KTH Computer Science
and Communication

Automatic speaker verification on site and by telephone: methods, applications and assessment

HÅKAN MELIN

Doctoral Thesis
Stockholm, Sweden 2006

TRITA-CSC-A 2006:26
ISSN-1653-5723
ISRN KTH/CSC/A--06/26--SE
ISBN 91-7178-531-0, ISBN 978-91-7178-531-2

KTH Datavetenskap och kommunikation
Tal, musik och hörsel
SE-100 44 Stockholm
SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framläggges till offentlig granskning för avläggande av teknologie doktorsexamen i tal- och musikkommunikation tisdagen den 19 december 2006 klockan 14.00 i sal F3.

Tryck: Universitetsservice US AB

Abstract

Speaker verification is the biometric task of authenticating a claimed identity by means of analyzing a spoken sample of the claimant's voice. The present thesis deals with various topics related to automatic speaker verification (ASV) in the context of its commercial applications, characterized by co-operative users, user-friendly interfaces, and requirements for small amounts of enrollment and test data.

A text-dependent system based on hidden Markov models (HMM) was developed and used to conduct experiments, including a comparison between visual and aural strategies for prompting claimants for randomized digit strings. It was found that aural prompts lead to more errors in spoken responses and that visually prompted utterances performed marginally better in ASV, given that enrollment data were visually prompted. High-resolution flooring techniques were proposed for variance estimation in the HMMs, but results showed no improvement over the standard method of using target-independent variances copied from a background model. These experiments were performed on Gandalf, a Swedish speaker verification telephone corpus with 86 client speakers.

A complete on-site application (PER), a physical access control system securing a gate in a reverberant stairway, was implemented based on a combination of the HMM and a Gaussian mixture model based system. Users were authenticated by saying their proper name and a visually prompted, random sequence of digits after having enrolled by speaking ten utterances of the same type. An evaluation was conducted with 54 out of 56 clients who succeeded to enroll. Semi-dedicated impostor attempts were also collected. An equal error rate (EER) of 2.4% was found for this system based on a single attempt per session and after retraining the system on PER-specific development data. On parallel telephone data collected using a telephone version of PER, 3.5% EER was found with landline and around 5% with mobile telephones. Impostor attempts in this case were same-handset attempts. Results also indicate that the distribution of false reject and false accept rates over target speakers are well described by beta distributions. A state-of-the-art commercial system was also tested on PER data with similar performance as the baseline research system.

Keywords: speaker recognition, speaker verification, speech technology, biometrics, access control, speech corpus, variance estimation

Acknowledgments

My first thanks go to my supervisors Björn Granström and Mats Blomberg for encouragement and great patience, and suggestions, feedback and proof-reading during the writing of the thesis.

I would also like to honor Björn and Rolf Carlson and other members of the senior staff for finding money for all the projects and keeping activities at TMH going in general.

Next I am greatly indebted to no less than 277 subjects who jointly provided more than 60 hours of speech to the Gandalf and PER corpora, the total time spent being more than three times that. Most of the results in this thesis are based on you! Many thanks also to Karin Carlsson and Ekaterina Melin for annotating PER data, and Antonio de Serpa-Leitão for much help with the collection of Gandalf data.

Friends and room mates Anders Roxström, Roger Lindell, Johan Lindberg, Botond Pakucs and Per-Anders Jande and other colleagues at TMH have made my time at the Department a memorable time. Extra thanks to Johan for many fruitful discussions on speaker verification and for co-authoring papers, Botond for joint PER development and interesting discussions on many aspects of speech technology, and Daniel Neiberg and Daniel Elenius for good discussions on speaker verification and for using and contributing to the development of GIVES, Giampiero Salvi for creating acoustic models for speech recognition that I used extensively, and Becky Hincks and David House for providing native guidance in the English language (remaining mistakes are naturally my own).

Much of the work consisted of programming and simulation work with computers carefully taken care of by Roger Lindell, Johan Lindberg, Tor Fodstad, Adam Soltan, Niklas Horney, Peter Reuterås, Payam Madjidi and Johan Berglund & Co over the many years I spent at TMH.

Participants of COST250 and COST275 have contributed to widen my knowledge in the areas of speaker recognition and biometrics, and our meetings have allowed me to visit many cities in Europe and see the best of their hospitality and food culture.

During 1994–1997, my work was supported by various parts of Telia. I would like to thank Bertil Lyberg, Fred Lundin, Kurt Fridh, Erik Sundberg and Tomas Nordström at Telia for many inspiring discussions. Tomas also for valuable mentorship during the years after he left Telia.

During 1998–2006, most of the research was carried out at the Centre for Speech Technology (CTT), a competence centre at KTH, supported by VINNOVA (The

Swedish Agency for Innovation Systems), KTH and participating Swedish companies and organizations.

I am also grateful to Roland Elander at JustDirect (during parts of 2001–2002) and Bengt Persson, Johan Thunström and others at Voice Provider (during January 2005) for giving me good opportunities to try my knowledge in the “real world”.

Thanks also go to an anonymous company for making the commercial ASV system available for experiments reported in this thesis.

This thesis was typeset in LaTeX by the author with useful help from Botond, Giampiero, Per-Anders and Karolina Smeds.

Finally, I would like to thank my family, Ekaterina and Alexander, my parents Ulf and Birgitta, and my brother Kennet for constant support and encouragement.

Contents

Definitions

Term	Meaning
claimant	a person who interacts with a verification system to verify a claimed identity (alternative terms: speaker, actual speaker)
target	the person who's identity is claimed in a verification test
enrollee	a person engaged in the process of enrolling to a verification system (alternative terms: speaker, client, user)
true-speaker test	a verification test where the claimant is the target, i.e. the claimant is making a legitimate identity claim (alternative terms: target test)
impostor test	a verification test where the claimant is not the target, i.e. the claimant is making a fraudulent identity claim
subject	a person who participates in a data collection experiment to provide speech data (alternative terms: speaker)
test speakers	group of subjects who provide speech data for verification tests (alternative terms: test group)
background speakers	group of subjects who provide speech data for other use than verification tests, for example for training background models or pseudo-impostor models (alternative terms: background speakers, non-client speakers)
client	function of subject in a test group who is a legitimate user of a verification system and claims his own identity while using it (as such, a subject normally provides enrollment and true-speaker test data) (alternative terms: true speaker, regular user, target, legitimate user)
legitimate user	(enrolled) claimant during a true-speaker test (alternative terms: client, true speaker)
impostor	**1** function of subject in a test group when making an impostor attempt; **2** the claimant during an impostor test
background speaker	subject in a group of background speakers
target model	model to represent the voice of a target in a verification test or an enrollment session (alternative terms: client model)
background model	model to represent voices of other speakers than a given target (alternative terms: non-client model)

List of abbreviations

API	application programming interface
ASR	automatic speech recognition
ASV	automatic speaker verification
ATM	automatic teller machine (Swedish: *bankomat*)
CI	confidence interval
CRS	call response sheet
DET	detection error trade-off (in DET curve or DET plot)
DTW	dynamic time warping
EER	equal error rate
EM	expectation maximization
FAR	false accept rate
FFT	fast Fourier transform
FRR	false reject rate
GMM	Gaussian mixture model
GUI	graphical user interface
HMM	hidden Markov model
HTER	half total error rate
IPC	inter-process communication
ISDN	integrated services digital network
JDBC	Java database connectivity (Java SQL database API)
JSAPI	Java speech API
LPCC	linear prediction cepstral coefficients
MAP	maximum a posteriori
MFCC	mel frequency cepstral coefficients
ML	maximum likelihood
pdf	probability density function
PIN	personal identification number
SQL	structured query language (database language)
SAPI	speech API (by Microsoft)
SRE	speaking or recording error
SRS	subject response sheet
SVAPI	speaker verification API (by Novell and partners)
TTS	text-to-speech
VQ	vector quantization

List of publications

Refereed journals and conferences

Hennebert, J., Melin, H., Petrovska, D., and Genoud, D. (2000). POLYCOST: A telephone-speech database for speaker recognition. *Speech Communication*, 31(2-3):265–270.

Karlsson, I., Banziger, T., Dankovicová, J., Johnstone, T., Lindberg, J., Melin, H., Nolan, F., and Scherer, K. (2000). Speaker verification with elicited speaking styles in the VeriVox project. *Speech Communication*, 31(2-3):121–129.

Lindberg, J. and Melin, H. (1997). Text-prompted versus sound-prompted passwords in speaker verification systems. In *Proc. 5th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 851–854, Rhodes, Greece.

Melin, H. (1996). Gandalf - a Swedish telephone speaker verification database. In *Proc. 1996 International Conference on Spoken Language Processing (ICSLP)*, pages 1954–1957, Philadelphia PA, USA.

Melin, H. (1998). On word boundary detection in digit-based speaker verification. In *Proc. Workshop on Speaker Recognition and its Commercial and Forensic Applications (RLA2C)*, pages 46–49, Avignon, France.

Melin, H., Koolwaaij, J., Lindberg, J., and Bimbot, F. (1998). A comparative evaluation of variance flooring techniques in HMM-based speaker verification. In *Proc. 1998 International Conference on Spoken Language Processing (ICSLP)*, pages 1903–1906, Sydney, Australia.

Melin, H. and Lindberg, J. (1999b). Variance flooring, scaling and tying for text-dependent speaker verification. In *Proc. 6th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1975–1978, Budapest, Hungary.

Nordström, T., Melin, H., and Lindberg, J. (1998). A comparative study of speaker verification systems using the Polycost database. In *Proc. 1998 International Conference on Spoken Language Processing (ICSLP)*, pages 1359–1362, Sydney, Australia.

Other

Melin, H. (1999a). Databases for speaker recognition: Activities in COST250 working group 2. In *Proc. COST 250 Workshop on Speaker Recognition in Telephony*, Rome, Italy.

