in Section 3.4.2). At the end of the adaptation, the speaker-dependent MLP should estimate speaker-dependent phone posterior probabilities, i.e., for speaker S_c:

$$g(x_t,\Theta_c) = \left\{P(q_1^t|x_t,\Theta_c),\ldots,P(q_k^t|x_t,\Theta_c),\ldots,P(q_K^t|x_t,\Theta_c)\right\} \tag{5.4}$$

5.3.2 Speaker verification

The SD-MLP Θ_c estimates for each input frame x_t the local phone posterior probabilities $P(q_k^t|x_t,\Theta_c)$ for each phoneme q_k, which will be then used to derive the word posterior probability. It is worth mentioning here that for verification, the use of *scaled likelihood* was not considered. As the amount of the adaptation data is very small and does not contain examples from all phonemes, the estimation of phone *a priori* probabilities from phone relative frequencies obtained from the adaptation data will not be reliable. Hence, all phone *priors* are assumed to be equal.

The verification of a speaker S, pronouncing X and claiming to be S_c with a password $\widehat{M}_c^\ell$ consists of the following steps:

1. Perform a forced Viterbi alignment of X on the speaker model $\widehat{M}_c^\ell$ using local phones *posterior probabilities* estimated by the speaker-dependent MLP Θ_c to estimate $P(\widehat{M}_c^\ell|X,\Theta_c)$. This probability represents the global *posterior probability* that X was actually pronounced by the claimed speaker S_c (since using Θ_c) and corresponds to the expected password $\widehat{M}_c^\ell$.

2. Perform Viterbi alignment of X on model M using local *posterior probabilities* estimated by the speaker-independent MLP Θ to estimate $P(M|X,\Theta)$. This probability represents the global *posterior probability* that X was produced by another speaker pronouncing any word M (hence the looped world model).

3. Another possibility, would be to estimate $P(\widehat{M}_c^\ell|X,\Theta)$, representing the *posterior probability* that the expected password has been produced by another speaker different than the customer S_c (since using Θ).

These probabilities are then used to derive the verification score, which will be compared to a speaker-independent threshold determined *a posteriori* to minimize the EER. Three scoring criteria are tested.

1. *Log posterior probability (LPP):* The decision rule to accept or reject a speaker is expressed as follows:

$$S = S_c \quad \text{if} \quad \log P(\widehat{M}_c^\ell|X,\Theta_c) \geq \delta_1 \tag{5.5}$$

2. *Unconstrained log posterior probability ratio (ULPPR):* Using this criterion, the decision rule can be expressed as follows:

$$S = S_c \quad \text{if} \quad \log P(\widehat{M}_c^\ell|X,\Theta_c) - \log P(M|X,\Theta) \geq \delta_2 \tag{5.6}$$

The LPP score in (5.5) is normalized by $\log P(M|X,\Theta)$, the posterior probability of the most probable phone sequence among all possible phone sequences. This means that there is no constraint during the Viterbi decoding to estimate the normalization score, hence we call it *unconstrained* $LPPR$

3. *Constrained log posterior probability ratio (CLPPR):* The decision rule can be expressed as follows:

$$S = S_c \quad \text{if} \quad \log P(\widehat{M}_c^\ell|X,\Theta_c) - \log P(\widehat{M}_c^\ell|X,\Theta) \geq \delta_3 \tag{5.7}$$

In this case, to estimate the normalization score, the Viterbi decoding is thus constrained by the topology of the customer model $\widehat{M}_c^\ell$.

By analogy with the decision rules used in HMM/GMM based UCP-SV, it seems that the $CLPPR$ score is probably the best in the case of customers or impostors pronouncing the expected password, while the $ULPPR$ score is the best in the case of customers or impostors pronouncing an invalid password.

5.3.3 Using confidence measures

The estimation of the different verification scores in (5.5) (5.6) and (5.7) can take advantage of the hybrid HMM/MLP system to investigate the use of posteriori probability based confidence measure. A Confidence Measure (CM) can be defined as a *function which quantifies how well a model matches some acoustic data, where the values of the function must be comparable across utterances* (Williams and Renals, 1998). It has been shown that the word CM derived from phone posterior probabilities estimated through an MLP outputs are useful for utterance verification task (Williams and Renals, 1997; Bernardis and Bourlard, 1998; Mengusoglu and Ris, 2001), which is in our interest. Moreover, if phone posteriors are estimated by a speaker-dependent MLP, these CMs might encode some speaker specific information. In this chapter, the effectiveness of some of these CMs for the UCP-SV system is evaluated. Two confidence measures are tested: standard posterior probability and double normalization.

- *Standard posterior probability:*
  The Standard Posterior Probability (SPP) based CM is defined as the average frame posterior probability. The different global posterior probabilities in (5.5) (5.6) and (5.7) are then estimated (by definition and under the assumption that acoustic vectors are independent), respectively, as follows:

$$\log P(\widehat{M}_c^\ell|X,\Theta_c) \equiv \frac{1}{T}\sum_{t=1}^{T}\log P(q_k^{(t,\ell)}|x_t,\Theta_c) \tag{5.8}$$

$$\log P(M|X,\Theta) \equiv \frac{1}{T}\sum_{t=1}^{T}\log P(q_k^t|x_t,\Theta) \tag{5.9}$$

  and

$$\log P(\widehat{M}_c^\ell|X,\Theta) \equiv \frac{1}{T}\sum_{t=1}^{T}\log P(q_k^{(t,\ell)}|x_t,\Theta) \tag{5.10}$$

  where $q_k^{(t,\ell)}$ and q_k^t are the decoded phone q_k using Viterbi alignment, respectively, on the customer model $\widehat{M}_c^\ell$ and the ergodic model M at time t associated with the frame x_t. T is the length of the test utterance after the removal of the silence frames. We have found that the normalization by T yields to better results (BenZeghiba *et al.*, 2001). The speech/silence segmentation is obtained by the customer-dependent HMM/MLP system $(\widehat{M}_c^\ell,\Theta_c)$.

  In this CM, all frames contribute equally to the matching score. Consequently, different phones will have different contributions depending on their respective length. The poorly matched phones (i.e., phones with small posterior probability) will have a short Viterbi duration, hence, less contribution.

- *Double normalization:*
  It has been shown in (Bernardis and Bourlard, 1998) and confirmed in (Mengusoglu and Ris, 2001) that the confidence of a model is better approximated by using a *Double Normalization* (DN) of the Viterbi score. This involves a normalization over each phonetic segment (average score over each phonetic segment) followed by a normalization over the number of phones. In our case, this yields the following estimation (by definition) of (5.5) (5.6) and (5.7), respectively:

$$\log P(\widehat{M}_c^\ell|X,\Theta_c) \equiv \left[\frac{1}{P_c}\sum_{p=1}^{P_c}\frac{1}{e_p-b_p+1}\sum_{t=b_p}^{e_p}\log P(q_p^{(t,\ell)}|x_t,\Theta_c)\right] \tag{5.11}$$

$$\log P(M|X,\Theta) \equiv \left[\frac{1}{P}\sum_{p=1}^{P}\frac{1}{e_p - b_p + 1}\sum_{t=b_p}^{e_p}\log P(q_p^t|x_t,\Theta)\right] \tag{5.12}$$

and

$$\log P(\widehat{M}_c^\ell|X,\Theta) \equiv \left[\frac{1}{P_c}\sum_{p=1}^{P_c}\frac{1}{e_p - b_p + 1}\sum_{t=b_p^c}^{e_p^c}\log P(q_p^{(t,\ell)}|x_t,\Theta)\right] \tag{5.13}$$

where b_p and e_p represent, respectively, the beginning and the end of the phone q_p resulting from the Viterbi alignment procedure and P_c and P are, respectively, the number of phones in the customer model $\widehat{M}_c^\ell$ and the number of recognized phones using M.

This CM gives the same importance to all phones in the word independently of the length of each phone. Consequently, the poorly matched phones will have more weight, making this CM useful for rejecting invalid utterances (Rivlin *et al.*, 1996).

5.3.4 Performance evaluation

Experiments are carried out to address several questions:

- How competitive are the hybrid HMM/MLP systems as used here to the HMM/GMM systems for UCP-SV task?
- How useful are the score normalization techniques originally used in HMM/GMM systems in hybrid HMM/MLP systems?
- How useful are the confidence measures initially developed for utterance verification task for a speaker verification task?

Experiments are carried out using both protocols $P1$ and $P2$.

Results

The performance of the baseline system on the protocol $P1$ using different scoring criteria are shown in Figures 5.2 and 5.3. They represent the DET curves using the SPP and DN confidence measures, respectively. The performance of the baseline system on the protocol $P2$ is shown in Figures 5.4 and 5.5. Table 5.1 reports the performance in term of EER using both protocols.

PROTOCOL	CONF. MEASURE	LPP	$CLPPR$	$ULPPR$
P1	STANDARD POST. (SPP)	12.5%	36.6%	11.5%
	DOUBLE NORM. (DN)	13.1%	35.0%	12.9%
P2	STANDARD POST. (SPP)	21.0%	43.6%	19.4%
	DOUBLE NORM. (DN)	22.4%	42.9%	22.3%

Table 5.1. *The EER of the HMM/ MLP based UCP-SV baseline system using different scoring criteria with standard posterior probability (SPP) and double normalization (DN) CM (see Section 2.7.3 for a description of* $P1$ *and* $P2$*).*

These results show that:

- The HMM/MLP based UCP-SV baseline system is not competitive with the HMM/GMM based UCP-SV system.

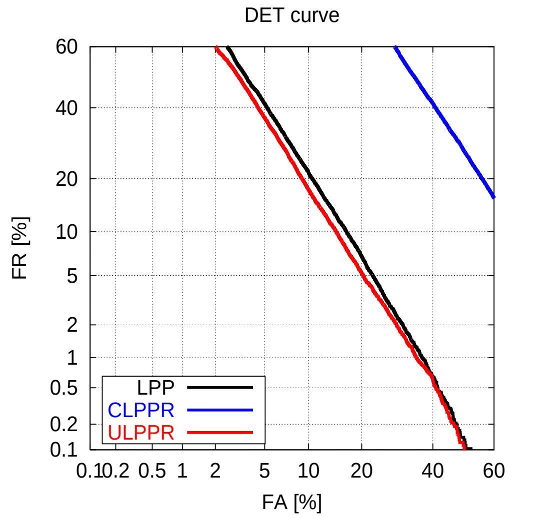

Figure 5.2. *DET curves showing the performance of the baseline HMM/MLP system using P1 with the standard posterior probability CM to estimate the verification score.*

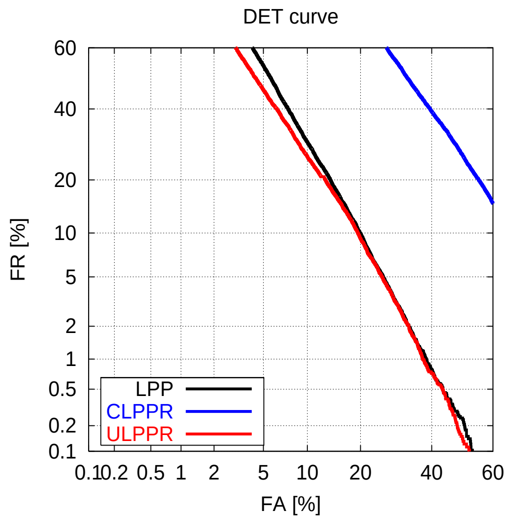

Figure 5.3. *DET curves showing the performance of the baseline HMM/ANN system using P1 with the double normalization CM to estimate the verification score.*

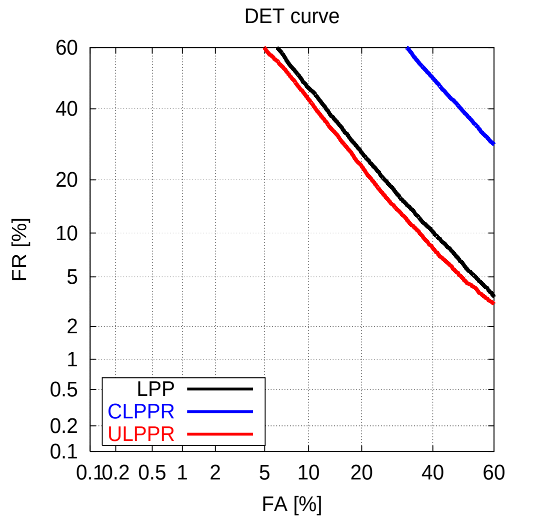

Figure 5.4. *DET curves showing the performance of the baseline HMM/ANN system using P2 with the standard posterior probability CM to estimate the verification score.*

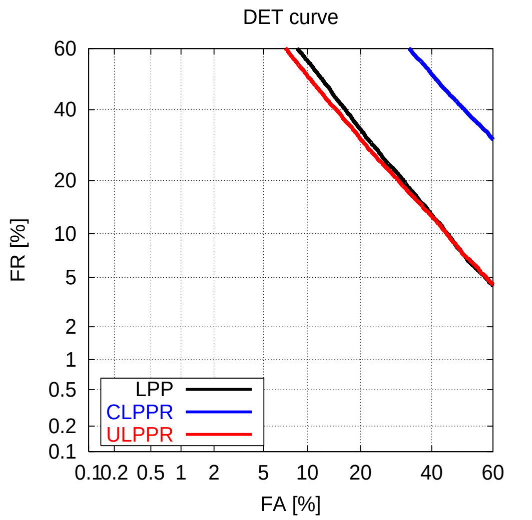

Figure 5.5. *DET curves showing the performance of the baseline HMM/ANN system using P2 with the double normalization CM to estimate the verification score.*

- LPP: *log posterior probability.*
- CLPPR: *Constrained log posterior probability ratio.*
- ULPPR: *Unconstrained log posterior probability ratio.*

- Surprisingly, and as opposed to what was usually observed in speech recognition task (Bernardis and Bourlard, 1998; Mengusoglu and Ris, 2001), results obtained using SPP confidence measure outperform those obtained using DN confidence measure.

- Compared to the use of LPP scoring criterion, the use of $ULPPR$ provides some improvement in EER on both protocols $P1$ and $P2$ and with both confidence measures. While the use of $CLPPR$ drops the performance significantly, even on the protocol $P2$ where we have expected better results than those obtained using $ULPPR$.

Analysis

To further analyze these results, we have plotted the distribution of the LPP, $CLPPR$ and $ULPPR$ scores using standard posterior probability (Figure 5.6) and double normalization (Figure 5.7) CMs. For clarity sake and without losing (much) generality, only male customers using the word *"cinema"* as password are selected. In each sub-figure, scores corresponding to customer accesses with the expected password (C-EP), customer accesses with invalid passwords (C-IP), male impostor accesses with the expected password (I-EP(M)), female impostor accesses with the expected password (I-EP(F)), male impostor accesses with invalid passwords (I-IP(M)), and female impostor accesses with invalid passwords (I-IP(F)) are plotted. To compensate the difference in dynamic range between different scores and to make the comparison easier, we mapped each of these scores to the $[0, 1]$ interval using a sigmoid function (Jourlin *et al.*, 1997):

$$s_{new} = \frac{1}{1 + \exp(\delta - s_{old})} \tag{5.14}$$

where s_{new} and s_{old} are the verification score after and before the mapping and δ is the threshold.

- *The use of double normalization:*
  The distribution of test access scores using the double normalization CM are plotted in Figure 5.7. Compared to the use of standard posterior probability CM (Figure 5.6), it appears that the use of DN gives lower posterior estimates to test accesses even for those belonging to the customer with expected password (i.e., many customer accesses with the expected password matched poorly the customer HMM model). It has been shown that the double normalization is more sensitive to the misrecognized phones making it a useful measure for rejecting utterances that are out-of-domain or contain speech disfluencies (Rivlin *et al.*, 1996). If we use this statement with what has been found in (Williams and Renals, 1998), that the performance of a CM at the word level depends upon the quality of the pronunciation model, we can say that many customer HMM models are not a good representative of the customer password, which increases the false rejection rate. This statement can be drawn from the DET curves plotted in Figures 5.2 to 5.5. For a given false acceptance rate, the false rejection rate using double normalization is always higher than that obtained using standard posterior, regardless the scoring criterion (i.e., LPP, $CLPPR$ and $ULPPR$).