Melin, H. (1999b). Databases for speaker recognition: working group 2 final report. In Falcone, M., editor, *COST250 Speaker Recognition in Telephony, Final Report (CD-ROM)*. European Commission, DG XIII-B, Brussels.

Melin, H. (2001). ATLAS: A generic software platform for speech technology based applications. *TMH-QPSR*, 42(1):29–42.

Melin, H., Ariyaeeinia, A., and Falcone, M. (1999). The COST250 speaker recognition reference system. In Falcone, M., editor, *COST250 Speaker Recognition in Telephony, Final Report (CD-ROM)*. European Commission, DG XIII-B, Brussels.

Melin, H. and Lindberg, J. (1996). Guidelines for experiments on the Polycost database (version 1.0). In *Proc. COST 250 Workshop on Applications of Speaker Recognition Techniques in Telephony*, pages 59–69, Vigo, Spain.

Melin, H. and Lindberg, J. (1999a). Guidelines for experiments on the Polycost database (version 2.0). In Falcone, M., editor, *COST250 Speaker Recognition in Telephony, Final Report (CD-ROM)*. European Commission, DG XIII-B, Brussels.

Melin, H., Sandell, A., and Ihse, M. (2001). CTT-bank: A speech controlled telephone banking system - an initial evaluation. *TMH-QPSR*, 42(1):1–27.

Part I

Introduction

Chapter 1

Introduction

Speaker verification is the biometric task of authenticating a claimed identity by means of analyzing a spoken sample of the claimant's voice. It is a binary detection problem where the claimant must be classified as the true speaker or as an impostor (Atal, 1976; Doddington, 1985; Furui, 1997; Campbell, 1997). Thus, two kinds of errors may occur: the false rejection of a genuine customer or the false acceptance of an impostor.

There are several potential applications where automatic speaker verification (ASV) could be used. Examples include *telephone-based* applications like voice dialing, telephone banking, telephone shopping and password reset systems where users access a service remotely from any location; and *on-site* applications like access control, border control and service kiosks where users are physically present at a point of entry.

On-site applications of speaker verification may offer much freedom in the choice of data collection equipment, the design of user interfaces, and means of controlling user behavior and the background environment. This facilitates the collection of high-quality speech samples under controlled conditions to allow for accurate speaker verification. But with physically present users, there are also other biometric technologies to choose from, such as finger print sensors, face detectors and iris scanners (Jain et al., 1999).

With telephone-based applications, on the other hand, speaker verification has the advantage over other biometric technologies in that a microphone and a speech transmission channel are already available in every telephone, while using other biometric traits than voice would require attaching additional sensor devices to telephone instruments. But designers of speaker verification systems have little possibility to influence what telephone instruments are used, resulting in varying input signal characteristics between users and possibly even between calls by a single user. They also have little possibility to control users or their acoustic environment, and user interfaces are often limited to speech, audio and button type interfaces. Therefore, compared to on-site applications, telephone-based applications of speaker verification generally need to deal with a wider range of inter-session variability not directly related to speakers' voice characteristics.

The present thesis deals with several different topics in automatic speaker verification. Much of the included work are related to a practical application of ASV technology realized at KTH, Department of Speech, Music and Hearing: an access control system that provides staff and students working at the Department on a regular basis with a means to unlock the central gate to their workplace. The system, called PER[1], is an example of a text-dependent on-site application of ASV. Chapter 3 describes the ASV-part of the system in detail, while Chapter 4 describes a generic software framework on which the system was built. Chapter 5 then briefly describes the PER system itself. In Chapter 10, experiments and results with the system are presented. An important point here is a comparison between ASV performance in the on-site application with performance in a corresponding telephone application of ASV. This was made possible through a parallel data collection of on-site and telephone data. This data is packaged as a speaker verification corpus (the PER corpus) and presented in Chapter 6. Another assessment-related chapter deals with error rate estimation methods and uses PER data as an evaluation corpus (Chapter 7).

Other thesis topics not directly related to the PER system include work done using a telephone speaker verification corpus, Gandalf, also described in Chapter 6. This data was used in experiments around two widely different aspects of ASV system design: password prompting strategies (Chapter 8) and variance estimation in hidden Markov models with mixture Gaussian state-observation densities (Chapter 9).

1.1 Outline

The thesis is divided into five parts consisting of one of more chapters.

Part I gives an introduction to and provides a background for the thesis.

Chapter 1 (this chapter) is the introduction.

Chapter 2 gives an overview of the field of speaker recognition with particular emphasis on the speaker verification task. It describes the current state-of-the-art in recognition methods as well as tools and methods for assessment of system performance.

Part II includes work on the development of speaker verification systems and of interactive systems where speaker verification is included as a component technology.

Chapter 3 describes a text-dependent speaker verification system. It also describes a text-independent speaker verification system and a joint system based on a combination of the text-dependent and the text-independent system subsequently

[1] pronounced as [p'æːr]

used in the PER application. A generic software platform used to implement the mentioned systems is also introduced.

Chapter 4 presents a software framework for building prototype speech technology based applications. Example applications built with the framework are described shortly.

Chapter 5 describes various aspects of the on-site and telephone versions of the PER system, including enrollment and access procedures.

Part III includes work on methods and tools for assessment of speaker verification performance.

Chapter 6 presents in detail two corpora collected for the purpose of speaker verification system development and evaluation. Work on experiment design related to two other speaker verification corpora is also described.

Chapter 7 looks at estimation methods for verification error rate, in particular the use of parametric score distribution models and Bayesian methods for the estimation of error rate in individual target speakers.

Part IV includes work on the evaluation of speaker verification technology.

Chapter 8 compares aural and visual strategies for prompting password strings to subjects in terms of speaker verification error rate and the rate of generated speaking-errors in subject responses.

Chapter 9 deals with variance estimation in the context of the text-dependent HMM system presented in Chapter 3. In particular, several variance flooring methods are proposed and evaluated.

Chapter 10 presents and discusses results from practical use of a speaker verification system in the on-site and telephone versions of the PER application. The importance of factors like appropriateness of development data, amounts of enrollment and test data, and system fusion effects are investigated.

Part V concludes the thesis with a summary, a general discussion and suggestions for future work.

Chapter 2

Speaker Recognition

Speaker recognition is a biometric technique for recognizing the identity of a person. Biometric techniques in general make use of some observation of biological phenomena in human beings. Such phenomena can usually be classified into *physical traits* such as finger prints, retina patterns or hand shape, or *behavioral patterns* such as handwriting signature or keyboard typing patterns. Physical traits are inherent in humans, while behavioral patterns are mainly learned. Speech is the result of a combination of physical traits and behavioral patterns. It is a behavioral pattern in the sense that speaking is something we do. Children grow up learning to speak in an environment influenced by many social and linguistic factors, including language, dialect and social status, and this environment influences how we speak. But the sound of a person's speech is also strongly affected by physical characteristics such as the size and shape of their vocal folds, vocal tract and nasal cavities, and these factors also influence how we speak and how the speech sounds. Somewhat simplified we could say that physical factors are more important for segmental features in the speech signal, such as voice quality and formant positions, while environmental (learned) factors are more important for supra-segmental features, such as prosody and word selection.

Whatever factors influence features of a person's speech, in the surface form they are all integrated into a one-dimensional time signal, the speech signal. Unless we have access to specialized measurement devices such as electropalatographs or electroglottographs (which are clearly not viable collection devices in a biometric system), or video cameras (which clearly may be viable), all information from the speech process must be extracted from a sampled version of the speech signal captured through one or more microphone devices. It is not trivial to separate information from each specific source mentioned above from this signal, and therefore it is tempting to treat the speech signal, or some tractable mathematical representation of it, such as a spectral representation, as a pattern to recognize. This may also be a possible approach for speaker recognition, as long as one keeps in mind that the speech signal hosts many types of variability and must be treated in a statistical sense rather than as a static image. Transferring the concept of a finger print into a "voice print" lies near at hand, but may be misleading (Bonastre et al., 2003a). In general, an important difference between physical traits and behavioral patterns is that the latter involve a time dimension and thus usually can only be captured through a sequence of observations, while a snapshot may be enough to capture the former.

The success of speaker recognition for biometric purposes depends on whether features of the speech signal, or the gross speech signal itself, can be measured and modeled in such a way that the model can be used to discriminate sufficiently well between individual speakers.

This chapter gives an overview of the field of automatic speaker recognition. It establishes the current state-of-the-art in recognition methods as well as tools and methods for assessment of system performance.

2.1 Task taxonomy

Speaker recognition as a task is usually divided into *speaker verification* and *speaker identification*. Speaker verification is the two-class problem of determining if an identity claim is true or false. Is the speaker (the *claimant*) who he claims to be (the *target*), or an impostor? A speaker verification system can make two types of error: a *false reject error* when a legitimate claim is rejected, or a *false accept error* when an impostor is accepted. *Speaker authentication* and *speaker detection* are equivalent terms for speaker verification. In conjunction with the term speaker detection, the two types of error are usually called *miss* and *false alarm* (e.g. Martin et al., 1997).

Speaker identification is the N-class or $N+1$-class problem of determining which out of N known target speakers is the current speaker. The problem is $N+1$-class in the open set case, where "none of them" is a possible answer, i.e. there is a rejection alternative.

Related tasks are speaker tracking, speaker change detection, speaker clustering and speaker diarization. Speaker tracking involves "following" a given target speaker during a conversation (Martin and Przybocki, 2000), while speaker change detection is to detect when the speaker changes during a conversation. Speaker clustering is to group speakers according to similarity defined by some similarity measure. Speaker diarization (Gravier et al., 2004; Ben et al., 2004) involves assigning a speaker label to every speaker turn in a conversation and to group turns spoken by the same speaker. Speakers are not known beforehand.