  In the following, only results obtained with the standard posterior probability CM will be discussed.

- *The use of LPP scores:*
  The distribution of scores for test accesses using the log posterior probability (5.6) with the standard posterior CM is shown in Figure 5.6-a. We can observe that the distribution of scores for impostor accesses (both males and females) with the expected password is highly overlapped with the distribution of scores for customer accesses with the expected password. This explains why the result with the second protocol $P2$ is very poor (EER =21.0%). Moreover,

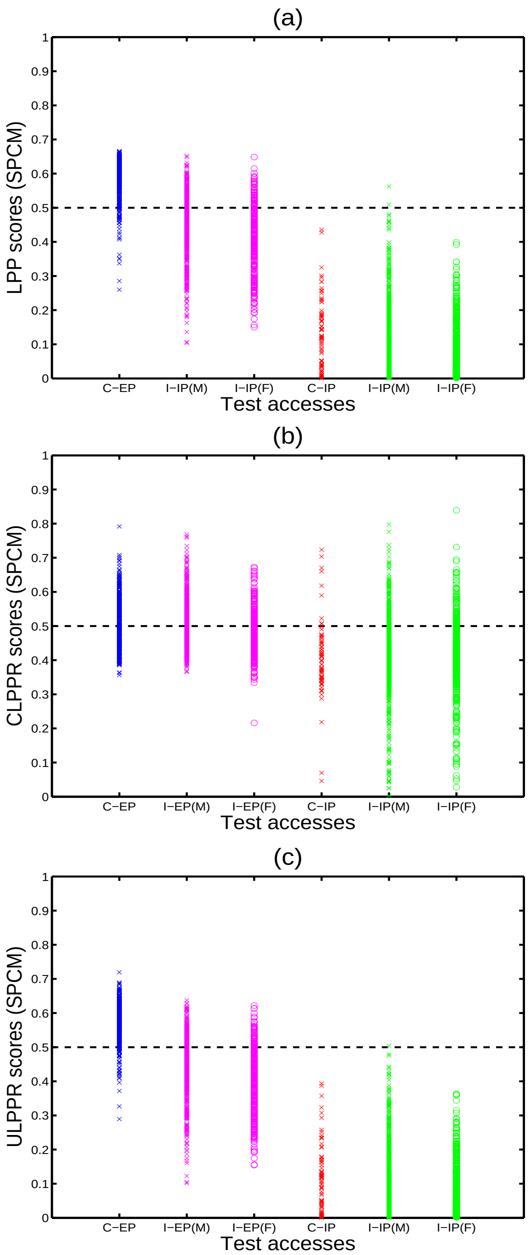

Figure 5.6. *Distribution of customer and impostor scores using (a) LPP (b) CLPPR and (c) ULPPR scoring criteria with standard posterior probability confidence measure*

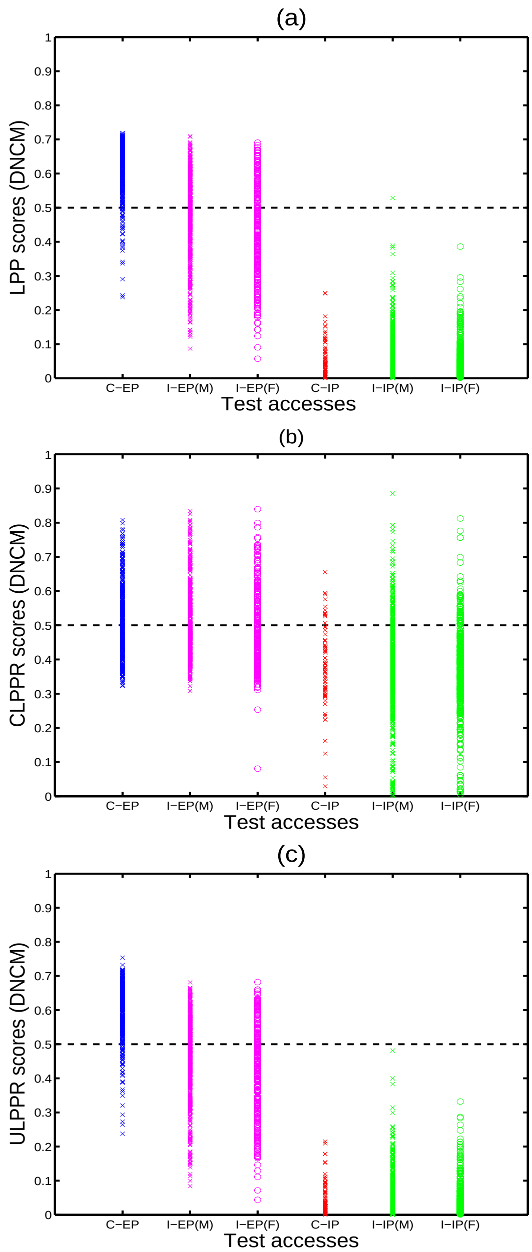

Figure 5.7. *Distribution of customer and impostor scores using (a) LPP (b) CLPPR and (c) ULPPR scoring criteria with double normalization confidence measure*

- C-EP: *Client access with the expected password.*
- C-IP: *Client access with invalid password.*
- M : *For male.*
- I-EP: *Impostor access with the expected password*
- I-IP: *Impostor access with invalid password.*
- F : *For female.*

though the frame level accuracy of the customer-dependent MLP was very high (an average of more than 95% on the cross-validation data), the discriminant capabilities of the MLP between accesses with expected passwords and accesses with invalid password are not strong. There is an overlap between the two types of accesses, even if most of test accesses with invalid passwords are below the threshold. This result is poorer than what we would normally expect for this task. A possible reason is that, not all phones are represented in the adaptation data. This point will be discussed below in more detail.

- *The use of CLPPR scores:*
  The distribution of scores for test accesses using the constrained log posterior probability ratio (5.6) with standard posterior CM are shown in Figure 5.6-b. We can see that most of the scores belong to the $[0.4, 0.7]$ interval, corresponding to the $[0.3, 1.5]$ interval in the original scores (before mapping). Consequently, any value of the threshold will result in a high EER (i.e., a small threshold will result in low FRR but a high FAR of impostors pronouncing the expected password and vice-versa). Moreover, because the original scores are quite close to zero, it means (see (5.8)), that the $\log P(\widehat{M}_c^\ell|X, \Theta_c)$ and the $\log P(\widehat{M}_c^\ell|X, \Theta)$ for most test accesses, in particular with the expected password are quite the same. This is due to the *maximum a posteriori* criterion used for MLP training (or adaptation). This point will be discussed later in Section 5.5.

- *The use of ULPPR scores:*
  The distribution of scores for test accesses using the unconstrained log posterior probability ratio with standard posterior CM are shown in Figure 5.6-c. A comparison with Figure 5.6-a shows that the normalization by $\log P(M|X, \Theta)$ gives lower scores to test accesses with mainly invalid password. This may reduce the false acceptance rate of this type of access, but not as much as we would expected.

Discussion

The poor performance of the baseline system can be ascribed to the amount and the nature of the adaptation data.

- *The amount of adaptation data:* When the amount of the adaptation data is very small, adapting entirely a huge neural network with $162,636$ parameters becomes a difficult task, and a good generalization capabilities of the network on the unseen data can not be ensured. To improve the generalization capabilities, the number of the adjusted parameters should be reduced with respect to the amount of data available for each speaker. A possible solution to this problem is to add a linear layer at the input layer of the SI-MLP and only adapt the parameters of this layer. Another possibility is to retrain the parameters of a small neural network. The use of these two techniques and the obtained results will be discussed later.

- *The nature of the data:* The adaptation data consists of a few repetitions of the same password. This password can be a long sentence containing most of the phonemes, but can also be a short word containing only a small number of phonemes, which actually corresponds to the case studied here. In the adaptation data, only examples of phonemes constitute the best phonetic transcription (referred to hereafter as the *seen* phonemes) are used for adaptation and no examples for other phonemes (referred to hereafter as the *unseen* phonemes) are available. The analysis of the variations of the average error at the outputs of the MLP during its adaptation shown that after 2 or 3 iterations the estimate of the outputs belonging to the *unseen* phonemes get down to almost zero. It means that the adapted MLP has the tendency to forget these phonemes, i.e., the MLP is biased to recognize the *seen* phonemes, which might affect negatively the performance of recognizing words different from the customer password . To alleviate this issue, a possible solution is to add some speaker-independent acoustic vectors belonging to the *unseen* phonemes (Burnett, 1997).

5.4 Improving the baseline system

Based on the prior analysis, our aim is to improve the adaptation technique, thus reducing the overlap between the distribution of customers' scores and impostors' scores. In the following, some techniques and heuristics towards reducing the effect of the limited amount and the nature of the adaptation data will be described.

5.4.1 Reducing the number of adapted parameters

Linear Input Network (LIN)

This approach (Neto *et al.*, 1995) introduces a new (trainable) linear input network to map the speaker-dependent (customer) input vectors to the speaker-independent MLP. The parameters of the additional linear layer were trained by minimizing the cross-entropy error at the output of the SI-MLP whose parameters have been frozen. Our SI-MLP has 9 frames as acoustic context, with 26 nodes for each frame, thus resulting in an additional LIN layer of 234 nodes. To better capture the characteristics of the customer, different LIN architectures were tested, using different connectivities between the additional linear layer and the input layer nodes of the SI-MLP.

1. *LIN1 : Fully connected LIN*:
   As illustrated in Figure 5.8-a, in the LIN1 architecture, all possible connections are used, i.e., all the nodes in the LIN are connected to all nodes in the input layer of SI-MLP. Consequently, the number of the adapted parameters is reduced from $162,636$ down to $54,756$.

2. *LIN2 : Frame-to-frame connections*:
   As illustrated in Figure 5.8-b, connections between the LIN input and the SI-MLP are limited to the 26 nodes of the associated frames, without inter-frame connections. As a result, the number of parameters to be adapted was significantly reduced, now equal to $(((26 \times 26) \times 9) = 6084)$. We have also tested the possibility of forcing all frames to share the same transformation matrix. In this case, the obtained results was not as good as if each frame has its own transformation Patrice.

3. *LIN3 : Node-to-node connections*:
   As illustrated in Figure 5.8-c, the LIN3 architecture limits the network to only node-to-node connections where each node in the LIN is only connected to its corresponding node in the input layer of SI-MLP, resulting in a further reduction of the number of parameters. In this case, the number of parameters to be adapted is simply equal to 234. Because the connectivity between the LIN and the input layer of the SI-MLP is weak, this might hurt the performance of the adapted MLP.

Single-Layer-Perceptron adaptation

In this approach, another MLP without hidden layer, i.e., Single-Layer Perceptron (SLP) with a set of parameters θ is introduced. Thus, the SI-MLP will be used for HMM inference, while the new SLP will be used for speaker adaptation. This SLP has 26 inputs corresponding to the dimension of the acoustic vector (i.e., without acoustic context at the input) and 36 outputs. This SLP is trained

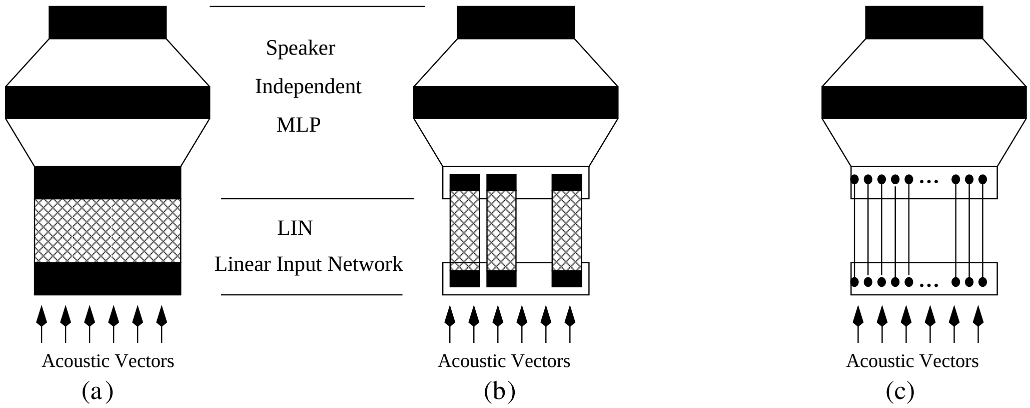

Figure 5.8. *Different types of LIN connections: (a) fully connected (LIN1), (b) frame-to-frame connections (LIN2), and (c) node-to-node connections (LIN3).*

using *PolyPhone* databases. The segmentation (desired outputs) is generated using the SI-MLP Θ. Given the limited number of parameters, the SLP performed quite poorly as a speaker-independent model, but its use for speaker adaptation has the advantage that the enrollment step is less time consuming, i.e., it converges much faster compared to the use of LIN.

Results

The results are presented in Figures 5.9 to 5.12 and Tables 5.2 and 5.3.

The DET curves in Figure 5.9 and Figure 5.10 are obtained using $P1$ and, respectively, LPP and $ULPPR$ scores. The DET curves in Figure 5.11 and Figure 5.12 are obtained using $P2$ and, respectively, LPP and $ULPPR$ scores. Tables 5.2 and 5.3 summarize the results. We did not use $CLPPR$ scores, as well as the double normalization CM, because they resulted in lower performance.