Speaker verification or detection can also be formulated in a multi-speaker context, where the problem is to detect if a target speaker is present in a conversation (Martin and Przybocki, 2000).

For a speaker recognition system to be able to recognize the speech of a known target speaker, the system must have prior access to a sample of speech from this speaker. The subtask of creating a target model from a sample of speech is called *speaker enrollment*.

This thesis deals with speaker verification in a single-speaker context.

2.2 Text dependence

We use the term *text dependence* to describe the relation between enrollment and test speech required by a speaker verification system. A system is text-dependent if it requires the same text to be spoken during enrollment and test, and text-independent if the same text does not need to be spoken. However, spoken *text* includes phrases, words, syllables, phonemes, prosody, etc., and therefore text dependence can be described using a more fine-grained scale going from most text-dependent to most text-independent (Bimbot et al., 1994):

1. *text-dependent using a fixed passphrase shared by all users* (e.g. Furui, 1981): all users speak the same passphrase during enrollment and test. Such a system is not likely to be used in a real application, but it provides a way to test speaker discriminability in a text-dependent system.

2. *text-dependent using fixed user-dependent passphrases* (e.g. Bernasconi, 1990; Higgins and Bahler, 2001; BenZeghiba and Bourlard, 2002; Bonastre et al., 2003b): a given user speaks the same phrase during enrollment and test.

3. *vocabulary-dependent* (e.g. Rosenberg et al., 1991; Schalk, 1991; Netsch and Doddington, 1992): users enroll by speaking examples of all words in a given vocabulary (e.g. digits 0 through 9, spelling-word sequences, or a small set of arbitrary words fitting into a carrier phrase (e.g. Doddington, 1985)) while test utterances are constructed from subsets of the vocabulary. Words do not necessarily need to appear in the same order during enrollment and test.

4. *event-dependent* (e.g. Bonastre et al., 1991; Gupta and Savic, 1992; Reynolds et al., 2003): the system models particular "events" in the speech signal, e.g. particular phonemes; word or bigram use; or the occurrence of grammatical errors. The text may be different during enrollment and test as long as the modeled events occur in sufficient numbers.

5. *text-independent, system-driven* (e.g. Matsui and Furui, 1994): text does not need to be the same during enrollment and test. However, the system prompts what text users shall speak, thus, the system knows what text to expect.

6. *text-independent, user-driven* (e.g. Reynolds, 1994; Bimbot and Mathan, 1994): the fully text-independent system, where users can say anything during enrollment or test.

Note that with this definition of text-dependence, a text-prompted ASV system, where the system prompts claimants with a text to speak (Higgins et al., 1991), may be either vocabulary-dependent, event-dependent or text-independent with system-prompted text. Also note that vocabulary-dependent and event-dependent systems need not be used with text-prompting, since an ASV system could be set up to rely on automatically spotting modeled words or events in text chosen and spoken by claimants.

2.3 Inter-speaker and intra-speaker variability

It is apparent that the speech of most people sounds different. Also, one single person's speech is likely to sound a bit different from time to time. This is obvious in the case of a person having a bad cold, for example. *Inter-speaker variance* is a measure of the variation in voices between people, while *intra-speaker variance* is a measure of the variation of one person's voice from time to time.

To illustrate the terms inter-speaker and intra-speaker variance, we consider a fictitious (two-dimensional) speech signal representation that spans a speaker space in Figure 2.1. Four speakers $S1$–$S4$ have been sampled and samples been plotted in the speaker space. Call the area including all the samples of a given speaker the speaker's spanned *subspace*. Loosely, *intra-speaker variance* for a given speaker is the average squared distance between samples from this speaker and the center of his subspace. Square roots of intra-speaker variances for speakers $S1$ and $S2$ are illustrated in the figure as distances $b1$ and $b2$. The mean intra-speaker variance in the speaker space is the average intra-speaker variance taken over all speakers in the space. *Inter-speaker variance* is the average squared distance between each speaker's subspace center and the center of the speaker space. Distance a in the figure illustrates the square root of the inter-speaker variance in our fictitious speaker space.

More precisely, inter-speaker variance for one-dimensional observations $x_n^{(i)}$ (observation n from speaker i) is the variance of speaker means

$$V_{\text{inter}} = E_i[(\mu_i - \bar{\mu})^2)] \tag{2.1}$$

where $\mu_i = E_n(x_n^{(i)})$ is the mean of observations from speaker i and $\bar{\mu} = E_i(\mu_i)$ is the overall mean value over all speakers. Define intra-speaker variance (for one-dimensional observations) for a single speaker i as

$$V_{\text{intra}}^{(i)} = E_n[(x_n^{(i)} - \mu_i)^2] \tag{2.2}$$

and the average intra-speaker variance

$$V_{\text{intra}} = E_i(V_{\text{intra}}^{(i)}). \tag{2.3}$$

Based on the terms inter-speaker and intra-speaker variance we can now discuss some of the issues that are central to the speaker recognition problem.

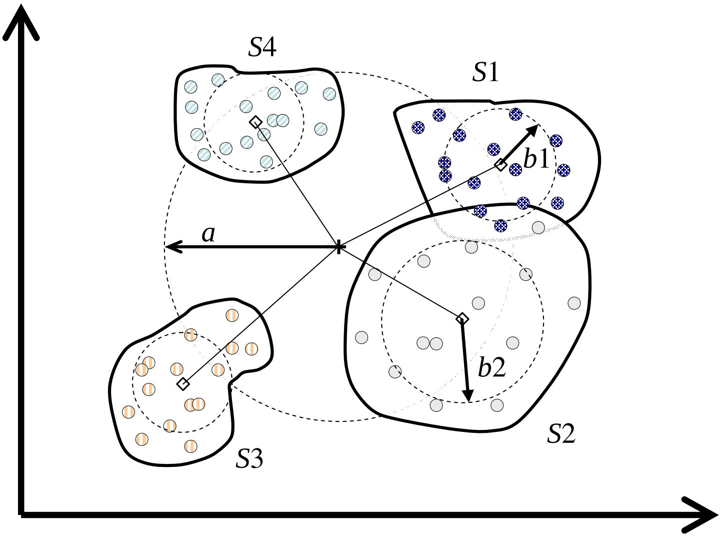


Figure 2.1: Four speakers ($S1$–$S4$) sampled in a fictitious two-dimensional speaker space to illustrate (square roots of) inter-speaker variance (a) and intra-speaker variance ($b1$, $b2$).

2.3.1 Speaker discriminability

To be able to discriminate between speakers, we need to measure and represent the speech signal suitably. For a particular (one-dimensional) speech signal representation to be efficient for speaker recognition, we want measurements from speakers in this representation to span small subspaces for each speaker, while subspaces for all speakers are well separated. This is to say that we want a large inter-speaker variance and a low intra-speaker variance, or a large F ratio (Wolf, 1972; Atal, 1976)

$$F = V_{\text{inter}}/V_{\text{intra}}. \tag{2.4}$$

To deal with multi-dimensional observations, inter-speaker and intra-speaker variances need to be extended to include the effect of correlation between observations. An extension of the F ratio into the multi-dimensional case is the Kullback divergence defined on an inter-speaker covariance matrix and an intra-speaker covariance matrix. See e.g. Atal (1976) for details. For our discussion of speaker recognition concepts we will settle for the simpler (more intuitive) terms inter-speaker and intra-speaker variance.

While the F ratio (divergence) is an overall measure for a group of speakers, there will naturally be variations in speaker discriminability within the speaker group. It is also well known that both false reject and false accept errors are usually unevenly distributed among target speakers (and impostors) for a given speaker verification system and a group of speakers. The terms *goat* and *sheep* are sometimes used to refer to target speakers that experience a high (goats) or low (sheep) false reject rate. Similarly, the terms *wolf* and *lamb* can be used for speakers that are often successful impostors (wolves) and target speakers that suffer a high false accept rate (lambs) (Campbell, 1997; Doddington et al., 1998).

Several *a priori* factors can be used to predict smaller expected inter-speaker distances between a given pair of speakers than between any two randomly chosen speakers. For example, identical twins can be expected to be positioned closer in the speaker space than a pair of random speakers (Cohen and Vaich, 1994; Homayounpour and Chollet, 1995; Dialogues Spotlight Consortium, 2000), and trained impersonators may be able to alter their speech to move closer to a given target in the speaker space (Luck, 1969; Ashour and Gath, 1999; Dialogues Spotlight Consortium, 2000; Elenius, 2001; Zetterholm et al., 2004). Impostor attempts by technical means (Lindberg and Blomberg, 1999; Genoud and Chollet, 1999; Pellom and Hansen, 1999; Matrouf et al., 2006) may also reduce inter-speaker distances between (virtual) impostors and a given target speaker. Speakers of the same language origin as a target speaker are likely to be more successful impostors than those with a different language background (Nordström et al., 1998), etc.

While better-than-average impostors may be predicted by *a priori* factors, it is more difficult to predict which target speakers will suffer high false reject rates due to large intra-speaker variability, maybe combined with low inter-speaker variability. Attempts have been made to predict such "goats" by means of automatic measurements on enrollment speech and/or comparison with other target models (Thompson and Mason, 1994; Koolwaaij et al., 2000; Gu et al., 2000).

2.3.2 Speaker variability

We usually want to be able to recognize speakers over long periods of time, and not only just after they enrolled to a speaker verification system. However, the human voice changes both long-term, mainly due to aging (Linville, 2001; Winkler et al., 2003), and on shorter terms due to other factors such as health (respiratory infections, head colds), speech effort level and speaking rate, emotional state (Murray and Arnott, 1993; Karlsson et al., 2000), vocal tiredness and user experience (list inspired by (Doddington, 1998)). "Random" (in lack of better understanding) inter-session effects can also be expected (Doddington, 1998). Thus, a desirable property of a speech signal representation for speaker recognition is therefore long-term stability, or robustness to variability in the speaker.

With our illustration of speaker subspaces in Figure 2.1 and in terms of intra-speaker variance, this means that unless our signal representation is independent of short and long-term variability in a speaker, this speaker's subspace is not static – it changes with time. Changes may be either translations of the subspace or changes in the intra-speaker variance. Furui (1986) found that the intra-speaker variation in a cepstrum representation of long-time average spectrum computed from samples collected over increasing intervals of time and averaged over nine male speakers, increased monotonously for intervals up to three months and was nearly constant for longer intervals. This can be interpreted in our graphical illustration that speaker subspaces can be expected to "grow" during the first couple of months, after which new samples will mainly fall within the already observed subspace.

2.3.3 Robustness

In addition to the speech waveform, a recorded signal may contain acoustical background noise and the effects of microphone characteristics and electrical transmission. Both noise and transmission effects may vary in amount and type both within and between recording sessions. From a speaker recognition point of view, the resulting effect is that the speech signal from a speaker is mixed with other variability factors that (usually) don't depend on the speaker. The main challenge here is not so much the presence of noise and transmission effects (as long as they are not too great relative to the speech signal itself), but that speaker-independent effects may *vary*. With our graphical illustration (Figure 2.1), additive noise in our signal representation has the effect of "blurring" subspaces and making them bigger. It makes intra-speaker variance (as defined above on the observed signal rather than on the embedded speech signal) larger and thus decreases the F ratio. A linear channel transfer function that changes from one session to another, has the effect of moving samples within the space, possibly making speakers' subspaces larger.

Attempts at making speaker recognition robust against noise and transmission effects usually follow one of the following approaches: selecting robust features or modeling techniques, removing unwanted effects from the signal, normalizing the signal, or adapting the speech models. Robustness methods can be applied to different parts of a speaker recognition system, such as the speech signal itself, the signal representation (feature domain), speech modeling, likelihood scores and the decision logic. Some of the methods reported in the literature will be reviewed below.

2.3.4 Modeling

A speaker recognition system needs to have some kind of model of each target speaker to recognize. The model can be either *similarity-based* or *discrimination-based* (or a combination). A similarity-based model represents how the speaker sounds, while a discrimination-based model represents what is different between the given speaker and other speakers. In the context of our illustration (Figure 2.1), the former aims to describe a speaker's subspace in the signal representation defining our speaker space, while the latter aims to describe differences between a given speaker's subspace and other speakers' subspaces.

In addition to target models, representing enrolled users, many similarity-based speaker recognition systems also use one or more background models to represent other speakers, or groups of speakers. Discrimination-based systems often use speech from other speakers when computing the discrimination model rather than using separate models of other speakers.

2.4 Methods

2.4.1 Feature extraction

Short-time spectral representations have been among the most popular speech feature types in experiments on automatic speaker recognition during the past two decades. These are usually cepstral forms of the spectrum based on a linear prediction model, a filter bank, or more sophisticated perceptually motivated models of the speech signal. The same feature types are also frequently used in automatic speech recognition. This may appear strange since speaker and speech recognition systems have different goals in extracting information from a spoken utterance: the former is looking for the speaker and the latter for the text message. But obviously, the short-time spectrum carries information about both the speaker and the message (along with other variabilities as stressed above). These feature types may be suboptimal for the respective recognition tasks. While early research in speaker recognition was directed at finding good features (e.g. Wolf, 1972; Atal, 1974; Sambur, 1975), focus has later shifted towards the use of statistical models. By training statistical models on the entities we want to recognize, such as a word, a word spoken by a particular speaker, or general speech from a particular person, the models do much of the job of extracting speaker or text-specific features "hidden" in the signal. The choice of low-level representation of the speech signal is then less critical, as long as the statistical models can capture the speaker- or text-specific variation present in them.

2.4.1.1 Linear prediction

In linear predictive coding (LPC), a segment of a speech waveform is represented parametrically by a discrete linear all-pole filter (Atal, 1974). Specifically, parameters of the all-pole filter are fitted such that the filter's transfer function matches the spectral envelope of the speech signal. Several methods have been suggested for fitting filter parameters to the signal, such as the covariance method, the autocorrelation method and the lattice method (see e.g. Huang et al. (2001, chapter 6) for details). The use of the LPC model in representing speech signals is motivated from a simplified speech production perspective, because of the all-pole filter's equivalence to electrical analogs of the vocal tract approximated as a series of short lossless tubes with varying diameter (Fant, 1960).

Often a cepstrum representation of the linear prediction spectrum (linear prediction cepstral coefficients, LPCC), which can be efficiently computed from prediction coefficients through a recursive formula, is used (e.g. Furui, 1981). Various other alternate forms related to the prediction parameters have also been used, such as line spectral (pair) frequencies (LSF or LSP) (Liu et al., 1990; Bonifas et al., 1995;

Campbell, 1997; Magrin-Chagnolleau et al., 2000), reflection coefficients (or partial correlation coefficients (PARCOR)), and log-area ratios (Higgins et al., 1991).

2.4.1.2 Filter banks and auditory models

While the LPC model is motivated from a speech production point-of-view, filter bank-based representations are motivated from a speech perception point-of-view, because the operation of the basilar membrane in the human ear can be modeled by a bank of band-pass filters.

A commonly used instance of a filter bank-based cepstral feature type is called Mel Frequency Cepstral Coefficients, or MFCC (Davis and Mermelstein, 1980). Here, center frequencies of band pass filters are equally spaced on a mel scale, and cepstrum vectors are computed from filter log amplitudes through a cosine transform. The filter bank is usually implemented through weighted sums of FFT points. Many parameters in the computation of MFCC vectors can be varied, such as the number and shape of filters and the number of cepstral coefficients. One particular setup has been standardized by ETSI as a front-end for distributed speech recognition[1]. This setup, including the VQ compression algorithm, was tested for speaker recognition by Broun et al. (2001) on the YOHO corpus and simulated GSM channels with good results.

The use of a filter bank and its frequency spacing according to the mel scale in the MFCC can be motivated as approximations of a basic psychophysical function in the human auditory system, namely the frequency resolution in the cochlea (critical bands).

Other signal representations have been proposed based on more sophisticated models of the auditory system. The perceptual linear prediction (PLP) representation (Hermansky, 1990) is the most well-known example. It simulates three concepts from psychophysics: the critical-band spectral resolution, the equal-loudness curve and the intensity-loudness power law. These concepts are implemented in a filter bank to compute an auditory spectrum which is then approximated by an all-pole filter like in the LPC case. PLP is often used in combination with RASTA processing mentioned below.

Other auditory-based signal representations were tested by Quatieri et al. (2003). Their work is based on two auditory models by Dau et al. (1996) and Dau et al. (1997). The first model simulates low-level processing in the cochlea and have similarities to MFCC and PLP processing. The second model simulates the processing of amplitude modulation in the brain of signals from the cochlea and is implemented through filter banks applied to the output of each auditory channel output from the first, low-level model. Improvements in speaker verification tasks were found when features from the two auditory models were combined (through likelihood score fusion) with MFCCs, compared to using the MFCCs alone.

[1] ETSI ES 201 108: "Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms", http://www.etsi.org

Colombi et al. (1993) tested a biologically motivated model of the auditory system, the Payton model (Payton, 1988). This model does not use a filter bank to simulate auditory processing, but simulates the dynamics of the basilar membrane. Its output is predicted neural firing responses for a number of points along the basilar membrane. Speaker recognition results comparable to those with LPCC features were found, however still with classifier techniques (VQ) and distance measures originally developed for cepstral features.

2.4.1.3 Other features and supplementary processing

Several other types of features have been tried in the context of automatic speaker recognition, for example prosodic features, voice source features, articulatory features and data-driven features. A number of feature post-processing methods have also been proposed to deal with for example noise and between-session channel variation.

Prosodic features have drawn much interest, mainly as a complement to short-time spectral representations, though early research in ASV also considered for example pitch as a stand-alone feature (Atal, 1972). Pitch have also been shown to be important for human perception in discriminating between speakers (Furui, 1986), and pitch and delta-pitch features have been successfully used in text-independent automatic speaker recognition (Matsui and Furui, 1990; Carey et al., 1996; Sönmetz et al., 1997). Duration and energy features have also been tested, for example by Shriberg et al. (2005) who combined duration, energy and pitch features at the syllable level through so called SNERFs (N-grams of Syllable-based Nonuniform Extraction Region Features). In all cases, ASV error rates were reduced when prosodic features were (somehow) combined with spectral features, compared to spectral features alone.

Voice source feature approaches include using parameter values of proposed voice source models that are automatically derived from the speech signal. For example, Darsinos et al. (1995) and Plumpe et al. (1999) employ the Liljencrants-Fant (LF) model (Fant, 1986), while Slyh et al. (2004) use a modified version of the Fujisaki-Ljungqvist model (Fujisaki and Ljungqvist, 1986). Also the voice source features were able to boost an ASV system based on spectral features. The LPC residual (error prediction) signal, which can be seen as coarsely related to the voice source, has also been tried for ASV purposes (Thévenaz and Hügli, 1995).

Articulatory features have recently been tried for ASV with promising results (LcLaughlin et al., 2002; Leung et al., 2004, 2005).

Feature sets have also been derived using data-driven methods, where the basic idea is to calculate optimal (in some sense) speech signal representations based on a representative corpus of speech. Features have been constructed through linear (Naik and Doddington, 1986; Olsen, 1998b) or non-linear (Konig et al., 1998; Chao et al., 2005) transforms of some short-time spectral representation, or directly from waveform samples (Chetouani et al., 2004). The resulting features can be target-dependent or target-independent.

In addition to basic feature types described thus far, several types of supplementary processing of features have been tried.