ADAPTATION TECHNIQUES	LPP	$ULPPR$
LIN1	10.2%	9.5%
LIN2	11.2%	10.1%
LIN3	18.5%	17.2%
SLP	12.2%	11.8%

Table 5.2. *The EER of the UCP-SV systems using different MLP adaptation techniques on the first protocol P1. The LPP and ULPPR scores are estimated using the standard posterior CM.*

ADAPTATION TECHNIQUES	LPP	$ULPPR$
LIN1	16.5%	15.6%
LIN2	18.5%	16.4%
LIN3	30.5%	28.6%
SLP	20.3%	19.7%

Table 5.3. *The EER of the UCP-SV systems using different MLP adaptation techniques on the second protocol P2. LPP and ULPPR scores are estimated using the standard posterior CM.*

While the use of LIN1 and LIN2 improves significantly the performance (more than 95% of confidence) on both protocols and with both LPP and $ULPPR$ scores, the use of LIN3 performs

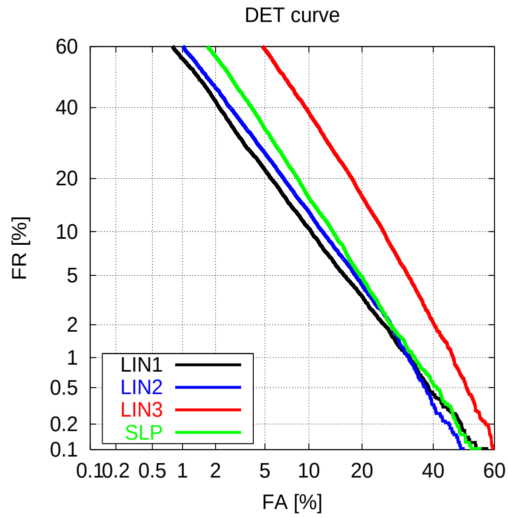

Figure 5.9. *DET curves showing the performance of different UCP-SV systems with different MLP adaptation techniques on the protocol P1 using LPP scores with the standard posterior confidence measure.*

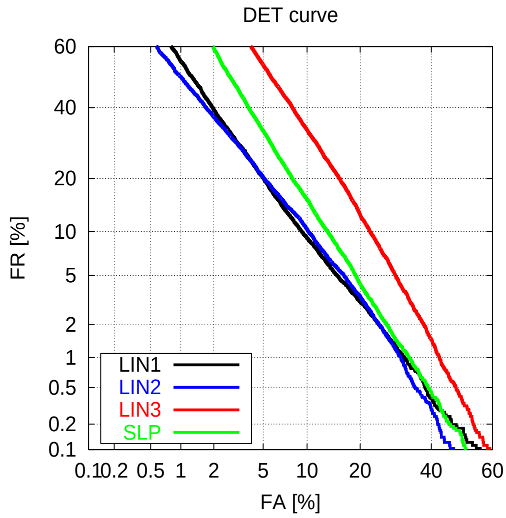

Figure 5.10. *DET curves showing the performance of different UCP-SV systems with different MLP adaptation techniques on the protocol P1 using ULPPR scores with the standard posterior confidence measure.*

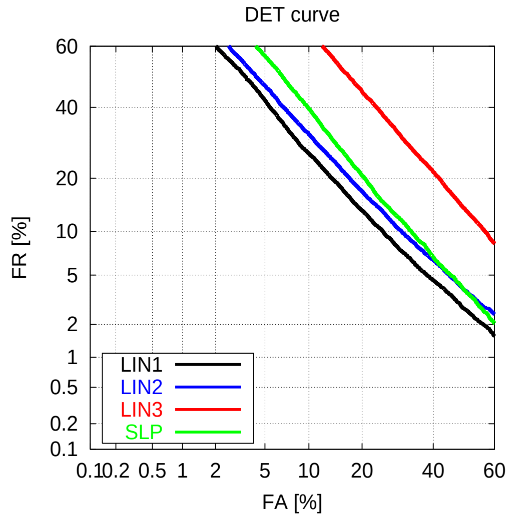

Figure 5.11. *DET curves showing the performance of different UCP-SV with different MLP adaptation techniques on the protocol P2 using LPP scores with the standard posterior confidence measure.*

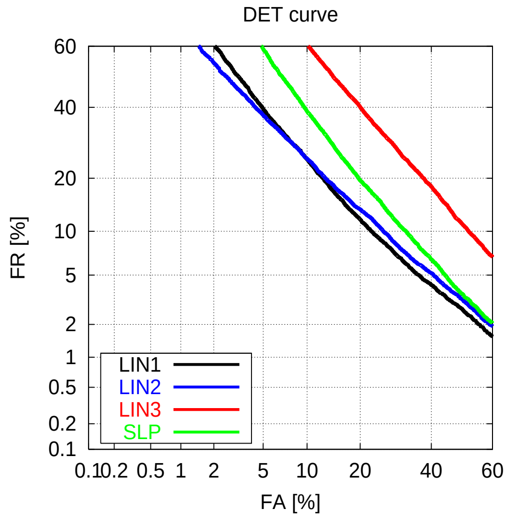

Figure 5.12. *DET curves showing the performance of different UCP-SV systems with different MLP adaptation techniques on the protocol P2 using ULPPR scores with the standard posterior confidence measure.*

worse than the baseline system in all cases. The use of the customer-dependent SLP gives the same performance as the baseline system.

These results indicate that, the neural network adaptation is a trade-off between the number of parameters to be adapted and the modeling capacity of the adapted neural network. That is, reducing the number of the adapted parameters can be effective as long as the modeling capacity of the neural network is not affected. LIN1 and LIN2 performed better than LIN3 and SLP, because the connectivity between nodes in LIN1 and LIN2 is more complex than the connectivity in LIN3 and SLP[2].

5.4.2 Enrichment of the adaptation data

One of the most important factors to ensure a good generalization capability of the neural network, is the availability of examples for all phonemes (classes) in the training or adaptation data. In UCP-SV, the speech data associated with the customer password contains -only- a few number of phonemes. Hence, some (unseen) phonemes will not have any example available in the adaptation data which might degraded the performance of the MLP.

To alleviate this problem, we have enriched the adaptation data provided by each customer with some speaker-independent acoustic vectors belonging to the unseen phonemes. These examples are chosen randomly (without using of any selection criterion such as the gender of the customer) from the *PolyPhone* database. To have a good balance, the number of the added examples for each phoneme was chosen to be equal to the average number of examples per phoneme in the original adaptation data.

This heuristic is practical and efficient for MLPs that do not take into account the acoustic context in the input layer (i.e., the number of input nodes is equal to the acoustic vector dimension). Otherwise, the number of examples to be added for each unseen phoneme will increase, since we have to choose examples with different acoustic context. For this reason, only the adaptation of the SLP is considered.

It is worth mentioning that this procedure should at least reduce the false acceptance rate corresponding to test accesses with invalid passwords.

Results

Table 5.4 reports the obtained EER on both protocols $P1$ and $P2$ before and after the enrichment of the adaptation data. The performance is significantly improved compared to the best adaptation techniques described before in Section 5.4.1.

Regarding the use of SLP, on the protocol $P1$, the EER dropped from 12.2% to 6.6% using LPP scores and from 11.8% to 6.6% using $ULPPR$ scores, giving a relative improvement of 45.9% and 44%, respectively. With the second protocol $P2$, the benefit of this approach is even more significant, with a relative improvement of 49.2% and 47.2%, respectively.

Analysis

To analyze the effect of the modified adaptation procedure on the generalization properties of the customer adapted SLP, and to understand from where this improvement comes from, we have plotted the distribution of the LPP and $ULPPR$ scores derived from the SLP outputs before (Figure 5.13) and after (Figure 5.14) the enrichment of the adaptation data.

[2]We have also tried the use of an SLP with 9 frames as acoustic context. The performance was slightly better than that obtained by an SLP without input context (i.e., the one used in this thesis), but not as good as LIN1 and LIN2. We preferred to keep using the SLP without input context, because it converges much faster and the adaptation will be easier when some speaker-independent data is added (see Section 5.4.2).

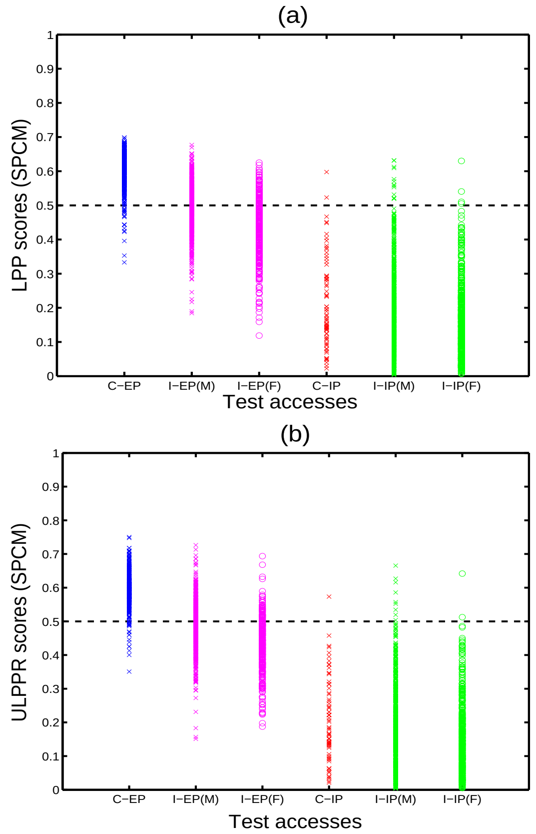

Figure 5.13. *Distribution of customer and impostor scores using (a) LPP and (b) ULPPR scores with standard posterior (SP). An SLP is used for adaptation without adding examples for unseen phonemes to the adaptation data.*

- C-EP: *Customer access with the expected password.*
- C-IP: *Customer access with invalid password.*

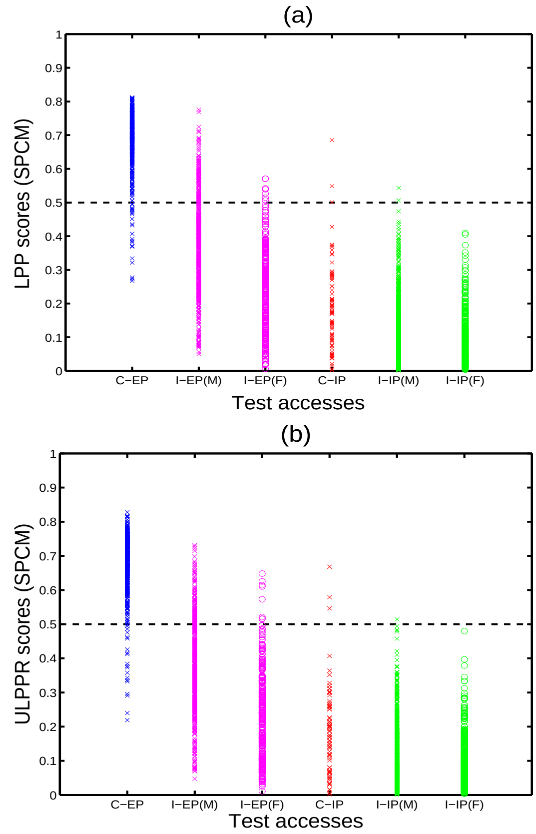

Figure 5.14. *Distribution of customer and impostor scores using (a) LPP and (b) ULPPR scores with standard posterior. An SLP is used for adaptation. the adaptation data is enriched with speaker-independent examples corresponding to the unseen phonemes.*

- I-EP: *Impostor access with the expected password*
- I-IP: *Impostor access with invalid password.*

Protocols	Add. examples	*LPP*	*ULPPR*
P1	Before	12.2%	11.8%
	After	6.6%	6.6%
P2	Before	20.3%	19.7%
	After	10.3%	10.4%

Table 5.4. *The EER of the HMM/SLP based UCP-SV system before and after the enrichment of the adaptation data. LPP and ULPPR are estimated using standard posterior CM.*

From these two figures, we can conclude that:

- When some speaker-independent data is added to the adaptation data provided by the customer, the adapted SLP gives lower matching score to test access with invalid password. This indicates that the SLP learned some acoustic characteristics of the unseen phonemes, which reduces the false acceptance of accesses with invalid passwords. This reduction is more important in case of female accesses than male accesses.
- Surprisingly, and compared to test accesses pronounced by males, the false acceptance of females pronouncing the expected password is significantly reduced. This means that the discriminative capabilities of the adapted SLP between (at least) males and females accesses are significantly improved. We should remember here that for the analysis, all customers are males. A possible interpretation of this result is that, in the adapted (with the modified procedure) customer SLP, there are two distinct types of parameters: (1) The parameters (weights) between input nodes and output nodes associated with the *seen* phoneme which, actually, learned some customer (male) specific characteristics, and (2) the parameters between input nodes and output nodes associated with the *unseen* phonemes which, actually learned some speaker-independent characteristics.

  Because female characteristics are closer to speaker-independent than male characteristics, for a given test utterance pronounced by a female speaker, the posteriori probabilities of the *unseen* phonemes will be higher than those estimated by a customer SLP adapted using only customer enrollment data. Which is equivalent to reducing the posteriori probabilities of the *seen* phonemes. As the forced Viterbi decoding used local posterior probabilities associated with only *seen* phonemes, this reduced the matching score of female accesses with both expected and invalid passwords.

5.4.3 Embedded training

For further improvement, we have performed an embedded adaptation procedure. The adapted customer SLP θ_c is used to estimate the phone posterior probabilities, which are used in a forced Viterbi alignment on the customer HMM model $\widehat{M_c^\ell}$ to generate a new segmentation used for one more adaptation step. Table 5.5 reports the obtained results.

PROTOCOLS	EMBEDDED ADAPTATION	LPP	ULPPR
P1	BEFORE	6.6%	6.6%
	AFTER	6.3%	6.3%
P2	BEFORE	10.3%	10.4%
	AFTER	9.7%	9.7%

Table 5.5. *The EER of the optimized HMM/ SLP based UCP-SV baseline system obtained using an embedded adaptation.*

Results show that an embedded adaptation can yield to some further improvement of the HMM/MLP UCP-SV system. This improvement is 95% and 90% significant with $P1$ and $P2$, respectively.

5.5 Discussion

Despite different improvements brought to the baseline system, one can note that the use of hybrid HMM/MLP systems for speaker verification is not effective. While for speech recognition task, these systems have shown comparative results with the state-of-the-art HMM/GMM systems, the performance of the UCP-SV system based on the hybrid HMM/MLP is significantly worse than that obtained by the state-of-the-art HMM/GMM systems using the same HMM inference procedure (i.e., the scaled likelihood as HMM phone probabilities). It is important to note that the HMM model and the MLP model are trained or adapted with two different criteria.

The HMM model is trained to maximize the likelihood of the acoustic enrollment utterances (i.e., modeling the speech signal). Because the use of raw likelihood score estimated by the speaker dependent model can not be used as a good confidence score for speaker verification, a likelihood ratio test is introduced, where the likelihood of the test utterance estimated by a speaker model is divided by the likelihood estimated by a background model.

The MLP is trained to maximize the posterior probabilities of phone classes associated with HMM states given an acoustic vector (i.e., ensuring that the correct phone is the most probable for each frame) (Renals *et al.*, 1994). For a speech recognition task, the phone posterior probabilities are shown to be a good confidence measure that can be used as they are without normalization [3]. However, their use for a speaker verification task appeared to be ineffective, even if we use score normalization techniques developed in the HMM/GMM framework.

Indeed, given the small amount of adaptation data, the MLP adaptation procedure consisted of slightly shifting the original decision boundaries between phone classes determined during the SI-MLP training. As the MLP is adapted to discriminate between different phone classes and not between customer and impostors, this shifting significantly improves the frame recognition rate of customer utterances with the expected password, but with no direct negative impact on posterior probabilities of impostor accesses with the expected password. That is, the estimated posterior probabilities give no information on how likely an acoustic vector belongs to a customer or impostors (Genoud *et al.*, 1999b,a). That is the maximum a posteriori probabilities are more dependent on the speech content than the speaker characteristics. Based on this statement, and assuming that we do not include the *a priori* probabilities during the score estimation, the two probabilities $P(\widehat{M}_c^\ell|X,\Theta_c)$ and $P(\widehat{M}_c^\ell|X,\Theta)$ should be equal, thus meaning that the $CLPPR$ score should be equal to 0. Since our models (neural networks and the inferred HMM models) are not perfect (due to insufficient amount of training data), the estimated posterior probabilities are not accurate and what we obtain is two probabilities that are quite close to each other (see Section 5.3.4). This is why the use of $CLPPR$ as speaker verification score gave poor results.

In conclusion, we can say that in the hybrid HMM/MLP framework, the phone posterior probabilities estimated by a speaker-dependent adapted MLP are more representative of the lexical content of the utterance than of the speaker characteristics, making the use of HMM/MLP systems for a speaker verification task less effective. Any attempt to make them more effective using score normalization techniques as done in standard HMM/GMM framework or using some confidence measure, is useless and may even degrade the speaker verification performance.

5.6 Modeling speaker information

To alleviate this problem, a new approach combining the advantages of the hybrid HMM/MLP systems and Gaussian mixture models, in the same probabilistic framework was proposed (Ben-Zeghiba and Bourlard, 2002, 2003a). The hybrid HMM/MLP model was adapted to learn the lexical content of the customer password, and was used mainly for *utterance verification* (UV), while a text-independent state-of-the-art GMM was adapted to capture the customer characteristics, and was used mainly for *speaker verification* (SV).

[3] It has been demonstrated (Morgan and Bourlard, 1995) that the use of posterior probabilities estimated by an MLP is equivalent to the use of likelihood ratio usually used for utterance verification.

5.6.1 Joint decision rules

Let us start from the posteriori probability based UCP-SV decision rules, i.e;

$$S = S_c \quad \text{if} \quad P(M_c, S_c|X) \geq P(M_c, \overline{S_c}|X) \tag{5.15}$$

and

$$P(M_c, S_c|X) \geq P(\overline{M_c}, S|X) \tag{5.16}$$

Using the conditional probability rule, decision rules (5.15) and (5.16) can be rewritten as follows:

$$S = S_c \quad \text{if} \quad P(M_c|S_c, X).P(S_c|X) \geq P(M_c|\overline{S_c}, X).P(\overline{S}_c|X) \tag{5.17}$$

and

$$P(M_c|S_c, X).P(S_c|X) \geq P(\overline{M_c}|S, X).P(S|X) \tag{5.18}$$

Using Bayes rule with the assumption that all speakers have the same *a priori* probability, decision rules (5.17) and (5.18) can be expressed as follows:

$$\left[\frac{P(M_c|S_c, X)}{P(M_c|\overline{S}_c, X)}\right]\left[\frac{p(X|S_c)}{p(X|\overline{S}_c)}\right] \geq \Delta_1 \tag{5.19}$$

$$\left[\frac{P(M_c|S_c, X)}{P(\overline{M}_c|S, X)}\right]\left[\frac{p(X|S_c)}{p(X|S)}\right] \geq \Delta_2 \tag{5.20}$$

where Δ_1 and Δ_2 are the thresholds.

The first term in the left hand side in (5.19) and (5.20) are, respectively, the constrained posterior probability ratio defined in (5.7) and the unconstrained posterior probability ratio defined in (5.6) with the assumption that $\overline{M}_c$ and M are representing the same ergodic HMM model. They will be used mainly for *utterance verification*.

The second terms $\frac{p(X|S_c)}{p(X|\overline{S}_c)}$ and $\frac{p(X|S_c)}{p(X|S)}$ are the *likelihood ratios* usually used in conventional text-independent speaker verification and actually represent the contribution of the speaker characteristics. This contribution will be estimated through usual Gaussian Mixture Model-Universal Background Model (GMM-UBM). These scores will be used for *speaker verification*.

Together, these two scores (i.e., posterior probability and likelihood ratio) give us the information about the speaker and the pronounced word and on which the decision will be made to accept or reject a speaker pronouncing a specific password.

The posterior probabilities are estimated through a neural network, which is trained (or adapted) in a discriminative way (unlike the maximum likelihood). It has been found (for speech recognition) that these posterior probabilities are equivalent to the likelihood ratio usually used for utterance verification (Gold and Morgan, 2000; Morgan and Bourlard, 1995). So, taking the ratio of two posterior probabilities estimated by two different neural network is not useful. Hence, the two posterior probability ratios in (5.19) and (5.20) will be estimated using the same neural network (the adapted speaker dependent neural network in occurrence). Thus yielding the following simplifications:

$$\frac{P(M_c|S_c, X)}{P(M_c|\overline{S}_c, X)} = 1 \tag{5.21}$$

and

$$\frac{P(M_c|S_c, X)}{P(M|S, X)} = \prod_{t=1}^{T} \frac{P(q_k^{(t,\ell)}|x_t, \theta_c)}{P(q_{best}^{t}|x_t, \theta_c)} \tag{5.22}$$

where T is the length of the utterance X and

$$P(q^t_{best}|x_t,\theta_c) = \max_{1\leq k\leq K} P(q^t_k|x_t,\theta_c) \tag{5.23}$$

where K is the number of phones.

Assuming that the transition probabilities are equal (Which is generally the case in HMM/MLP speech recognition systems), $\frac{P(q_k^{(t,\ell)}|x_t,\theta_c)}{P(q^t_{best}|x_t,\theta_c)}$ represents the posterior probability of being in the decoded (according to the forced Viterbi alignment) state q_k at time t given the frame x_t divided by the best posterior probability of that frame at the time t. This confidence measure is called Relative Posterior Confidence Measure (RPCM) (Mengusoglu and Ris, 2001) and it tells us how close is the score of the decoded word to the best acoustic score. This RPCM criterion was found to be efficient for rejecting the Out-Of-Vocabulary (OOV) words. If a word is correctly recognized (i.e., the decoded phone at each time has the best local posterior probability, even if it is not high), this RPCM will be equal to 1, corresponding to the first case (5.21).

Substituting (5.21) and (5.22) into (5.19) and (5.20) respectively, and taking the logarithm, decision rules (5.19) and (5.20) can be rewritten as follows:

$$\log\left[\frac{p(X|S_c)}{p(X|\overline{S}_c)}\right] \geq \delta_1 \tag{5.24}$$

$$\sum_{t=1}^{T}\log\left[\frac{P(q_k^{(t,\ell)}|x_t,\theta_c)}{P(q^t_{best}|x_t,\theta_c)}\right] + \log\left[\frac{p(X|S_c)}{p(X|S)}\right] \geq \delta_2 \tag{5.25}$$

As explained in Section 4.2, a weighted sum combination technique is used to estimate the final score. If we refer to the scores in (5.24) and (5.25) as s_2 and s_1, respectively, the combined score s_{com} can be written as follows:

$$s_{com} = \alpha s_1 + (1-\alpha)s_2 \tag{5.26}$$

and $0 \leq \alpha \leq 1$.

As done in the previous chapter, S and $\overline{S}_c$ are represented by the same GMM model, referred to as "world model". By expending (5.26) and normalizing by the length T of the test utterance, we obtain the following decision rule to accept a speaker:

$$\alpha\left(\frac{1}{T}\sum_{t=1}^{T}\log\left[\frac{P(q_k^{(t,\ell)}|x_t,\theta_c)}{P(q^t_{best}|x_t,\theta_c)}\right]\right) + \frac{1}{T}\log\left[\frac{p(X|S_c)}{p(X|\overline{S}_c)}\right] \geq \delta \tag{5.27}$$

The parameter α indicates how much is the contribution of the posterior probability score in the final decision. As we can see, the weight of the log likelihood ratio (related to the speaker verification) is equal to 1, indicating the importance of the GMM score in the final decision.

It is important to note here that the RPCM score does not contain any information about the speaker. It might happen that for all frames, the two local posterior probabilities $P(q_k^{(t,\ell)}|x_t,\theta_c)$ and $P(q^t_{best}|x_t,\theta_c)$ will be equal even if the corresponding value is not high. Hence, the RPCM is completely speech dependent. We have compared the RPCM with the standard posterior CM (SPCM) where $\left(\sum_{t=1}^{T}\log\left[\frac{P(q_k^{(t,\ell)}|x_t,\theta_c)}{P(q^t_{best}|x_t,\theta_c)}\right]\right)$ is replaced by $\left(\sum_{t=1}^{T}\log P(q_k^{(t,\ell)}|x_t,\theta_c)\right)$.

5.6.2 Speaker acoustic modeling

For each customer, an adapted GMM is created in addition to the previous adapted SLP. The GMM adaptation consisted of adapting only the mean parameters, of Gaussians of a speaker independent GMM model (referred to as “world model”) with 120 diagonal covariance matrix. The world model is trained on *PolyPhone* databases. Both the training and the adaptation of the GMM are performed using only speech segments. The speech/silence segmentation is obtained using a bi-Gaussian model of the log energy distribution as described in Section 2.6. The adaptation is performed using a simplified version of MAP adaptation algorithm (4.9).

5.6.3 Results and discussion

Figure 5.15 shows the EER variations of the UCP-SV system and the TD-SV system as a function of the combined parameter α, using the first protocol $P1$. Table 5.6 reports EERs of both systems using both protocols $P1$ and $P2$. Results show that all UCP-SV systems perform comparably and do not exhibit any significant improvement compared to the use of only likelihood ratio derived from GMMs.

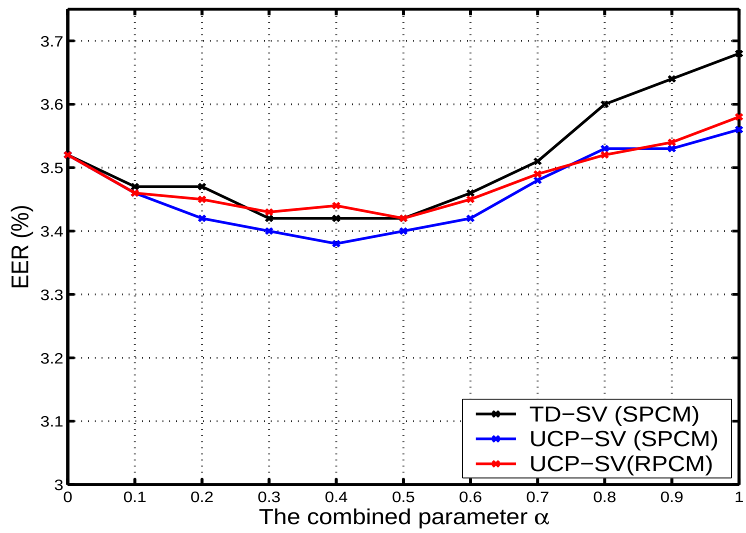

Figure 5.15. *EER variations as a function of the combined parameter α for the UCP-SV (HMM/SLP+GMM-UBM) and TD-SV systems. Both standard posterior (SPCM) and relative posterior (RPCM) confidence measures are used.*

PROTOCOLS	CONF. MEASURE	α	EER [%]
P1	TD-SV (SPCM)	0.5	**3.4%**
	UCP-SV (SPCM)	0.4	**3.4%**
	UCP-SV (RPCM)	0.5	3.4%
	GMM-UBM (LLR)	0.0	**3.5%**
P2	GMM-UBM (LLR)	-	**5.3%**

Table 5.6. *EER of the combined (HMM/SLP + GMM-UBM) UCP-SV system using different scoring criteria with standard posterior and relative posterior CM to estimate the utterance verification score. The EER obtained using only GMM-UBM is also reported.*

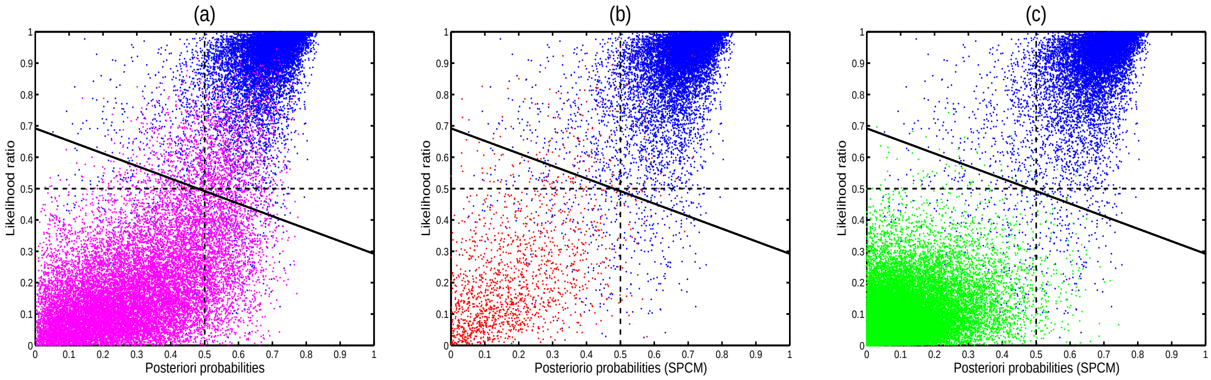

Figure 5.16. *Distribution of the posteriori probability against likelihood ratio: Blue points correspond to customer accesses with expected password (C-EP), magenta points correspond to impostor accesses with expected password (I-EP), red points correspond to customer accesses with invalid password (C-IP) and green points correspond to impostor accesses with invalid password (I-IP).*

To analyze these results, we have plotted in Figure 5.16 the distribution of the *posterior probability* scores using SPCM against the *likelihood ratio* scores estimated by the speaker verification part (i.e., GMM-UBM). To compensate the difference in the dynamic range between the two scores, we mapped them to the $[0, 1]$ interval using (5.14).

In each figure, the distribution of scores is divided into four regions using vertical and horizontal dashed lines corresponding to the individual posterior probability and likelihood ratio thresholds. The upper-right region $R1$ corresponds to the case where the speaker is accepted by both the utterance verification (UV) (i.e., high posterior probability) and speaker verification (SV) (i.e., high likelihood ratio) parts. The bottom-right region $R2$ corresponds to the case where the speaker is accepted by the UV part and rejected by the SV part. The bottom-left region $R3$ corresponds to the case where the speaker is rejected by both UV and SV parts. The upper-left region $R4$ corresponds to the case where the speaker is rejected by the UV part and accepted by the SV part. The diagonal line corresponds to the decision boundary as found by the UCP-SV system. From these figures, we can conclude that:

1. Many impostors accesses (Figure 5.16-a, regions $R1$ and $R2$, magenta points), with the expected password are accepted as a customer by the HMM/SLP, confirming that the neural network mainly modeled the lexical content of the password.

2. Most of impostor accesses with invalid passwords (Figure 5.16-c, region $R3$, green points) have a low likelihood ratio score and low posteriori probability score. So, they are rejected by both UV and SV part, making this approach very robust to such situation.

3. Surprisingly, most of customer accesses with invalid passwords (Figure 5.16-b, region $R3$, red points) have a low likelihood ratio score and therefore they are rejected. This indicates that the customer-dependent GMM did not properly capture all the customer characteristics. One possible explanation is that the customer password is short, hence, the phonemic content of the adaptation data is very poor. The customer-dependent GMM partially kept the speaker characteristics extracted from those phonemes, which are not sufficient to properly model all the customer characteristics. As a results, the adapted GMM becomes not only customer-dependent but also text-dependent. As a consequence, the decision made by the combined UCP-SV system mainly uses the likelihood ratio estimated by the speaker verification part (i.e.,GMM-UBM). This explains why the combined UCP-SV system (3.4%) gave no improvement compared to the use of GMM part only (3.5%).

4. Both HMM/SLP and GMM are adapted with two different criteria. Hence, the posteriori probability and likelihood ratio scores might contain some complementary information that can be useful to improve the performance of the combined UCP-SV system. The contribution of the posteriori probability is determined by the value of the parameter α (5.26). Because in the combined UCP-SV system, the GMM is speaker and text-dependent, it does some work that the HMM/SLP supposed to do by giving a low likelihood ratio to the customer accesses with the invalid password. This makes the contribution of the posterior probability less important in the verification score. This is why the value of the parameter α is small.

5.7 Conclusion

This chapter has investigated and discussed the use of hybrid HMM/MLP for UCP-SV. Two issues were addressed: score normalization and MLP adaptation.

A posteriori probability based score normalization techniques, similar to those developed in standard HMM/GMM based speaker verification, were tested. We have found that these normalization techniques are not useful and can degrade the performance of the UCP-SV system significantly. The amount of the data provided by the customer is very small, hence, making the phonemic coverage sparse. Therefore, reducing the number of the MLP parameters to be adapted and adding examples of phonemes that are not in the adaptation data are the two main properties of the MLP adaptation techniques we have tested. Despite the significant relative improvement that was obtained ($\approx 46\%$ with $P1$ and $\approx 50\%$ with $P2$ relative improvement), the resulting performance was still below expectations.

The conclusion of this investigation is that the phone posterior probabilities estimated by the adapted MLP reflect the lexical content of the utterance rather than the speaker characteristics. Consequently, they are not well suited to perform speaker verification.

Based on this investigation, a new framework where HMM/MLP systems are combined with a GMM-UBM based text-independent speaker verification is proposed. The combined system resulted in significant improvements of the performance of the UCP-SV system, but showed no significant improvement over the performance using only GMM-UBM. It was shown that the reason for this is that when a GMM is trained with speech of limited phonemic coverage (which is the case here), it becomes both speaker and text dependent.

We should noted here that when the passwords are sentences, the GMM-UBM based speaker verification part will behave as a text-independent speaker verification. In this case, the contribution of the HMM/MLP will have more weight.