Delta features (Furui, 1981; Soong and Rosenberg, 1986) are complementary to static features and have been claimed to be more robust than static features to between-session channel variation.

The well-known and often used Cepstral Mean Subtraction (CMS) (Atal, 1974; Furui, 1981; Rosenberg et al., 1994) and RASTA (Hermansky et al., 1991; Morgan and Hermansky, 1992) methods aim at removing the influence of between-session channel variation and are often used with speech collected through telephone channels.

Feature Warping (FW) (Pelecanos and Sridharan, 2001) maps individual cepstral coefficients such that their distribution during a time window in the test signal is mapped (warped) into a pre-determined target distribution, for example a standardized normal distribution. This non-linear mapping technique gives more importance to the relative value of a cepstral feature than to its absolute value. The technique is claimed to be robust to linear channel effects and additive noise. Short-time Gaussianization (Xiang et al., 2002) is proposed as a simplified case of Gaussianization (Chen and Gopinath, 2001) and is equivalent to a linear transform followed by the same type of transform used in FW.

Cepstral Mean and Variance Normalization (CMVN) (Koolwaaij and Boves, 2000) is an extension to CMS that in addition to translating cepstral features to have zero mean also scales them to have a fixed variance. The main difference between CMVN and FW is that in CMVN, the assumed type of distribution of a feature in the test signal over a sliding window and the target distribution are fixed (normal distributions), while FW, in principle, do not assume a specific shape of the two distributions.

Feature Mapping (Reynolds, 2003) is a model-based feature transform that aims to map feature vectors into an all-channel space. Mappings are computed on a mixture-to-mixture basis from the top-scoring mixture term in a channel-dependent GMM to the corresponding mixture term in an all-channel “root” GMM, where channel-dependent models are MAP-adapted from the root GMM. The method requires some kind of channel detection to identify the most suitable channel-dependent GMM.

2.4.2 Classification

2.4.2.1 Similarity-based methods

Early speaker recognition experiments often used long-time averages of some speech signal representation, and a suitable distance measure, to compare speech from a known target speaker to an unknown sample. At that point the choice of representation was crucial for recognition performance.

Starting in the 1970s, dynamic programming techniques were employed in speech science to compare sequences of observation vectors (*templates*), taking into account the dynamic and non-linear variation of speech in time (Sakoe and Chiba, 1978; Rabiner et al., 1978). This technique, also known as *dynamic time warping* (DTW) in the speech community, was also used successfully for speaker recognition purposes (e.g. Naik and Doddington, 1986). DTW enabled text-dependent modeling of speakers.

In the 1980s, vector quantization techniques were borrowed from the source coding field for use in speaker modeling (Rosenberg and Soong, 1987; Burton, 1987). A vector quantizer is defined by a number of centroid vectors (the *codebook*) and a rule to assign observation vectors to centroids, using some distance measure to compute the distance (distortion) between centroid vectors and observation vectors. To use the quantizer for speaker recognition, a codebook is trained for each target speaker by optimizing the location of the centroid vectors relative to the target's enrollment speech to minimize the over-all distortion. The test metric is the average distortion introduced when using a target's codebook to code a test utterance – the smaller distortion, the more similarity between enrollment and test speech. The introduction of VQ techniques improved modeling of speakers relative to the use of long-time averages because the distribution of observation vectors in the speaker space was captured in some way, not only the location of the distribution mean. The VQ technique is inherently text-independent since the order of observation vectors is not modeled.

A similar technique to VQ is Nearest Neighbor (NN) classification (Higgins et al., 1993). While in VQ we represent speakers by clusters of their enrollment data, with NN we use the unclustered enrollment data. Each vector in a test utterance is compared to enrollment data by finding the nearest vector according to some distance measure. The average distance is used as test metric.

While the DTW technique was able to model some of the time variations in speech, it was poor at modeling spectral variation and higher-level variation, such as pronunciation alternatives. The advent of *hidden Markov models* (HMM) in speech recognition provided a better tool for the modeling of spectral dynamics, because they use statistical models instead of template vectors to represent observation vectors in training data. HMMs were also tried for text-dependent speaker modeling, where they were taken to model sub-word units (Rosenberg et al., 1990; Matsui and Furui, 1993), whole words (Rosenberg et al., 1991) or entire passphrases. For a detailed description of HMMs and historical remarks, see e.g. Huang et al. (2001).

When used as target model in speaker recognition, an HMM is trained on repetitions of a chosen unit (sub-word, word or phrase) by the target speaker. During training, the parameters of the HMM are chosen to optimize some criterion, e.g. to maximize the likelihood that the model generated the seen training data. (An overview of training methods and related problems and techniques is given below.) The test metric is then usually the likelihood that the model generated an observed sequence of observation vectors. Depending on the topology of the HMM, a model can be either text-dependent (left-right HMMs (Naik et al., 1989; Rosenberg et al., 1990)) or text-independent (ergodic HMMs (Poritz, 1982; Savic and Gupta, 1990)).

In the special case of a single-state (continuous density) HMM, the model degenerates to being the probability density function (pdf) used for model observations in that state. If the pdf is a multi-variate Gaussian mixture, the model is usually called a Gaussian mixture model.

Gaussian mixture models (GMM) were introduced into the speaker recognition field by Reynolds (1994, 1995). Like in the HMM case, the parameters of a (target) GMM are trained to optimize some criterion defined on enrollment data from a target speaker, and the test metric is the likelihood that a model generated some observed test data. The GMM is inherently text-independent since it does not model the order of observations.

The modeling techniques mentioned so far are all similarity-based, in the sense that a speaker model is meant to describe how a target speaker "sounds", or, to be more precise, to describe how samples from the speaker are distributed in the speaker space defined by a chosen signal representation. DTW, VQ and NN modeling techniques are based on average distances between sample vectors, while HMM and GMM techniques are based on likelihoods through the comparison of sample vectors to statistical distributions. While the distance-based models, and initially also HMM models, have been tried as self-contained classifiers by comparing their test metric to a threshold, *likelihood ratio-like detectors* are today known to give better results. In such detectors, some kind of complementary model of other signal sources than the target speaker himself are used as a reference to compute a relative measure, the ratio, that is used as the decision variable in classification.

Recall that speaker verification is a two-class classification problem. The task is to determine if an unknown sample of speech originates from either the target speaker or an impostor. Denote as ω_t and ω_i the two classes "target speaker" and "impostor speaker", respectively. Further denote as $p(\mathbf{O}|\omega_\mathrm{t})$ and $p(\mathbf{O}|\omega_\mathrm{i})$ the class-conditional pdfs for speech represented by the random variable $\mathbb{O}$, where $\mathbf{O}$ is an outcome of $\mathbb{O}$, i.e. an observed test utterance. When viewed as a function of ω with $\mathbb{O} = \mathbf{O}$ fixed, $p(\mathbf{O}|\omega)$ is a likelihood function, often denoted as $l(\omega|\mathbf{O})$. The log-likelihood ratio for a given $\mathbf{O}$ is then

$$\begin{aligned} \log lr(\mathbf{O}) &= \log l(\omega_\mathrm{t}|\mathbf{O}) - \log l(\omega_\mathrm{i}|\mathbf{O}) \\ &= \log p(\mathbf{O}|\omega_\mathrm{t}) - \log p(\mathbf{O}|\omega_\mathrm{i}) \end{aligned} \tag{2.5}$$

and the log-likelihood ratio detector decides on the target speaker if $\log lr(\mathbf{O}) > \theta$ and an impostor speaker otherwise, where θ is a decision threshold. If the class-conditional pdfs are known exactly, the (log-)likelihood ratio detector is known to be an optimal classifier in the minimum probability of error sense (e.g. Huang et al., 2001). In this case, the decision threshold can also be computed from the *a priori* class probabilities (for minimum error probability classification) and error costs (for minimum error cost classification). However, in practice pdfs are not known exactly but have to be learned from enrollment data. Using one of our pdf-based models (HMM or GMM) and some learning procedure, a straight-forward approximation of the target class-dependent pdf can be implemented using a target model λ_t and a test metric function[2] $P(\mathbf{O}|\lambda_\mathrm{t})$:

$$\log p(\mathbf{O}|\omega_\mathrm{t}) \approx \log P(\mathbf{O}|\lambda_\mathrm{t}). \tag{2.6}$$

An approximation of the impostor class-conditional pdf is not as obvious, since the exact pdf involves all possible impostor speakers, and even all possible sound sources, in the universe. The two initial approaches to likelihood ratio classification in speaker verification used either a *world model* (Carey et al., 1991) or a *cohort model* (Rosenberg et al., 1992) to approximate the impostor class-conditional pdf. Essentially, the world model approach uses a single background model λ_world trained on a large number of speakers (assuming pdf-based models like the HMM or GMM)

$$\log p(\mathbf{O}|\omega_\mathrm{i}) \approx \log P(\mathbf{O}|\lambda_\mathrm{world}), \tag{2.7}$$

while the cohort approach uses the sum over a target-dependent selection of N single-speaker models λ_{c_n}

$$\log p(\mathbf{O}|\omega_\mathrm{i}) \approx \frac{1}{N} \sum_{n \in \mathrm{cohort}} \log P(\mathbf{O}|\lambda_{c_n}). \tag{2.8}$$

(Several variants have been proposed, see (Rosenberg et al., 1992; Tran and Wagner, 2001) for an overview.) The so-called *cohort speakers* are selected by similarity to the target speaker with the aim of creating a background model that is a good approximation of the impostor class-conditional pdf in the neighborhood of the target speaker's subspace in the speaker space. Training data for the world or cohort models must be selected with care to reduce approximation errors in (2.7) and (2.8). This includes selecting speakers, recording conditions, text material, etc. similar to those expected in the application in which the speaker verification system is going to be used.