Chapter 6

Joint Speech and Speaker Recognition

6.1 Introduction

Speech signal conveys (among other things) two important types of information, the speech content (text) and the speaker identity. Speech recognizers aim to extract the lexical information from the speech signal independently of the speaker by reducing the inter-speaker variability. Speaker recognizers aim to recognize the speaker's identity from the speech signal. Consequently, both recognizers should have different speech analysis component to extract the useful information and discard the others. In practice, this is seldom true. Indeed, both recognizers use the same acoustic features. However, these acoustic features are represented or modeled differently. Speech recognizers try typically, to model the phonetic variations (discriminate between phonemes), while speaker recognizers try to model the general speech without focusing on its phonemic content. As a results, the outputs (scores, recognized text and speaker's identity) of both recognizers might contain some complementary information that can be useful to improve the performance of each of the recognizers independently or the joint speech and speaker recognition performance.

The combination and the integration of the outputs of speech and speaker recognizers have several potential applications, such as:

1. Speaker identification can be used as a front-end processor to select the speech recognizer (Reynolds and Heck, 1991).
2. Speaker identification can be used to help guide the speech recognizer search for the identity claim. (Heck, 2002).
3. Automatic recognition of co-channel speech, where more than one speaker are speaking at the same time (Heck, 2002).
4. Performing continuous speaker recognition and knowledge/content recognition, such as speech biometric (Maes, 1999), verbal information verification (Li *et al.*, 2000) and conversational spoken dialog systems (Hazen *et al.*, 2003).

This chapter investigates a new probabilistic approach that maximizes the joint posterior probability of the pronounced word and the speaker identity given the observed data (BenZeghiba and Bourlard, 2003b, 2004c). This probability can be expressed as a product of the posterior probability of the pronounced word estimated through a speaker-dependent MLP and the likelihood of the data estimated through a speaker-dependent GMM. More precisely, we use posteriori probability scores to improve the performance of a likelihood based speaker recognizer. We thus end-up with a joint model that can be used for text-independent speaker identification and for speech recognition and mutually benefiting from each other.

The evaluation of this approach is examined in two applications: speaker identification (for both closed-set and open-set tasks) and speaker clustering. Also, the proposed approach will be compared with two other conventional approaches in both applications.

6.2 Task description

This section describes two practical tasks where the use of the proposed approach could be beneficial.

6.2.1 Speaker identification enhanced by ASR

There are some applications where all registered speakers share the same set of commands (words) and where each command is associated with a specific service. To make the application more user friendly, a convenient way is to let customers access the system by just pronouncing the command associated with the service. For example, for a command like *"get my messages"*, the system should display or play only messages for the current speaker. Hence, the task of the system is to simultaneously recognize the command and the speaker's identity. An unauthorized speaker should be detected and rejected. A typical system for this application is to use a speaker-independent speech recognizer to recognize the command and a speaker recognizer to identify the speaker. The joint speech and speaker recognition performance of the application depends on the individual performance of speech and speaker recognizers. To reduce the errors made by this system, we should enhance the performance of the speech recognizer, speaker recognizer or both. A typical way to improve the speech recognizer performance is to use a speaker-dependent speech recognizer, while for speaker recognizer we have to use another system with better performance. In the approach proposed here, we take advantages from the improvement obtained by the speech recognizer to improve the speaker identification performance. Hence, the joint speech and speaker recognition performance will be improved.

6.2.2 Speaker clustering enhanced by ASR

To improve the performance of speech recognizers, speaker clustering techniques are introduced (Mathan and Miclet, 1990; Abdulla and Kasabov, 2001). The general idea is that similar speakers are grouped together into the same cluster according to a certain criterion such as gender, dialect, accent,...etc, and their data is used to train one speech recognizer that characterizes this cluster (these speakers).

During recognition, the usual approach is to run the speech recognizer associated with each cluster and then select the recognizer with the highest score. While this procedure is accurate, it is time consuming. To reduce this time, a common way is to assign the test speaker to the cluster which is acoustically close according to a certain criterion, then the recognition will be performed using the speech recognizer belonging to the selected cluster.

Unfortunately, such a procedure is not optimal. The speech recognition performance depends on the performance of the clustering approach and selecting the closest cluster does not guarantee that its corresponding speech recognizer is the best for speech recognition. To alleviate this problem, we propose a new criterion where the selected cluster is the one which is acoustically close to the test speaker in the joint cluster and speech space.

6.3 Problem Statement

In the applications targeted here, our goal is to find the word (command) $\widehat{W}$ from a finite set of possible words $\{W\}$ and the speaker $\widehat{S}_c$ from a finite set of registered speakers $\{S_c\}$ that maximize the joint posterior probability $P(\widehat{W}, \widehat{S}_c|X)$, i.e.,

$$\begin{aligned} (\widehat{W}, \widehat{S}_c) &= \underset{\{W, S_c\}}{\text{argmax}} \; P(W, S_c|X) \\ &= \underset{\{W, S_c\}}{\text{argmax}} \; [P(W|S_c, X) P(S_c|X)] \end{aligned} \tag{6.1}$$

Taking the logarithm, and using Bayes rule with the assumption that the *a priori* probability of the speaker $P(S_c)$ is uniform over all speakers, (6.1) can be rewritten as:

$$(\widehat{W}, \widehat{S}_c) = \underset{\{W, S_c\}}{\text{argmax}} \; [\log P(W|S_c, X) + \log p(X|S_c)] \tag{6.2}$$

The first term, $\log P(W|S_c, X)$, corresponds to the log posterior probability of the word W estimated in our case through a speaker-dependent hybrid HMM/MLP with parameters Θ_c as follows (and under the assumption that feature vectors are independent):

$$\log P(W|S_c, X) = \frac{1}{T} \sum_{t=1}^{T} \log P(q_k^t|x_t, \Theta_c) \tag{6.3}$$

where q_k^t represents the "optimal" state q_k decoded at time t along the Viterbi path, and T is the length of X after removing the decoded silence frames.

The second term, $\log p(X|S_c)$, corresponds to the likelihood of the observed data X estimated by a speaker-dependent GMM model with parameters Λ_c as follows (and under the assumption that feature vectors are independent):

$$\log p(X|S_c) = \frac{1}{T} \sum_{t=1}^{T} \log p(x_t|\Lambda_c) \tag{6.4}$$

Obviously, $\log P(W|S_c, X)$ and $\log p(X|S_c)$ represent, respectively, the contribution of the speech and speaker recognition systems in the combined score.

It can be observed that using (6.2) for all registered speakers is time consuming. Therefore, similar to the work done in (Heck, 2002), we generate a list of T-best speakers according to their likelihoods (6.4) and then re-score this list using (6.2).

6.4 Database and experimental setup

The experiments were curried out using the *PolyVar* database (Chollet *et al.*, 1996). The database is divided into 2 data sets:

- *Dataset1:* A set of 19 speakers (12 males and 7 females) who were in more than 26 sessions are selected. For each speaker, the first 5 sessions [1] are used for training (adaptation). For closed set experiments, an average of 19 sessions per speaker are used as test data, resulting in a total of 6430 test utterances.

[1] One session consists of one repetition of the same set of 17 words common for all speakers (see appendix A).

- *Dataset2:* For the open set experiments, an additional set of 19 speakers with the same set of 17 words are used as impostors. There are a total of 6452 impostor test utterances.
- For speaker clustering experiments, only *Dataset2* is used (i.e., all test utterances come from non registered speakers).

We also used the *PolyPhone* database (Chollet *et al.*, 1996) to train a speaker-independent Gaussian mixture model (GMM) with a set of parameters Λ. This GMM is modeled by 240 (diagonal covariance) Gaussians and trained with the EM algorithm. It will be used only as an initial distribution for speaker adaptation and it will be referred to as *world model*.

6.5 Approaches

In the first task, the aim is to correctly recognize both the pronounced word and the speaker identity for each test utterance. The vocabulary is limited and contains 17 words. The approaches examined and compared here to perform this task use the same text-independent GMM based speaker identification system, but they have different speech recognizers and different ways to exploit speaker and speech recognizer outputs. Before describing different approaches tested here, first, a description of the speaker identification system is given.

6.5.1 Speaker identification system

The speaker identification system is a text-independent GMM based. For each registered speaker, a speaker-dependent GMM model with parameters Λ_c is adapted using MAP adaptation technique, from the world model Λ. The performance of the speaker identification system is equal to 95.9%.

6.5.2 Baseline approach

A typical system to perform the first task described in Section 6.2.1 is to use a speaker-independent speech recognizer to recognize the pronounced word and a speaker identification system to identify the speaker identity.

The speech recognizer is a hybrid HMM/MLP based. We first tried the use of the SI-MLP with parameter Θ, the one used previously for HMM inference (see Section 4.3). However, given that this SI-MLP is trained in different conditions and with a large amount of data that is not appropriate for this task (i.e., the data is task independent), the recognition rate was low. To enhance the performance of this MLP, we decided to adapt its parameters using data that is specific to our task. The adaptation consists of re-training all the parameters Θ using data (referred to as *world data*) provided by 56 speakers different from *Dataset1* and *Dataset2* with the same set of 17 words. The segmentation is obtained by forced alignment of the adaptation utterances on the hypothesized word HMM model using local posterior probabilities estimated by the SI-MLP Θ. A cross-validation technique is used to avoid overtraining. The word recognition rate of the adapted SI-MLP (Θ_1) is equal to 97.2% and 96.8% on *Dataset1* and *Dataset2*, respectively.

In this approach, the recognition of the pronounced word and the identification of the speaker are performed independently. In the closed set application, this is done as follows:

$$\widehat{W} = \underset{\{W\}}{\operatorname{argmax}} \; [\log P(W|\Theta_1, X)] \tag{6.5}$$

and

$$\widehat{S}_c = \underset{\{\Lambda_c\}}{\operatorname{argmax}} \; [\log p(X|\Lambda_c)] \tag{6.6}$$

In open set application, the speaker is accepted if:

$$llr(X) = \frac{1}{T}\left[\log p(X|\Lambda_{\widehat{c}}) - \log p(X|\Lambda_1)\right] \geq \delta \tag{6.7}$$

where $llr(X)$ is the likelihood ratio, δ is a speaker and word independent threshold, $\Lambda_{\widehat{c}}$ is the speaker-dependent GMM model associated with the most likely speaker $\widehat{S}_c$ selected according to (6.6), Λ_1 is the GMM *world model* where its parameters are derived from Λ using MAP adaptation and the *world data* set and T is the utterance length after having removed the silence frames and is used to compensate the difference in utterance durations. The speech/silence segmentation is obtained by the HMM/MLP speech recognizer.

6.5.3 Sequential decision

To improve the speech recognizer performance, a speaker-dependent HMM/MLP system is used. In this approach, each speaker is modeled by two models, a SD-GMM Λ_c and a SD-HMM/MLP Θ_c. The parameters Θ_c of the SD-MLP are derived by re-training all the parameters Θ of the SI-MLP, using speaker's training data. The segmentation is obtained as described above in the baseline approach.

To perform speech and speaker recognition, the most likely speaker determined by the speaker identification subsystem using (6.6) is used to select the SD-HMM/MLP for speech recognition. The speaker in both closed set and open set applications is identified as in the baseline approach using (6.6) and (6.7), respectively.

The recognition of the pronounced word is performed as follows:

$$\widehat{W} = \underset{\{W\}}{\operatorname{argmax}} \left[\log P(W|\Theta_{\widehat{c}}, X)\right] \tag{6.8}$$

where $\Theta_{\widehat{c}}$ is the parameters of the SD-MLP associated with the most likely speaker $\widehat{S}_c$. With perfect recognition of the speaker identity (i.e., we know from whom the test utterance comes), the word recognition rate is equal to 98.9%.

The main advantage of this approach compared to the baseline is the gain we got in speech recognition performance. This gain should improve the performance of the joint speech and speaker recognition. A system using this approach can be viewed as performing "speaker quantization" or "speaker clustering" before speech recognition (Reynolds and Heck, 1991). That is the speaker identification system associates a new speaker to the most likely similar speaker in the registered speaker set.

6.5.4 Combined decision

As we have seen in the previous chapter, the posteriori probabilities estimated by the speaker dependent HMM/MLP are more effective for speech recognition than speaker recognition. Nevertheless, these posteriori probabilities can be used to improve the speaker recognition performance if they are combined with more speaker specific information.

In the closed set application, the recognition of both the pronounced word and the speaker identity is performed as follows:

$$(\widehat{W}, \widehat{S}_c) = \underset{\{W, \Lambda_c\}}{\operatorname{argmax}} \left[\log P(W|\Theta_c, X) + \log p(X|\Lambda_c)\right] \tag{6.9}$$

Given both speaker-dependent GMM and speaker-dependent HMM/MLP systems are trained with two different criteria, the posteriori probability and the likelihood scores estimated by each system might have some complementary (new) information that can be useful to improve the performance of each individual system or the joint speech and speaker recognition.

Criterion (6.9) should be applied for every registered speaker and every possible word in the vocabulary. Compared to the baseline and sequential approaches, we need an additional computational cost of $(N-1)$ times the cost of a speech recognition task, where N is the number of the registered speakers, making the computation requirements very costly. To reduce this cost, we first generate a list of the N-best candidates according to their likelihoods using the text-independent speaker identification (6.6). Then, for each speaker S_c in the N-best list, we use his/her associated SD-HMM/MLP Θ_c to perform a speech recognition step. The output of this step is the hypothesis word with its estimated posteriori probability score. Finally, we rescore the N-best list according to the combined likelihood and posterior probability scores using (6.9). This procedure generates a new N-best list, where the new identified speaker is selected according to the following criterion:

$$\widehat{S}_c = \underset{\{W, \Lambda_c\}}{\text{argmax}} \; [\log P(W|\Theta_c, X) + \log p(X|\Lambda_c)] \tag{6.10}$$

It is worth mentioning here that for a N-best list with $N = 1$, the sequential and combined approaches are equivalent.

For the open set application, the combined score of the identified speaker (6.10) is compared with the score estimated by the world model, in order to accept or reject the speaker. Hence, the speaker is accepted if:

$$[\log P(W|\Theta_{\widehat{c}}, X) + \log p(X|\Lambda_{\widehat{c}})] - \log p(X|\Lambda_1) \geq \delta \tag{6.11}$$

which is equivalent to:

$$\log P(W|\Theta_{\widehat{c}}, X) + llr(X) \geq \delta \tag{6.12}$$

where $\log P(W|\Theta_{\widehat{c}}, X)$ and $llr(X)$ are estimated according to (6.3) and (6.7). As we have done in the previous chapter (Section 5.6.1), a linear combination technique is used to combine posterior probabilities and likelihoods. The combined score in (6.9) and (6.12) are, then estimated, respectively, as follows:

$$(\widehat{W}, \widehat{S}_c) = \underset{\{W, \Lambda_c\}}{\text{argmax}} \; [\alpha_1 \log P(W|\Theta_c, X) + \log p(X|\Lambda_c)] \tag{6.13}$$

and the speaker is accepted if:

$$\alpha_2 \log P(W|\Theta_{\widehat{c}}, X) + llr(X) \geq \delta \tag{6.14}$$

where α_1 and α_2 weight the contribution of the posteriori probability for close and open set applications, respectively.

The criteria (6.6) and (6.9) will be used to determine the best speaker model (cluster) to perform a speaker-independent speech recognition.

6.6 Experiments and results

The aim of these experiments is to evaluate, analyze and compare the effectiveness of the three different approaches described above in three different tasks, closed set speaker identification, open set speaker identification and speaker clustering for speech recognition.

In closed and open set speaker identification experiments, our interest is to improve the joint speaker and speech recognition rate. In the results, this will be referred to as *joint recognition rate*. In speaker clustering experiments, our aim is to perform and improve a speaker-independent speech recognition performance using only the set of registered speakers.

6.6.1 Closed set results

In closed set application the unknown (test) speaker should be one of the registered speakers. Table 6.1 reports the performance of each approach in terms of speech, speaker and joint recognition rate for an optimized α_1.

For clarity sake, the results using only hybrid speaker-dependent HMM/MLP are also reported. This approach is referred to as *connectionist* and more analysis can be found in (BenZeghiba and Bourlard, 2003b). In this approach the *joint recognition* rate is estimated as follows:

$$(\widehat{W}, \widehat{S}_c) = \underset{\{W, \Theta_c\}}{\text{argmax}} \; [\log P(W|\Theta_c, X)] \tag{6.15}$$

It is worth mentioning here, that the best *joint recognition* rate we can achieve will be equal to the lowest recognition rate given by speech or speaker systems.

APPROACHES	BASELINE	SEQUENTIAL	COMBINED (N = 2, $\alpha_1 = 0.5$)	CONNECTIONIST
SPEECH RECO.	97.2%	98.7%	98.7%	98.0%
SPEAKER RECO.	95.9%	95.9%	**96.8%**	67.0%
JOINT RECO.	93.4%	95.1%	**95.9%**	66.8%

Table 6.1. *Speech, speaker and joint recognition rates using different approaches with optimized* α_1

From these results, we can see that:

1. The sequential approach gives better performance in terms of *joint recognition* rate than the baseline approach. This is due mainly to the improvement in the speech recognition rate. From the computational cost point of view, both approaches have the same cost. It is interesting to note here that the speech recognition rate in the sequential approach (98.7%) is almost equal to that obtained with perfect speaker identification (98.9%). This means, that the hybrid HMM/MLP model Θ_c of the speaker S_c still recognizes correctly the pronounced word even if the speech segment comes from another mis-identified speaker.

2. Compared to the two other approaches, the combination of the speech recognition score and speaker recognition score improves the speaker recognition rate. As a consequence, the *joint recognition* performance is also improved.

   The size of the N-best list should be chosen as a trade-off between the *joint recognition* rate and the computational cost. Looking at the speaker recognition performance obtained by the connectionist approach (67.0%) and the speaker identification system (95.9%), we can deduce that using the combined criterion (6.10) for speaker recognition, the likelihood estimated by the SD-GMM Λ_c will have more contribution than the posteriori probability. Hence, if the correct speaker does not appear in the first few positions in the N-best list according to the maximum likelihood criterion, the use of (6.9) is not going to help much.

   In Figure 6.1, the variations of speech, speaker and *joint recognition* rates as a function of the size of the N-best list are plotted. The recognition rates corresponding to $N = 1$ are those obtained by the sequential approach. The figure shows that the most significant improvement is obtained by selecting the first two best likely speakers according to (6.6), and then use (6.9) for re-scoring. From the computational cost point of view, the combined approach needs only one more speech recognition step which depends on the size of the MLP and the length of the test utterance.

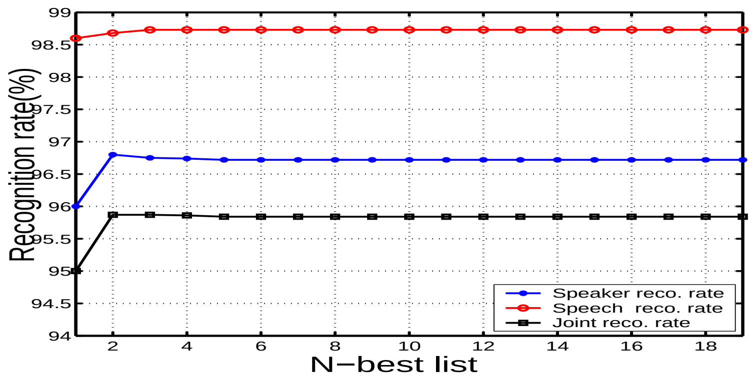

Figure 6.1. *Speaker, speech and joint recognition rates as a function of the size of the N-best list*

6.6.2 Open set results

To evaluate the combined approach in a more practical application, open set experiments are conducted. This is similar to a one step speaker verification. The goal is to detect an unknown speaker (impostor) and reject him. Three types of errors are considered here (Hazen *et al.*, 2003), false acceptance (FA), false rejection (FR) and Confusion Acceptance (CA), that is, when a registered speaker is accepted but confused with another speaker. Based on the closed set experimental results, we can expect that the combined criterion should reduce the confusion acceptance rate.

The variations of these errors as a function of a threshold for the sequential [2] and combined approaches are plotted in Figure 6.2 and Figure 6.3, respectively.

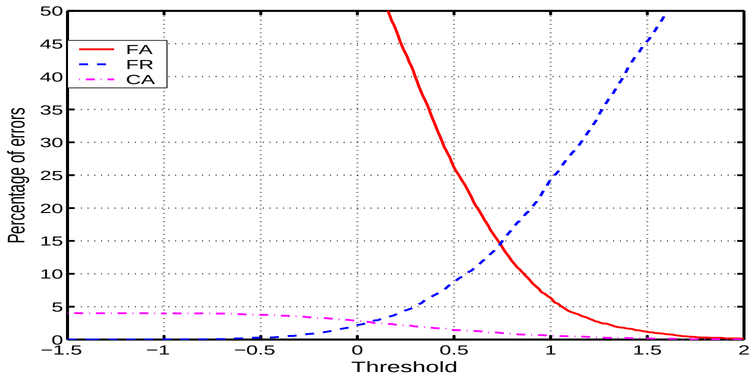

Figure 6.2. *False acceptance (FA), false rejection (FR) and confusion acceptance (CA) variations as a function of the threshold for baseline and sequential approaches*

The EERs (FA = FR) obtained by the sequential and combined approaches were equal to 14.5% and 13.1%, respectively. Moreover, and as we have expected, the combined criterion reduces the confusion acceptance errors. It is not surprising that the EER in both approaches is quite high. In fact, in an open set application and unlike speaker verification, no claimed identity is provided to the system. The test speaker will be given the identity of a registered speaker whose model matches best (i.e., the maximum score) the test utterance, which increases the chance of acceptance for that speaker (i.e., a high false acceptance).

[2] Both baseline and sequential approaches use the same speaker identification criterion (6.7) in an open set application.

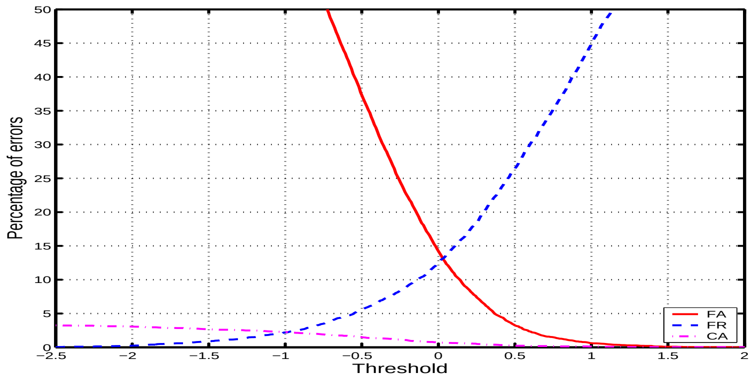

Figure 6.3. *False acceptance, false rejection and confusion acceptance variations as a function of the threshold for the combined approach*

If we consider only the registered speakers that have been accepted, the *joint recognition* rates with the baseline, sequential and combined approaches is equal to 82%, 83.8% and 85.7%, respectively, confirming the tendency we have seen in closed set application.

6.6.3 ASR based speaker clustering

In this experiment, we evaluate the use of the sequential and combined approaches to perform a speaker-independent speech recognition in open set application. This is done by selecting the registered speaker that is acoustically close to the test speaker and using the speech recognizer associated with the selected speaker to perform speech recognition. The main issue here is the choose of a proper criterion that will be used to select the closest registered speaker. We have tested the sequential and combined criteria given, respectively, by (6.6) and (6.10). Results of the speech recognition performance are shown in Table 6.2. For comparison purposes, the average performance of using each enrolled speaker is also reported ("Single speaker" column). To make the task harder, we have used only impostor utterances (*Dataset2* with 6452 utterances).

Approaches	Single speaker	Sequential	Combined
Speech Reco.	85.2%	92.3%	**93.5%**

Table 6.2. *Speech recognition on Dataset2 (lexical size* = 17, 6452 *test utterances)*

As we can see, the use of the sequential criterion gives 8.4% relative improvement compared to the single speaker results. The best improvement is achieved by the combined criterion (11% relative improvement). This results indicate that selecting a reference speaker (cluster) that is close to the test speaker in the acoustic reference speaker (cluster) space is not optimal. Using (6.10) criterion, the selected reference speaker is acoustically close to the test speaker in the joint speech and speaker space.

6.7 Conclusion

This chapter has presented a probabilistic approach that maximizes the joint posterior probability of speech and speaker. The joint posterior probability is then expressed as a combination of posteriori probability estimated by a hybrid HMM/MLP system and likelihood estimated by a text-independent GMM. In this approach, the score of the recognized word (i.e., posteriori probability) is used to enhance the performance of a likelihood based speaker identification. This approach was evaluated and compared with two other conventional approaches in three different applications: closed set speaker identification, open set speaker identification and speaker clustering with the goal to perform speaker-independent speech recognition. In all these applications, results showed the effectiveness of the proposed approach.

Chapter 7

Conclusion

The goal of this thesis was to investigate and optimize different approaches towards User-Customized Password Speaker Verification (UCP-SV) system. Speaker verification has already been widely investigated, as summarized in Chapters 3 and 4. However, although more user-friendly, UCP-SV systems are less understood and actually exhibits several new challenges, including: automatic inference of HMM password models (from a speaker-independent ASR system), fast speaker adaptation of the resulting acoustic models, score normalization, and verification. Evaluation of such systems is then based on their ability to simultaneously verify two hypotheses: (1) the identity of a claimed speaker and (2) the validity of the password. Beyond HMM inference and speaker adaptation, it was shown in this thesis that using an appropriate score normalization was critical in getting good performance, and several solutions had to be investigated to improve the baseline system and to make our UCP-SV system actually practical.

Since UCP-SV involves both speech recognition and speaker verification, a natural extension of our work was also to investigate new approach towards jointly using ASR together with speaker recognition (SR) to simultaneously improve both ASR and SR systems. In this thesis, we have shown that optimization and recognition based on a joint ASR-SR posterior probability was yielding better ASR and SR performance, beyond what could be achieved through a "sequential" approach (e.g., first performing speaker ID/clustering, followed by speech recognition).

7.1 User-customized password speaker verification

In UCP-SV systems, passwords are modeled by phoneme-based HMMs whose phonetic transcriptions are automatically inferred using a speaker-independent speech recognizer and whose states emission probabilities were estimated using either an adapted GMM or an adapted MLP. Two speaker acoustic modeling approaches were investigated, the HMM/GMM approach and the HMM/MLP approach.

7.1.1 HMM/GMM approach

In the HMM/GMM approach, the goal was to study and evaluate the effect of the accuracy of the inferred phonetic transcriptions on the performance of the UCP-SV system. This was initially done by developing a baseline UCP-SV system that used the best inferred phonetic transcription to create the customer specific HMM password. To evaluate the quality of the inferred model, the performance of this baseline system was compared with a TD-SV system as reported in Chapter

4. Possible ways to improve the performance of the baseline system using all the inferred phonetic transcriptions were then investigated, yielding the following conclusions:

1. With a reasonable speaker-independent HMM/MLP speech recognizer, a UCP-SV system with satisfactory performance can be developed. A possible reason is that the errors made by the speech recognizer at the HMM inference level can be compensated by the speaker adaptation.

2. The HMM inference is more important for background acoustic modeling than customer acoustic modeling. Indeed, the background model should be as general as possible to cover how impostors pronounce the password. In UCP-SV, the background model should ideally be acoustically speaker-independent but lexically customer dependent.

3. The different inferred phonetic transcriptions can be considered as a source of high level information about how customers pronounce their passwords. This information can be exploited during the verification to reduce the mismatch due to the variability in password pronunciations.

4. During speaker verification hypothesis testing, it is known that better performance can be achieved if the background model used to estimate the denominator of the likelihood ratio (i.e., likelihood of the background model) is as competitive as possible to the customer model. In this thesis, it was shown that the (lexical) competitiveness of the background model to the customer model can be greatly improved using multiple background models, each of them is associated with one of the inferred phonetic transcriptions.

5. If for each enrollment utterance (i.e., for each inferred HMM), we create a UCP-SV subsystem, then dynamically selecting (during verification) the subsystem with the minimum *log likelihood ratio* always yields the best performance. This performance was comparable to that obtained when the average of *log likelihood ratios* estimated over all subsystems is used or by fusing the local decision made by each subsystem.

7.1.2 HMM/MLP approach

Our motivation at using HMM/MLP systems for UCP-SV was their good performance at least to the HMM/GMM systems usually reported on speech recognition tasks. While still using HMM/MLP for HMM inference, the approach investigated here was to use an adapted MLP instead of the adapted HMM/GMM to estimate state posterior probabilities of the inferred HMM model. These probabilities are then used in Viterbi decoding to obtain the verification score as reported in Chapter 5. MLP adaptation and score normalization were the two main investigation issues, resulting in the following conclusions:

1. While the use of double normalization as a confidence measure for speech recognition task shown good performance, its use for a speaker verification task found to be less effective and degraded the performance. This confidence measure (i.e., double normalization) appears to be a good estimate to evaluate how good a word HMM model matches the data. Compared to the use of standard posterior probability confidence measure, the double normalization confidence measure is more sensitive to the poorly matched phones, increasing the false rejection rate due to the variations in pronouncing the password during the test.

2. While the use of score normalization techniques (i.e., *log likelihood ratio*) in HMM/GMM was found to be effective in speaker verification system, their use in hybrid HMM/MLP did not bring any improvement and, in some cases, degraded the performance of the UCP-SV system (such as constrained log posterior probability ratio given by (5.8)).

3. Reducing the number of the adapted parameters with respect to the amount of the adaptation data is effective as far as the modeling capacity and the generalization capabilities of the adapted MLP are not affected. In our task (UCP-SV system) and given the particularity of the data, we have found that the use of an SLP without acoustic context in the input is a reasonable choice.

4. Adding some speaker-independent examples (acoustic frames) corresponding to the *unseen* phonemes in the original adaptation data (i.e., the enrollment data provided by the customer) reduced the false acceptance rate. This reduction was found to be significant in case of impostor with opposite gender than the customer.

5. Embedding adaptation using the adapted customer SLP to generate a new segmentation was found to yield some further improvement.

6. Compared to the HMM/GMM based UCP-SV systems, the use of the hybrid HMM/MLP systems for UCP-SV as usually used for speech recognition, is not effective. The MLP is adapted to discriminate between phone classes and not between a specific customer and impostors. In fact, the adaptation procedure significantly improved the frame recognition rate of customer utterances without affecting strongly the posteriori probabilities of impostors' utterances, making the use of posteriori probabilities for speaker verification scoring not effective.

7. These posteriori probabilities can be combined with likelihood ratio estimated by a conventional GMM-UBM based text-independent speaker verification system to perform a UCP-SV task. In this case, the posteriori probabilities are used for utterance verification while likelihood ratios are used for speaker verification.

8. When a GMM model is trained with data associated with a short password, the GMM becomes text and speaker dependent, although the temporal phonetic structure is not preserved.

7.2 Combined speech and speaker recognition

The goal was to investigate a new probabilistic framework that maximizes the joint posteriori probability of the pronounced word (ASR) and the speaker identity (SR) given the observed data. This probability is expressed as a product of posteriori probability estimated by a hybrid HMM/MLP system (ASR) and a likelihood estimated by a GMM model (SR). This framework was examined and tested in three applications: closed set speaker identification, open set speaker identification, and speaker clustering. The main conclusions can be summarized as follows:

1. The speech recognition score can be used to enhance the speaker recognition performance, so, the joint speech and speaker recognition rate get improved. It has been shown that this framework is superior to a system performing SR and ASR sequentially (i.e., speaker identification followed by speech recognition).

2. This framework can be used as a good criterion to select the closest cluster (speaker clustering) to perform speaker-independent speech recognition.

7.3 Future directions

- *Threshold setting:* Evaluating the optimal decision threshold is a very important and difficult issue in speaker verification, and even more so in UCP-SV systems. Indeed, to set a predictable *priori* threshold, a pseudo-impostor data that is close (contains the same content) to the customer enrollment data should be available. In UCP-SV systems, finding such utterances with such constraint is quite impossible.