2.4.2.2 Discrimination-based methods

The Support Vector Machine (SVM) (Vapnik, 1995; Burges, 1998) is a class of discrimination-based binary classifiers that model boundaries between two classes of training data in some (usually high order dimension) feature space, with no intermediate estimation of observation densities. An SVM is characterized mainly by its kernel function.

SVMs seem to be the currently most popular type of discrimination-based method in the ASV research community. They were initially explored in the speaker recognition field by Schmidt and Gish (1996) for a speaker identification task using a polynomial kernel function. Speaker verification experiments with SVMs and different types of kernels include Wan and Campbell (2000); Gu and Thomas (2001);

[2] We avoid referring to the test metric function as a pdf (and write P instead of p) to accommodate for cases where this function is not an exact pdf, such as when a Gaussian pre-selection method is used to truncate a sum, like in Eq. (3.22) on p. 46.

Wan and Renals (2002); Campbell (2002). The latter reference introduces generalized linear discriminant sequence kernels, with the key point of a sequence kernel being to classify entire sequences of observation vectors rather than individual vectors. In all these cases, the input to SVMs were short-time spectral representations of speech.

Many approaches where SVMs are combined with similarity-based models have also been suggested. Fine et al. (2001) use an SVM to perturb GMM scores on frames where the GMM is indecisive. Kharroubi et al. (2001) use scores from individual GMM terms in target and background models as input to an SVM that replaces the traditional log-likelihood ratio computation. Wan and Renals (2002); Moreno and Ho (2003, 2004) use GMMs as part of SVM kernels. Campbell et al. (2004b) combine scores from GMMs and SVMs.

Methods for model-based channel compensation in SVMs have been also proposed (Solomonoff et al., 2005).

SVM approaches with other types of features than short-time spectral features include using prosodic features (Shriberg et al., 2005) and high-level features such as word or bigram frequencies (Campbell et al., 2004a).

The Polynomial classifier (Campbell et al., 2002) is discrimination-based and uses polynomial discriminant functions. It has been used successfully in text-dependent speaker verification. The Polynomial classifier is similar to an SVM with a polynomial kernel.

Earlier approaches to discrimination-based methods include artificial neural networks, such as multi-layer perceptrons (MLP), time delay neural networks (TDNN), radial basis functions (RBF) and the neural tree network (NTN). For an overview and references, see e.g. (Farrell et al., 1994). Recent attempts with new forms of neural networks have also been made (e.g. Ganchev et al., 2004a).

2.4.2.3 Model estimation

This section gives an overview of estimation and adaptation techniques used for creating HMM and GMM-based target speaker models. An estimation technique is defined by an optimization criterion and an algorithm to find parameter values (the model) that optimizes the criterion.

ML estimation Many estimation techniques are based on the Maximum Likelihood (ML) principle. When applied to a target speaker model, the ML principle amounts to finding the model $\hat{\lambda}$ that maximizes the likelihood of enrollment data $\mathbf{O}$ given the model

$$\hat{\lambda} = \arg\max_{\lambda} P(\mathbf{O}|\lambda). \tag{2.9}$$

Usually, an iterative algorithm based on the Expectation-Maximization (EM) method is used. Since the algorithm is iterative, a starting guess for model parameter values is needed. The values of a speaker-independent background model is one possibility.

The EM method is efficient and is guaranteed to lead to a locally optimal ML solution. With good starting values, only a few iterations are usually required. One problem with the original ML/EM technique is that, apart from the information provided through the choice of starting values, all training is based on observed training data. If training data is scarce relative to the target speaker's "true" distribution with respect to his intra-speaker variability, the vocabulary used in the application, etc., the training algorithm will focus on the available data, and there is no fallback for regions of the (true) speaker space that are not represented in training data. In particular, this is often a problem in estimating variance parameters, that require more training samples per parameter than mean parameters. Modifications to remedy this problem have been suggested, including variance flooring (Bimbot et al., 2000; Melin and Lindberg, 1999b)[3] and fixed variances (Matsui and Furui, 1993). With the former modification, the values of variance parameters are not allowed to decrease below a given flooring value. With the latter, variance parameters are not trained at all. Instead, speaker independent values are used, which may for example be copied from a background model.

MAP estimation The Maximum A Posteriori (MAP) criterion provides a way to handle scarce training data by relying on *a priori* information in regions where there are few training data examples. With MAP, the model parameters are treated as stochastic variables with an assumed known *a priori* distribution $P(\lambda)$. The optimization objective is to maximize the *a posteriori* probability of model parameters, i.e. to find the most likely model $\tilde{\lambda}$ given training data

$$\tilde{\lambda} = \arg\max_{\lambda} P(\lambda|\mathbf{O}) = \arg\max_{\lambda} P(\mathbf{O}|\lambda)P(\lambda). \tag{2.10}$$

If output probability distributions in the model are for example Gaussian mixtures, the EM algorithm can again be used to solve the optimization problem (Gauvain and Lee, 1992; Lee and Gauvain, 1996). While the EM algorithm provides an iterative solution, often only a single update is used (e.g. Reynolds et al., 2000), especially when the purpose is (conservative) adaptation of an existing model rather than creating a new target speaker model (Fredouille et al., 2000; Barras et al., 2004). The EM update equations for model parameters have the intuitively appealing form of weighted sums of prior and new information, where the weights are determined by the amount of new data such that if there is much new data available to estimate a given parameter, new data is given more weight; otherwise estimation relies more heavily on prior data. Theoretically, this should allow also variance parameters to be robustly estimated from scarce training data. However, ASV experiments have shown that it may still be better to leave variances unadapted and only adapt mean parameters (and possibly mixture term weights) (Carey et al., 1997; Reynolds et al., 2000; Barras and Gauvain, 2003).

[3] see also Chapter 9

Values for the prior distribution are in practice often determined from a speaker independent background model (Lee and Gauvain, 1996; Reynolds et al., 2000), though a more realistic prior distribution for speaker adaptation would probably result from observing the distribution of model parameters in many single-speaker models.

Discriminative training The decision rule of a likelihood ratio detector is to maximize the *a posteriori* probability of the detected class given observed data, which is known to minimize error rate if class-dependent pdfs are known exactly. When the ML or MAP criteria are used to train the target and impostor class-dependent pdfs (target and background models) separately for use in a likelihood ratio detector, there is a mismatch between the decision rule and the training criterion in the detector. A matching training criterion would optimize target and impostor class-dependent pdfs jointly. Methods based on such criteria are called *discrimination-based methods*, and when used in speaker verification aim at maximizing the acceptance of true speakers, while simultaneously minimizing the acceptance of impostors.

Several discriminative optimization criteria have been proposed for speaker recognition such as Maximum Mutual Information (MMI) training (Li et al., 1995), Minimum Error training (also referred to as Minimum Classification Error (MCE) or Minimum Verification Error (MVE) training) (Liu et al., 1995; Rosenberg et al., 1998), and Generalized Minimum Error Rate (GMER) training (Li and Juang, 2003). Li (2004) showed that these methods are related. For example, MMI is equivalent to minimizing the error rate, while MCE are GMER are more general and flexible. The two latter criteria can be made equivalent to minimum error rate training with particular choices of parameter values in their definition.

If one of the mentioned discriminative training methods are used in a speaker verification system in their original design, either all target models must be trained together with the background model(s), or one separate (set of) background model(s) must be trained together with each new target model. The first case poses a problem with enrolling new speakers into the system, since the background model and all other target models need to be re-trained for optimal performance (Liu et al., 1995). In the second case, determining a decision threshold may be an issue, and the speaker-dependent background model(s) adds to model storage requirements.

Two other discriminative training criteria were suggested by Navrátil and Ramaswamy (2002) and called DETAC by a common name. They are formulated to translate and rotate the DET curve (Martin et al., 1997) to optimize it for a given operating point by modifying score distributions through either a feature space transform (fDETAC) or at the score level in score-level fusion of multiple classifiers (pDETAC). Li et al. (2002) proposed a method called Figure of Merit (FOM) training that also operates on a specified region of the DET curve. They use a gradient descent scheme to adjust target model GMM parameters to optimize false accept and false reject rates, where the gradient is computed from a smoothed DET curve.

Model transformation techniques Two interesting and recently proposed model transformation techniques are reviewed here. The first technique aims at reducing between-session handset and channel mismatch in speaker models, while the purpose of the other is to increase modeling accuracy of Gaussian mixtures with diagonal covariance matrices.

Speaker Model Synthesis (SMS) (Teunen et al., 2000) is a speaker-independent transform of GMMs that aims at reducing channel mismatch between enrollment and test. Like in the Feature Mapping method (p. 17), a root GMM is trained on data from all available channels, and channel-dependent (background) GMMs are created through MAP adaptation of the root model. A transform can then be computed for each pair of corresponding mixture components from the (speaker-independent) GMM for channel A to that for channel B. These transforms can be applied to a target model trained on data from channel A to synthesize a model matching channel B, assuming that the target model has been created through MAP adaptation from the background model for channel A. This method requires channel detection to identify the most suitable channel GMM.

Maximum Likelihood Linear Transform (MLLT) (Chaudhari et al., 2003) makes use of a set of linear, mixture component-dependent feature-space transforms. The transform for a particular Gaussian mixture component is derived to minimize the loss of likelihood in the assumption of diagonal covariance in the mixture component. Once computed, the MLLT transform is applied to both the (original) GMM parameters and the (original) feature space, with the result that the model is evaluated in a feature space translated and scaled in different ways in different parts of the original feature space.

2.4.3 Score normalization methods

This section gives a short overview of some of the normalization methods proposed for use in speaker recognition. The purpose of normalization is usually to reduce mismatch between enrollment and test conditions, and thereby add robustness to various sorts of variability, such as channel, noise or text variability, or to make a target-independent threshold more efficient. Methods can be grouped according to in which part of an ASV system they operate, whether in the feature, model or score domain. Some of the methods operating in the feature domain were reviewed in the section on Feature extraction (p. 17), and two methods operating in the model domain were described above. We will continue here with methods operating in the score domain.

A collection of popular score normalization methods have in common that they perform distribution scaling by translating and scaling the log-likelihood score of a likelihood ratio detector to standardize its distribution according to some criterion. Their purpose is to compensate for some structural mismatch between for example target models or between enrollment and test data. Methods include Zero normalization (Z-norm) (Reynolds, 1997a), its handset-dependent extension H-norm

(Reynolds, 1997a), Test normalization (T-norm) (Auckenthaler et al., 2000; Navrátil and Ramaswamy, 2003), and Distance normalization (D-norm) (Ben et al., 2002).

The world model (Carey et al., 1991) and cohort model (Rosenberg et al., 1992) approaches to implementing the impostor class-conditional pdf in a likelihood ratio detector, introduced on p. 20, can also be viewed as score normalization methods.

For good overviews and systematizations of score normalization methods for text-independent speaker recognition, see (Auckenthaler et al., 2000; Mariethoz and Bengio, 2005). (Barras and Gauvain, 2003) compared several feature and score normalization methods (CMS, CMVN, Feature Warping, T-norm, Z-norm and the cohort method) on text-independent cellular data (NIST 2002 evaluation data) in a GMM-based system. The best performance was found for a combination of Feature Warping and T-norm.

While most of the mentioned score normalization methods were originally proposed for text-independent speaker verification, in principle, they can all be applied also in text-dependent verification.

World and cohort modeling are standard techniques in text-dependent systems using small vocabularies, for example vocabulary-dependent systems based on digits. In the case of unrestricted vocabularies such as when a client is free to choose his own passphrase, the problem of how to create good background models occurs. Ideally, background models should have perfect lexical coverage, i.e. be trained on the same phrase or at least the same vocabulary as target models. However, it is not realistic to collect speech data for training phrase-dependent or even word-dependent models for every selected passphrase in a large-scale system, so some other solution is needed. A number of solutions have been proposed, for example sub-word background models (Parthasarathy and Rosenberg, 1996), using a crude model trained on the target's enrollment data (Siohan et al., 1999), lexically biasing a text-independent background model towards the target model (Hébert and Peters, 2000), or synthesizing background models by selecting individual observation distributions from a set of pseudo-impostor models (Isobe and Takahashi, 1999). A similar problem occurs when applying T-norm in the text-dependent case (Hébert and Boies, 2005).

2.5 Performance assessment

To assess the performance of a speaker verification system, error rates need to be estimated. There are several issues related to error rate estimation. First, test trials are needed to provide data for our estimation. A suitable speaker verification corpus must be identified, or a field trial or corpus evaluation needs to be designed and implemented to collect test trials. Second, relevant estimates of error rate from test trial results must be computed. This involves formulating relevant quantities and estimating them. The output is a list of measures of what we like to call *technical error rate*. Third, the statistical uncertainty in estimated error rates should be quantified. Fourth, technical error rates need to be interpreted with respect to some particular use of the speaker verification system – the target application. The most critical element in this interpretation is to compare the *evaluation factors*, i.e. conditions under which our data were collected, to the conditions that will prevail in our target application. Are the two sets of conditions equivalent, or are there differences that may affect error rate? Conditions such as the amount and quality of enrollment and test data, channel and noise variability, inter- and intra-speaker variability, speaker group homogeneity, impostor dedication, and impostors' possibility of finding out secret passwords and account information may play a role here. See also (Oglesby, 1995; Doddington, 1998) for good discussions on these topics.

While error rate is usually the most important aspect of an ASV system, seen as a component technology, other aspects may also be of interest for an overall assessment of how "good" the system is. For example, it follows from the third issue listed above that error rate figures do not make much sense if taken out of context, and hence, evaluation factors like requirements on the amount and quality of enrollment and test data, channel and noise variability, etc. must be taken into account. Oglesby (1995) suggested the use of a performance profile, where a few key numbers are used to present the performance of an ASV system, including error rate, storage requirement for target models, speech quality, speech quantity and speaker density.

Mansfield and Wayman (2002) is a good "best practices" document for assessment of technical error rate in biometric systems in general.

When assessing an entire application where an ASV system is a component, issues such as user interface design and user psychology (Ashbourn and Savastano, 2002) are also important.

2.5.1 Performance measures

The most commonly used error measures for an ASV system in a real application are the average false reject rate (FRR) and the average false accept rate (FAR). These measures assume an *a priori* decision threshold and are conventionally estimated with Maximum Likelihood (ML) estimates

$$\begin{cases} \mathrm{FRR} = \dfrac{\text{number of false rejects}}{\text{number of true-speaker tests}} \\ \mathrm{FAR} = \dfrac{\text{number of false accepts}}{\text{number of impostor tests}}. \end{cases} \tag{2.11}$$

Other names for the false reject rate are *miss rate* or *Type-I error rate*, while corresponding names for the false accept rate are *false alarm rate* or *Type-II error rate*. The *Detection Cost Function* (DCF) (Martin and Przybocki, 2000) combines the FRR and FAR into a single number

$$\mathrm{DCF} = C_{\mathrm{FR}} \cdot P(\text{true speaker}) \cdot \mathrm{FRR} + C_{\mathrm{FA}} \cdot P(\text{impostor}) \cdot \mathrm{FAR} \tag{2.12}$$

where C_{FR} and C_{FA} are costs assigned to the two types of error and $P(\text{true speaker})$ and $P(\text{impostor})$ are *a priori* probabilities. The *Half Total Error Rate* (HTER) is a special case of the DCF

$$\mathrm{HTER} = \frac{1}{2}(\mathrm{FRR} + \mathrm{FAR}). \tag{2.13}$$

For cases where a decision threshold is not specified *a priori*, results are often presented in terms of a *Detection Error Trade-off* (DET) curve (Martin et al., 1997), where the trade-off between FRR and FAR for all possible (*a posteriori*) threshold values are drawn as a curve, or in terms of the *equal error rate* (EER). The EER is the error rate at an *a posteriori* threshold at which the FRR equals the FAR.

The purpose of the DET curve is to show FRR/FAR trade-off over a range of operating points (threshold selection criteria), to for example let application developers choose the appropriate operating point and the corresponding threshold suitable for their application. To a selected operating point on the DET curve corresponds a pair of FRR/FAR that could be taken as a prediction of ASV error rate in the application. However, the pair of FRR/FAR at a selected point is only a good estimate of actual application performance if evaluation factors affecting the choice of the decision threshold are really equivalent between a data set used to compute the DET curve and application use. If there is a systematical difference, there will be a threshold bias. The result is that the criterion for the selected operating point, for example an EER criterion or a targeted FAR, will no longer be met. One ASV system may be robust in the sense that the threshold bias is small, while other ASV systems may be very sensitive in this respect. To assess such robustness, separate data sets must be used: a development or validation set for selecting a threshold and a test set for testing the threshold. Bengio and Mariéthoz (2004a) suggested a procedure to implement this assessment and visualize the result, still for a range of possible operating points. A resulting curve is called *Expected Performance Curve* (EPC). An EPC shows what error rate is found on a test set, given that the decision threshold was selected according to some criterion and a validation data set.

2.5.2 Statistical significance

Performance figures estimated from an experiment are usually point estimates of some underlying “true” performance parameter. For example, assume we want to measure the overall FRR p of an ASV system. We let a test group make a number of true-speaker attempts and then estimate a value $\hat{p}$ for p as the fraction of attempts that resulted in a reject decision. $\hat{p}$ is then a *point estimate* of p on our observations. This point estimate may have a bias and a random error. A bias is a systematic error that may result for example from the test group not being a representative sample of the intended user population, while random errors are errors that result from chance. If the same experiment were to be repeated several times under identical conditions, the result would be different from time to time due to random errors. The purpose of an *interval estimate* is to specify a lower and an upper limit on the “true” value of the parameter given our experiment, thereby quantifying the random error inherent in the measurement. If the limits are defined such that they capture the “true” parameter value with a given probability, they are called confidence limits, and the interval between the limits is a *confidence interval.* Interval estimates in general, and confidence intervals in particular, can be useful in determining the statistical significance of results from an experiment, or in testing a hypothesis about the performance of an ASV system. In this section we discuss how to determine confidence intervals in ASV experiments.

To derive confidence intervals for error rate performance measures in ASV we first need a statistical model for how verification errors are generated. Assume false reject errors by an ASV system for a given target speaker are generated at random and without memory at a constant rate, i.e. assume each true-speaker test is a Bernoulli trial with a constant error probability p. Define X as the random variable for the number of errors observed in $N = n$ independent trials[4]. X is then binomially distributed with probability mass function

$$P_X(x|p) = \binom{N}{x} p^x (1-p)^{N-x} \tag{2.14}$$

with mean Np and variance $Np(1-p)$, while the observed error rate x/N has mean p and variance $p(1-p)/N$. A $1-\alpha$ confidence interval[5] for x is defined as an interval $[a_1, a_2]$ that satisfies[6]

$$\sum_{x=a_1}^{a_2} P_X(x|p) \geq 1-\alpha \tag{2.15}$$

and indicates an interval within which a measurement x falls with probability $1-\alpha$. In other words, if we make N independent true-speaker tests with a target speaker and an ASV system with the “true” false reject error rate p for this target speaker, we have at least a $1-\alpha$ chance that the observed number of errors will be between a_1 and a_2 inclusive, thus the observed error rate will be between a_1/N and a_2/N, inclusive. For example, assume $p = 0.04$ and $N = 300$. Then a 95% confidence interval ($\alpha = 0.05$) for the observed number of errors is $6 \leq x \leq 19$ and the corresponding interval for the observed error rate is $0.020 \leq x/N \leq 0.063$. Call this type of confidence interval a *pre-trial confidence interval.*

A confidence interval for the true error rate p given an observation of x in N independent trials can be defined similarly. Note that in this case, the confidence

[4] Later in this section we will take n to denote the number of trials from a single speaker, and N the total number of trials in a series of trials from multiple speakers.