To alleviate this problem, a possible solution is to use a phone-based speaker verification score (or decision). That is the *log likelihood ratio* score will be estimated from the phone-based *log likelihood ratio* scores rather than from utterance level. To determine the beginning and the end of each phoneme, a *synchronous alignment* technique can be used (Mariéthoz *et al.*, 2000), where the Viterbi path is forced to be the same for both speaker and background model. The use of phone-based speaker verification score has many advantages:

1. For each phone q_k in the customer HMM password model, examples from pseudo impostor data of the same phone q_k can be used to estimate the phone-based log likelihood ratio. These phone-based log likelihood ratio can be then summed over all possible phones in the customer HMM password model to estimate the speaker verification score. With this technique, the customer and impostor score distributions can be estimated on a development set and *a priori* threshold can be determined.
2. The phone-based log likelihood ratio can be used to estimate the customer and impostor score distributions for each phone q_k to determine a phone-dependent threshold. In this case, a decision fusion technique can be used to derive the final decision to accept or reject a speaker.
3. In both cases, the phone-based log likelihood ratio or the phone-dependent threshold can be weighted according to the phone discriminant capabilities. It is well known that some phonemes carry more speaker specific information than others, hence, weighting phonemes according to their potential speaker discrimination could improve the speaker verification performance (Auckenthaler *et al.*, 1999; Chan and Siu, 2004).

- *Language-independent UCP-SV:* The UCP-SV systems studied in this thesis are language-dependent (French for instance). A more user friendly UCP-SV system could be language-independent, where a customer can choose a password in any language.

  A possible technique for speaker acoustic modeling is to segment the speech signal into subword units without using a speaker-independent speech recognizer (Sharma and Mammone, 1996). Discriminative or generative models can be then created for each subword units. The difficulty with such techniques though is that the impostor data should be properly segmented to easily search for subword units that are close to the speaker subword units determined by the segmentation procedure.

- *Online adaptation:* In this thesis, we have shown the importance of adaptation techniques when the amount of the available enrolment data is not large enough to create a new speaker model from scratch. There are two modes of adaptation, supervised and unsupervised modes. In the former mode, the target model to be adapted is known, while in the later mode, the system has to determine which model the data belongs to. In the unsupervised adaptation mode, the performance of the adapted model depends on how accurate the *a priori* distribution (initial model) determined by the system is. To create a speaker-dependent speech recognizer for a male speaker, starting from a male-dependent speech recognizer is a good choice. In such applications, the general approach is that, the system, first determine the closest cluster (which can be language, gender,...etc) to the new speaker (cluster identification) and then used the corresponding speech recognizer for adaptation. As we have shown, for such applications, the use of the combined criterion will improve the cluster identification performance, equivalently, the accuracy of the adapted model.

Appendix A

List of passwords

	WORD	PHONETIC TRANSCRIPTION
EXPECTED PASSWORDS	annulation	/aa/nn/uu/ll/aa/ss/yy/on/
	casino	/kk/aa/zz/ii/nn/au/
	cinema	/ss/ii/nn/ai/mm/aa/
	concert	/kk/on/ss/ai/rr/
	exposition	/ai/kk/ss/pp/au/zz/ii/ss/yy/on/
	galerie du manoir	/gg/aa/ll/ee/rr/ii/dd/uu/mm/aa/nn/ww/aa/rr/
	gianadda	/jj/yy/aa/nn/aa/dd/aa/
	louis moret	/ll/ww/ii/mm/oo/rr/ei/
	message	/mm/ai/ss/aa/jj/
	mode d'emploi	/mm/oo/dd/an/pp/ll/ww/aa/
	musee	/mm/uu/zz/ei/
	precedent	/pp/rr/ai/ss/ai/dd/an/
	quitter	/kk/ii/tt/ei/
	suivant	/ss/uy/ii/vv/an/
INVALID PASSWORDS	manifestation	/mm/aa/nn/ii/ff/ai/ss/tt/aa/ss/yy/on/
	corso	/kk/oo/rr/ss/au/
	guide	/gg/ii/dd/

Appendix B

Acronyms

ANN	Artificial neural network
ASR	Automatic speech recognition
CLPPR	Constrained log posterior probability ratio
CM	Confidence measure
DET	Detection Error trade-off
DN	Double normalization
DNCM	Double normalization confidence measure
EER	Equal error rate
EM	Expectation-maximization algorithm
FA	False acceptance
FAR	False acceptance rate
FR	False rejection
FRR	False rejection rate
GMM	Gaussian mixture models
HMM	Hidden Markov model
HMM/GMM	Hidden Markov model/Gaussian mixture model
LIN	Linear input network
LLR	Log likelihood ratio
LPCC	Linear prediction cepstral coefficients
LPP	Log posterior probability
MAP	Maximum *a posteriori*
MFCC	Mel-frequency cepstral coefficients
MLP	Multi-layer perceptron
NLLR	Normalized log likelihood ratio
PT	Phonetic transcription
RPCM	Relative posterior confidence measure
ROC	Receiver operating characteristic
SID	Speaker identification

SD-HMM	Speaker-dependent hidden Markov model
SI-HMM	Speaker-independent hidden Markov model
SD-MLP	Speaker-dependent multi-layer perceptron
SD-SLP	Speaker-dependent single-layer perceptron
SI-MLP	Speaker-independent multi-layer perceptron
SI-SLP	Speaker-independent single-layer perceptron
SLP	Single layer perceptron
SPCM	Standard posterior confidence measure
SR	Speaker recognition
SV	Speaker verification
SVS	Speaker verification score
TD-SV	Text-dependent speaker verification
TI-SV	Text-independent speaker verification
UCP-SV	User-customized password speaker verification
ULPPR	Unconstrained log posterior probability ratio
UV	Utterance verification
UVS	Utterance verification score
VS	Verification score

Appendix C

Notation

S	speaker
S_c	customer
$\overline{S}_c$	impostor
x_t	acoustic vector at time t
$x^{\ell}_{(t,c)}$	acoustic vector at time t in the ℓ^{th} repetition of the customer's password
$X = \{x_1, ..., x_t, ..., x_T\}$	acoustic vector sequence of length T
$X^{\ell}_c = \{x^{\ell}_{(1,c)}, ..., x^{\ell}_{(t,c)}, ..., x^{\ell}_{(T,c)}\}$	acoustic vector sequence of the ℓ^{th} repetition of the customer password
M	ergodic HMM/MLP model
M^{ℓ}_c	the phonetic transcription associated with the ℓ^{th} repetition of the customer's password S_c
λ	speaker-independent ergodic HMM/GMM model
Λ	speaker-independent GMM model
Λ_c	speaker-dependent GMM model
Θ	speaker-independent MLP
Θ_c	speaker-dependent MLP
θ	speaker-independent SLP
θ_c	speaker-dependent SLP
(M^{ℓ}_c, λ)	speaker-independent left-to-right password HMM/GMM model associated with M^{ℓ}_c
(M^{ℓ}_c, λ_c)	speaker-dependent left-to-right password HMM/GMM model associated with M^{ℓ}_c
$(\widehat{M}^{\ell}_c, \lambda_c)$	speaker-dependent left-to-right password HMM/GMM model associated with the best phonetic transcription M^{ℓ}_c
(M^{ℓ}_c, Θ)	speaker-independent left-to-right password HMM/MLP model associated with M^{ℓ}_c
(M^{ℓ}_c, Θ_c)	speaker-dependent left-to-right password HMM/MLP model associated with M^{ℓ}_c

q_k^t	phone q_k (associated with HMM state) observed at time t
$q_k^{(t,\ell)}$	phone q_k decoded at time t using the model M_c^ℓ
q_{best}^t	best phone observed at time t
$P(.)$	posterior probability
$p(.)$	likelihood
$P(q_k)$	prior probability of phone q_k
$P(q_k^t\|x_t,\Theta)$	local posterior probability of phone q_k at time t estimated through speaker-independent MLP Θ
$P(q_k^{(t,\ell)}\|x_t,\Theta)$	local posterior probability of the decoded phone q_k at time t using the model M_c^ℓ and estimated through a speaker-independent MLP Θ
$P(q_k^{(t,\ell)}\|x_{(t,c)}^i,\Theta)$	local posterior probability of the decoded phone q_k at time t using the model M_c^ℓ given the t^{th} fame of the i^{th} customer enrollment utterance
$P(q_k^{(t,\ell)}\|x_t,\Theta_c)$	local posterior probability of the decoded phone q_k at time t using the model M_c^ℓ and estimated through a speaker-dependent MLP Θ_c
$P(M,S\|X)$	joint posterior probability of the password M and the speaker S given X
$P(M_c^\ell\|X,\Theta)$	posterior probability of the Markov model M_c^ℓ given the utterance X
$p(X\|M_c^\ell,\lambda)$	likelihood of X given the Markov model (M_c^ℓ,λ)
$p(X\|\Lambda)$	likelihood of X given the GMM model Λ
$llr(X)$	log likelihood ratio given the utterance X
LLR_s	speaker verification log likelihood ratio
LLR_u	utterance verification log likelihood ratio
$LLR_s^{M_c^\ell}$	speaker verification log likelihood ratio using the phonetic transcription M_c^ℓ
$LLR_u^{M_c^\ell}$	utterance verification log likelihood ratio using the phonetic transcription M_c^ℓ

Bibliography

Abdulla, W. H. and Kasabov, N. K. (2001). "Improving Speech Recognition Performance through Gender Separation". In *Artificial Neural Networks and Expert Systems Inter. Conf. (ANNES)*, pages 218–222.

Abrash, V. (1997). "Mixture Input Transformations for Adaptation of Hybrid Connectionist Speech Recognizers". In *European Conference on Speech Communication and Technology, Eurospeech'97*, volume 1, pages 299–302.

Abrash, V., Franco, H., Sankar, A., and Cohen, M. (1995). "Connectionist Speaker Normalization and Adaptation". In *European Conference on Speech Communication and Technology, Eurospeech'95*, volume 2, pages 165–168.

Ahn, S., Kang, S., and Ko, H. (2000). "Effective Speaker Adaptation for Speaker Verification". In *Inter. Conf. on Speech and Signal Processing, ICASSP'00*, volume 1, pages 1081–1084.

Ariyaeeinia, A. M. and Sivakuaran, P. (1997). "Analysis and Comparison of Score Normalization Methods For Text-Dependent Speaker Verification". In *European Conference on Speech Communication and Technology, Eurospeech'97*, pages 1379–1382, Rhodes, Greece.

Auckenthaler, R., Parris, E. S., and Carey, M. J. (1999). "Improving a GMM Speaker Verification System by Phonetic Weighting". In *Inter. Conf. on Speech and Signal Processing, ICASSP'99*, volume 1, pages 313–316, Phoenix.

Auckenthaler, R., Carey, M., and Lioyd-Thomas, H. (2000). "Score Normalization for Text-Independent Speaker Verification Systems". *Digital Signal Processing*, **10**(1-3), 42–54.

Bahl, L. R., Brown, P. F., de Souza, P. V., and Mercer, R. L. (1986). "Maximum Mutual Information of Hidden Markov Model Parameters". In *Inter. Conf. on Speech and Signal Processing, ICASSP'86*, volume 1, pages 49–52.

Bakis, R. (1976). "Continuous Speech Word Recognition via Centisecond Acoustic states". In *Proc. of 91st Annual Meeting of the Acous. Society of America*.

Bengio, S. and Mariéthoz, J. (2004). "A Statistical Significant Test for Person Authetication". In *The Speaker and Language Recognition Workshop, Odyssey*, volume 1, pages 237–244.

Bennani, Y. and Gallinari, P. (1995). "Neural Networks for Discrimination and Modelization of Speakers". *Speech Communication*, **17**, 159–175.

BenZeghiba, M. and Bourlard, H. (2004a). "User-Customized Password Speaker Verification Using Multiple Reference and Background Models". IDIAP-RR 41, IDIAP.

BenZeghiba, M., Bourlard, H., and Mariéthoz, J. (2001). "Speaker Verification based on User-Customized Password". IDIAP-RR 13, IDIAP.

BenZeghiba, M. F. and Bourlard, H. (2002). "User-Customized Password Speaker Verification based on HMM/ANN and GMM Models". In *Inter. Conf. on Spoken Language Processing, ICSLP'02*, volume 2, pages 1325–1328, Denver, USA.

BenZeghiba, M. F. and Bourlard, H. (2003a). "Hybrid HMM/ANN and GMM Combination for User-Customized Password Speaker Verification". In *Inter. Conf. on Speech and Signal Processing, ICASSP'03*, volume 1.

BenZeghiba, M. F. and Bourlard, H. (2003b). "On the Combination of Speech and Speaker Recognition". In *European Conference on Speech Communication and Technology, Eurospeech'03*, pages 1361–1364, Geneva.

BenZeghiba, M. F. and Bourlard, H. (2004b). "Confidence Measures in Multiple Pronounciations Modeling for Speaker Verification". In *Inter. Conf. on Speech and Signal Processing, ICASSP'04*, volume 1, pages 389–392, Montreal.

BenZeghiba, M. F. and Bourlard, H. (2004c). "Posteriori Probabilities and Likelihoods Combination for Seech and Seaker Rcognition". In *Inter. Conf. on Spoken Language Processing, ICSLP'04*, volume 1.

Bernardis, G. and Bourlard, H. (1998). "Improving Posterior Based Confidence Measures in Hybrid HMM/ANN Speech Recognition Systems". In *Inter. Conf. on Spoken Language Processing, ICSLP'98*, volume 3, pages 775–779.

Bimbot, F., Bonastre, J.-F., Fredouille, C., Gravier, G., Magrin-Chagnolleau, I., Meignier, S., Merlin, T., Ortega-Garcia, J., Petrovska-Delacrétaz, D., and Reynolds, D. A. (2004). "A Tutorial on Text-Independent Speaker Verification". *EURASIP Journal on Applied Signal Processing*, **4**, 430–451.

Bourlard, H. and Morgan, N. (1994). *"Connectionist Speech Recognition: A hybrid approach"*. Kluwer Academic Publisher.

Bourlard, H., Kamp, Y., Ney, H., and Wellekens, C. J. (1985). "Speaker-Dependent Connected Speech Recognition via Dynamic Programing and Statistical Methods". In *Speech and Speaker Recognition*, volume 12, pages 115–148. Karger, Basel.

Bridle, J. S. (1990). "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition". In *Neurocomputing, Algorithms, Architectures and Applications*, pages 227–236. F. Fogelman Soulié and J. Hérault.

Burnett, D. C. (1997). *"Rapid Speaker Adaptation for Neural Network Speech Recognizers"*. Ph.D. thesis, Oregon Graduate Institute of Science.

Campbell, J. P. (1997). "Speaker Recognition: A Tutorial". *"Proceedings of IEEE"*, **85**(9), 1437–1462.

Carey, M., Parris, E., and Bridle, J. (1991). "A Speaker Verification System using alpha-nets". In *Inter. Conf. on Speech and Signal Processing, ICASSP'91*, volume 1, pages 397–400, Toronto, Canada.

Chan, S.-M. and Siu, M.-H. (2004). "Discrimination Power Weighted Subword-Based Speaker Verification". In *Inter. Conf. on Speech and Signal Processing, ICASSP'04*, volume 1, pages 45–48, Montreal.

Che, C., Lin, Q., and Yuk, D.-S. (96). "An HMM to Text-Prompted Speaker Verification". In *Inter. Conf. on Speech and Signal Processing, ICASSP'96*, volume 1, pages 673–676, Atlanta, Georgia, USA.

Chen, K. (2003). "Towards better Making a Decision in Speaker Verification". *Pattern Recognition*, **36**, 329–246.

Chollet, G., Cochard, J.-L., Constantinescu, A., Jaboulet, C., and Langlais, P. (1996). "Swiss French Polyphone and Polyvar: Telephone speech databases to model inter- and intra-speaker variability". IDIAP-RR 01, IDIAP.

Chou, W., Juang, B.-H., Liu, C.-S., Lee, C.-H., and Rosenberg, A. E. (1995). "A Study on Minimum Error Discriminative Training for Speaker Recognition". *JASA*, **97**(1), 637–648.

Collobert, R., Bengio, S., and Mariéthoz, J. (2002). " Torch: A Modular Machine Learning Software Library". IDIAP-RR 46, IDIAP.

Davis, S. B. and Mermelstein, P. (1980). "Comparison of Parametric Representation for Monosyllabic Word Recognition in Continously Spoken Sentences". *IEEE Trans. on Acoust. Speech and Signal Processing*, **28**(4), 357–366.

Dempster, A. P., Laird, N. M., and Rubain, D. D. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". *Journal of the Royal Statistical Society*, **39**(1), 1–38.

DeVeth, J. and Bourlard, H. (1995). "Comparison of Hidden Markov Model Techniques for Automatic Speaker Verification in real-world Conditions". *Speech Communication*, **17**, 81–90.

Doddington, G. R. (1985). "Speaker Recognition: Identifying people by their voices". *Proceedings of the IEEE*, **73**, 1651–1664.

Fontain, V. and Bourlard, H. (1997). "Speaker-Dependent Speech Recognition based on Phone-Like Unit Model– Application to Voice Dialing". In *Inter. Conf. on Speech and Signal Processing, ICASSP'97*, pages 1527–1530, Munich, Germany.

Fukunaga, K. (1990). *"Statistical Pattern Recognition"*. Academic Press.

Furui, S. (1981). "Cepstral Analysis Technique for Automatic Speaker Verification". *IEEE Trans. on Acoustics Speech and Signal Processing*, **ASSP- 29**(2), 251–272.

Furui, S. (1994). "An Overview of Speaker Recognition Technology". In *ESCA workshop on Automatic speaker Recognition, Identification and Verification*, pages 1–9.

Gagnon, L., Stubley, P., and Mailhot, G. (2001). "Password-Dependent Speaker Verification Using Quantized Acoustic Trajectories". In *Inter. Conf. on Speech and Signal Processing, ICASSP'01*, volume 1, pages 449–452, Salt Lake City, Utah.

Gales, M. (1996). "The Generation and Use of Regression Class Trees for MLLR Adaptation". TR 263, Combridge University Engineering Departement.

Gauvain, J. L. and Lee, C.-H. (1994). "Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observation of Markov Chains". *IEEE Trans. on Speech and Audio Processing*, **2**, 291–298.

Genoud, D., Ellis, D., and Morgan, N. (1999a). "Combined Speech and Speaker Recognition with Speaker-Adapted Connectionist Models". In *Proc. Automatic Speech Recognition and Understanding Workshop*, Keystone,CO.

Genoud, D., Ellis, D., and Morgan, N. (1999b). "Simultaneous Speech and Speaker Recognition using Hybrid Architecture". TR 99-012, Inter. Computer Science Institute (ICSI).

Gish, H. and Schmidt, M. (1994). "Test-Independent Speaker Identification". *IEEE Signal Processing Magazine*, **October**(9), 18–32.

Gold, B. and Morgan, N. (2000). *"Speech and Audio Processing"*. Wiley.

Hattori, H. (1994). "Text-Independent Speaker Verification Using Neural Network". In *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 103–106.

Hazen, T. J. (1998). *'The Use of Speaker Correlation Information for Automatic Speech Recognition"*. Ph.D. thesis, Massachusetts Institute of Technology.

Hazen, T. J., Jones, D. A., Park, A., Kukolich, L. C., and Reynolds, D. A. (2003). "Integration of Speaker Recognition into Conversational Spoken Dialogue Systems". In *European Conference on Speech Communication and Technology, Eurospeech'03*, pages 1961–1964.

Hebert, M. and Peters, S. D. (2001). "Improved Normalization Without Recourse To An Impostor Database For Speaker Verification". In *European Conference on Speech Communication and Technology, Eurospeech'01*, pages 2557–2560.

Heck, L. P. (2002). "A Bayesian Framework for Optimizing the Joint Probability of Speaker and Speech Recognition Hyphothesis". In *Workshop on The Advent of Biometrics On the Internet, ROME, Italy*, pages 33–42.

Higgins, A., l. Bahler, and Porter, J. (1991). "Speaker Verification using Randomized Phrase Prompting". *Digital Signal Processing*, **1**, 89–106.

Huang, X., Acero, A., and Hon, H.-W. (2001). *"Spoken Language Processing"*. Prentice Hall.

Isob, T. and Takahashi, J.-I. (1999). "A New Cohort Normalization Using Local Acoustic Information For Speaker Verification". In *Inter. Conf. on Speech and Signal Processing, ICASSP'99*, pages 841–844.

Jain, N., Cole, R., and Barnard, E. (1996). "Creating Speaker-Specific Phonetic Templates with a Speaker-Independent Phonetic Recognizer: Implications for Voice Dialing". In *Inter. Conf. on Speech and Signal Processing, ICASSP'96*, volume 1, pages 881–884.

Jankowski, C., Quatieri, T., and Reynolds, D. (1995). "Measuring Fine Structure in Speech: Application to Speaker Identification". In *Inter. Conf. on Speech and Signal Processing, ICASSP'95*, volume 1, pages 325–328.

Jourlin, P., J. Luettin, D. G., and Wassner, H. (1997). "Acoustic Labial Speaker Verification". *Pattern Recognition letter*, **18**(9), 853–858.

Kajarekar, S., Ferrer, L., Venkataraman, A., Sonmez, K., Shriberg, E., Stolcke, A., Brattt, H., and Gadde, R. R. (2003). "Speaker Recognition using Prosodic and Lexical Features". In *IEEE Worshop on Automatic Speech Recognition and Understanding, ASRU'03*, volume 1, pages 19–24.

Kowalczyk, J. (2004). " Une Application de Reconnaissance du Locuteur: Le User-Customized Password Speaker Verification". IDIAP-COM 04-04, IDIAP.

Kung, S. Y., Mak, M. . W., and Lin, S. H. (2004). *"Biometric Authentication: A Machine Learning Approach"*. Prentice Hall PTR.

Leggetter, C. and Woodland, P. C. (1995). "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density HMM". *Computer Speeach and Language*, **9**, 171–185.

Li, Q., Juang, B.-H., Zhou, C.-H. L. Q., and Soong, F. (1999). "Recent Advancements in Automatic Speaker Authentication". *IEEE Robotics & Automation Magazine*, pages 24–34.

Li, Q., Juang, B., Zhou, Q., and Lee, C. H. (2000). "Automatic Verbal Information Verification for User Authentication". In *IEEE Trans. on Speech and Audio Processing*, volume 8, pages 585–596.

Maes, S. H. (1999). "Conversational Biometric". In *European Conference on Speech Communication and Technology, Eurospeech'99*, volume 3, pages 1219–1222.

Makhoul, J. (1973). "Spectral Analysis of Speech by Linear Prediction". *IEEE Trans. on Acoust. Speech and Signal Processing*, **21**(3), 140–148.

Mariéthoz, J. and Bengio, S. (2002). "A Comparative Study for Adaptation Methods for Speaker Verification". In *Inter. Conf. on Spoken Language Processing, ICSLP'02*, pages 581–584.

Mariéthoz, J., Linberg, J., and Bimbot, F. (2000). "A MAP Approach with Synchronous Decoding and Unit-based Normalization for Text-Dependent Speaker Verification". In *Inter. Conf. on Spoken Language Processing, ICSLP'00*.

Martin, A., Doddingtone, G., Kamm, T., Ordowski, M., and Przybock, M. (1997). "The DET Curve in Assessment of Detection Task Performance". In *European Conference on Speech Communication and Technology, Eurospeech'97*, volume 4, pages 1895–1898.

Mathan, L. and Miclet, L. (1990). "Speaker Hierarchical Clustering for Improving Speaker-Independent HMM Word Recognition". In *Inter. Conf. on Speech and Signal Processing, ICASSP'90*, volume 1, pages 149–152.

Matsui, T. and Furui, S. (1992a). "Comparison of Text-Independent Speaker Recognition Methods using VQ-Distorsion and Discrete/Continuous hmms". In *Inter. Conf. on Speech and Signal Processing, ICASSP'92*, volume 1, pages 157–160.

Matsui, T. and Furui, S. (1992b). "Speaker Recognition Using Concatenated Phoneme Models". In *Inter. Conf. on Spoken Language Processing, ICSLP'92*, volume 1, pages 603–606.

Matsui, T. and Furui, S. (1994). "Similarity Normalization Method for Speaker Verification based on a Posteriori Probability". In *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 59–62.

Mengusoglu, E. and Ris, C. (2001). "Use of Acoustic Prior Information for Confidence Measure in ASR Applications". In *European Conference on Speech Communication and Technology, Eurospeech'01*, pages 2557–2560.

Morgan, N. and Bourlard, H. (1995). "Neural Networks for Statistical Recognition of Continous Speech". *Proceedings of the IEEE*, **83**(5), 742–770.

Naik, J. M. and Lubensky, D. (1994). "A Hybrid HMM/ANN Speaker Verification Algorithm for Telephone Speech". In *Inter. Conf. on Speech and Signal Processing, ICASSP'94*, volume 1, pages 153–156.

Neto, J., Almeida, L., hochberg, M., Martins, C., Nunes, L., Renals, S., and Robinson, T. (1995). "Speaker-Adaptation for Hybrid HMM/ANN Continous Speech Recognition System". In *European Conference on Speech Communication and Technology, Eurospeech'95*, pages 2171–2174.

Pierrot, J. B., Lindberg, J., Koolwaaij, J., Hutter, H. P., Genoud, D., Blomberg, M., and Bimbot, F. (1998). "A Comparison of *a priori* Threshold Setting Procedures for Speaker Verification in the CAVE Project". In *Inter. Conf. on Speech and Signal Processing, ICASSP'98*, volume 1, pages 125–128.

Przybocki, M. A. and Martin, A. F. (1998). "NIST Speaker Recognitin Evaluation- 1997". In *Proc. RLA2C, Avignon*, pages 120–123.

Rabiner, L. (1989). "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". *Proceedings of IEEE*, **2**, 257–286.

Rabiner, L. R. and Juang, B. H. (1993). *"Fundamentals of Speech Recognition"*. Prentice Hall.

Renals, S., Morgan, N., Bourlard, H., Cohen, M., and Franco, H. (1994). "Connectionist Probability Estimators in HMM Speech Recognition". *IEEE Trans. on Speech and Audio Processing*, **2**.

Reynolds, D. A. (1995). "Speaker Identification and Verification using Gaussian Mixture Speaker Models". *Speech communication*, **17**, 91–108.

Reynolds, D. A. and Heck, L. P. (1991). "Integration of Speaker and Speech Recognition Systems". In *Inter. Conf. on Speech and Signal Processing, ICASSP'91*, volume 1, pages 869–872, Toronto, Canada.

Reynolds, D. A. and Rose, R. C. (1995). "Robust Text-Independent Speaker Identification using Gaussians Mixture Speaker Models". *IEEE Trans. on Speech and Audio Processing*, **3**(1), 72–83.

Reynolds, D. A., Quatieri, T. F., and Dunn, R. B. (2000). "Speaker Verification Using Adapted Gaussian Mixture Models". *Digital Signal Processing*, **10**(1-3), 19–41.

Richard, M. D. and Lippman, R. P. (1991). "Neural Network Classifiers Estimate Bayesian a Posteriori Probabilities". *Neural Computation*, **3**(4), 461–483.

Rivlin, Z., Cohen, M., Abrach, V., and Chung, T. (1996). "A Phone-Dependent Confidence Measure for Utterance Rejection". In *Inter. Conf. on Speech and Signal Processing, ICASSP'96*, pages 515–517.

Rodriguez-Linares, L., Garcia-Mateo, C., and Alba-Castro, J. L. (2003). "On Combining Classifiers for Speaker Authentication". *Pattern Recognition*, **36**(2), 347–359.

Rosenberg, A. E. and Parthasarathy, S. (1996). "Speaker Background Models for Connected Digit Password Speaker Verification". In *Inter. Conf. on Speech and Signal Processing, ICASSP'96*, pages 81–84.

Rosenberg, A. E. and Parthasarathy, S. (1997). "Speaker Identification with User-Selected Password Phrases". In *European Conference on Speech Communication and Technology, Eurospeech'97*, pages 1371–1374.

Rosenberg, A. E., DeLong, J., Lee, C.-H., Juang, B.-H., and k. Soong, F. (1992). "The Use of Cohort Normalized Scores for Speaker Verification". In *Inter. Conf. on Spoken Language Processing, ICSLP'92*, pages 599–602.

Rosenberg, A. E., Siohan, O., and Parthasarathy, S. (1998). "Speaker Verification Using Minimum Verification Error Training". In *Inter. Conf. on Speech and Signal Processing, ICASSP'98*, pages 105–108.

Rudasi, L. and Zahorian, S. A. (1990). "Text-Independent Talker Identification using Recurrent Neural Networks". *Journal of Acoustic Society of America (JASA)*, **87**, Supp. 1, S104.

Rudasi, L. and Zahorian, S. A. (1991). "Text-Independent Talker Identification with Neural Networks". In *Inter. Conf. on Speech and Signal Processing, ICASSP'91*, volume 1, pages 389–392.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). "Learning Internal Representations by Backprobagation Errors". In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 318–362. D.E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press.

Sarma, S. (1997). *"A Segment Based Speaker Verification Using SUMMIT"*. Ph.D. thesis, M.I.T Departement of Electrical Engineering and Computer Science.

Sharma, M. and Mammone, R. (1996). "Subword-Based Text-Dependent Speaker Verification System With User-Selectable Passwords". In *Inter. Conf. on Speech and Signal Processing, ICASSP'96*, volume 1, pages 93–96, Atlanta.

Siohan, O., Rosenberg, A. E., and Parthasarathy, S. (1998). "Speaker Identification Using Minimum Classification Error Training". In *Inter. Conf. on Speech and Signal Processing, ICASSP'98*, pages 109–112.

Siohan, O., Lee, C.-H., Surendran, A., and Li, Q. (1999). "Background Model Design for Flexible and Portable Speaker Verification Systems". In *Inter. Conf. on Speech and Signal Processing, ICASSP'99*, volume 1, pages 825–828.

Sonng, F. and Rosenberg, A. E. (1988). "On the Use of Instantaneous and Transitional Spectral Information in Speaker Recognition". *IEEE Trans. on Acoust. Speech and Signal Processing*, **36**(6), 871–879.

Surendran, A. C. and Lee, C. H. (2000). "A Priori Threshold Selection for Fixed Vocabulary Speaker Verification System". In *Inter. Conf. on Spoken Language Processing, ICSLP'00*.

Viterbi, A. (1967). "Error Bounds For Convolutional Codes and Asymptotically Optimum Decoding Algorithm". *IEEE trans. on Information Theory*, pages 260–269.

Wan, V. (2003). *"Speaker Verification using Support Vector Machines"*. Ph.D. thesis, Departement of Computer Science, University of Sheffield.

Williams, G. and Renals, S. (1997). "Confidence Measures for Hybrid HMM/ANN Speech Recognition". In *European Conference on Speech Communication and Technology, Eurospeech'97*, pages 1955–1958, Rhodes, Greece.

Williams, G. and Renals, S. (1998). "Confidence Measures for Evaluating Pronunciation Models". In *Modeling Pronunciation variation for Automatic speech Recognition*, pages 1955–1958, Rolduc.

Zhang, W. D., Yiu, K., Make, M. W., Li, C. K., and He, M. X. (1999). "A Priori Threshold Determination for Phrase-Prompted Speaker Verification. In *European Conference on Speech Communication and Technology, Eurospeech'99*, volume 2, pages 1023–1026, Budapest.