[5] α is called the level of significance

[6] We formulate (2.15) as an inequality because the binomial distribution is discrete and it may not be possible to find values for a_1 and a_2 that give an interval with probability exactly $1-\alpha$.

limits are functions of the random variable X, $a_1 = a_1(X)$ and $a_2 = a_2(X)$, and the interval indicates a range that covers p with probability $1 - \alpha$. Call this type of confidence interval a *post-trial confidence interval.*

A pre-trial confidence interval can be useful for example when planning a corpus collection or a service trial since it can give an indication of the required number of samples. A problem is that before the samples have been collected we have little possibility to evaluate the assumptions our derivation of the significance interval were based on, and we run a risk of under-estimating the required sample size. Post-trial confidence intervals are used to present uncertainty in observation. Since we in this case have access to the observation data, chances are better for making a good interval estimate.

2.5.2.1 Dealing with invalid assumptions

To apply confidence intervals to real data, we are faced with the issue of judging whether the above assumptions about false reject errors being generated at random and without memory at a constant rate for a given target are valid. In particular, are trials independent ("without memory") and is the error rate constant?

The made assumptions are obviously not strictly valid. For example, mismatching input channels, background noise, learning effects and temporary voice changes from head colds are likely to vary the error rate over time as they occur. Recent experience of erroneous decisions may influence a claimant to alter his speaking, causing attempts to be not strictly without "memory". Thus, the model of true-speaker tests as a series of Bernoulli trials with constant error probability is not strictly valid, i.e. our model is flawed. Though some of the deviations from the model may result in smaller variance, the gross effect of all deviations should be an increased uncertainty in observations (increased variance in both X and X/N) and thus a wider confidence interval than that resulting from the binomial distribution. To incorporate the full effect of model mismatches and observation dependencies into a mathematical model is hardly realistic. However, a first-order approximation of an increased variance could be included in the calculations of the confidence interval by multiplying variances by a factor $k > 1$, for an adjusted error rate variance

$$\sigma_p^2 := \frac{p(1-p)k}{N}. \tag{2.16}$$

With a normal approximation of the resulting sample distribution, the multiplication of the variance by k results in a confidence interval $\sqrt{k}$ times wider. If we still want to use the binomial distribution, we can think of the modified variance under (2.16) as resulting from an assumed number

$$N' = N/k \tag{2.17}$$

of independent trials from the binomial model *corresponding* (in terms of variance in observed error rate) to the N partially dependent trials from the real world. N' would then be used in the calculation of the confidence interval together with a proportionally reduced number of observations $x' = x/k$. Resulting confidence limits on the number of observations will relate to x', while in the limits on error rate the scale factor will cancel out since $x'/N' = x/N$. With this simple approximation of the effect of partial dependence between observations and flaws in our error generation model, we assume observations x' are still binomially distributed. In our above example, assume the 300 trials are partially dependent and in a statistical sense correspond to say 100 independent trials. The 95% confidence interval for the observed error rate is then $0.010 \leq x/N \leq 0.080$ instead.

But how do we determine the equivalent number of independent trials, and is the binomial distribution still valid at all? For the determination of pre-trial confidence intervals, these are open questions. With post-trial intervals, on the other hand, one possibility may be to estimate the variance $s^2_{\hat{p}}$ in observed error rate using a resampling method[7], such as the bootstrap (e.g. Politis, 1998), and then compute a corresponding N' such that the variance from the binomial $\hat{p}(1-\hat{p})/N'$ equals the estimated variance, i.e.

$$N' = \frac{\hat{p}(1-\hat{p})}{\hat{s}^2_{\hat{p}}} \tag{2.18}$$

and subsequently

$$k = \frac{N}{N'} = \frac{N\hat{s}^2_{\hat{p}}}{\hat{p}(1-\hat{p})}. \tag{2.19}$$

This still assumes the binomial distribution is valid.

Confidence intervals were defined above for the task of estimating FRR for a single target speaker from n trials by this speaker. A more common task is to estimate an overall FRR for a population of target speakers. Assuming a single trial from each of M independent target speakers and a "true" fixed overall error rate p, confidence intervals could be computed like above from the binomial distribution. Here, practical issues are to judge if subjects are representative of the intended population, and if trials from subjects are independent. Again we have a problem with the model assumptions in that the error rate has been shown to depend on the target (Doddington et al., 1998), and thus it is not constant over trials.

The situation becomes more complex if we want to use confidence intervals with pooled true-speaker tests from several target speakers with multiple trials per speaker, say M targets with n trials per target for a total of $N = Mn$ trials[8]. Using (2.17) we get $N' = Mn/k$. Schuckers (2003b) suggests the beta-binomial distribution as a more appropriate model for this case than the binomial. With this model, error rate for a given target speaker is described as being drawn from a beta distribution, and the usual binomial distribution then describes the generation of observations from that speaker. According to Schuckers, this leads to a variance in

[7] statistical resampling refers to a variety of methods based on repeated sampling of already collected samples

[8] This is a *cluster sampling* technique, where each target corresponds to a sampling unit and we take multiple samples from each sampling unit (Snedecor and Cochran, 1967)

observed overall error rate the same as in (2.16) with $N = Mn$, and with k expressed in the model parameters (α, β and the number of observations per speaker n), or, even more interestingly, in (Schuckers et al., 2004) as[9]

$$k = 1 + (n - 1)\rho \tag{2.20}$$

where ρ is said to represent the *intra-speaker correlation* and n is the number of observations per speaker. Choices $k = 1$ ($\rho = 0$) and $k = n$ ($\rho = 1$) correspond to the two extremes in the assumption of partial dependence between observations. The first case assumes that all tests in a test set are independent, while the second case assumes that repeated true-speaker tests by a given target are dependent (always give the same result) and thus the number of independent tests equals the number of targets. The independence assumption in the first case is true in the sense that tests are based on distinct recordings, but it is false in the sense that recordings from a given target are dependent through the target. Basically, the problem is that tests are dependent at (at least) two different levels[10]: two tests from the same speaker are more dependent than tests from two different speakers. Thus, relevant confidence intervals probably result from a choice of k somewhere in between the two extreme cases, i.e. $1 < k < n$ ($0 < \rho < 1$).

For post-trial confidence intervals with n trials per target, the variance of the estimate $\hat{p}$ can be estimated from observed data. Snedecor and Cochran (1967); Mansfield and Wayman (2002) provide formulas based on observed individual error rates as

$$\hat{s}^2_{\hat{p}} = \frac{\sum_i (p_i - \hat{p})^2}{M(M-1)} \tag{2.21}$$

where p_i is the fraction of false reject errors (individual error rate) observed for target i. In the case of n_i trials per target, the variance estimate can be approximated as

$$\hat{s}^2_{\hat{p}} = \frac{\sum_i \left\{ \left(\frac{n_i}{\bar{n}}\right)^2 (p_i - \hat{p})^2 \right\}}{M(M-1)} \tag{2.22}$$

where $\bar{n}$ is the average number of attempts per target. Schuckers et al. (2004) provide other formulas to estimate $\hat{s}^2_{\hat{p}}$ based on the assumption of a beta-binomial model, but they are also based on individual error rates as input. (Schuckers et al. already drew the parallel between the methods and also claimed that the two methods, when used to estimate confidence intervals, produce similar results, at least for large M.)

Given an estimate of $\hat{s}^2_{\hat{p}}$, (2.18) can be used to determine an estimate of an equivalent (in terms of variance) total number of tests N' under the binomial. With $N = Mn$ (or $N = M\bar{n}$ in conjunction with (2.22)) a value for k can then be calculated from (2.19). Mansfield and Wayman (2002) proposed this approach

[9] for the case of an equal number of tests per speaker

[10] within and between sampling units

to computing post-trial confidence intervals, together with the use of a normal approximation of the binomial distribution.

In both the mentioned approaches, confidence intervals are computed from a normal approximation of the sample distribution, where the variance of the normal distribution is calculated from data under assumptions of binomial or beta-binomial distribution. However, the binomial is approximately normal only for large N and not too small p, but in the case of ASV p may be quite small. Hence, in some cases it may be better to compute confidence intervals from the binomial directly[11] using the N' calculated from the estimated variance. A drawback with using the binomial with $N' < N$ is that there are fewer values on x' than on x and confidence limits are therefore quantized by larger steps between values on x'. Furthermore, values x'/N' are "virtual" in that they don't (necessarily) correspond to observable error rates on x/N.

The approaches described above for determining confidence intervals are parametric since they are based on a binomial (or beta-binomial) model. Non-parametric approaches have also been proposed, based on resampling techniques. Bolle et al. (2004) compared three resampling (bootstrap) methods to the parametric binomial method in terms of coverage[12] on cluster sampled fingerprint data (multiple impressions from each person and index finger) and concluded that subset bootstrap methods performed better than the conventional "global" bootstrap and the binomial method. With the subset methods, the original cluster sampling strategy is maintained in resampling, and Bolle et al. claim the advantage of the subset methods is that they better capture dependencies within clusters (in this case persons or fingers). Schuckers et al. (2004) compared these and additional methods on simulated data from a correlated binary distribution, and found that in general, a parametric Logit beta-binomial method (Schuckers, 2003a) worked best in terms of coverage, except when the correlation between samples (ρ) is "large" and the product $M\hat{p}$ is "small", in which case a non-parametric resampling technique called Balance Repeated Replicates (Michaels and Boult, 2001) was better.

Dass and Jain (2005) proposed a semi-parametric method to compute confidence bands for DET curves.

See Dialogues Spotlight Consortium (2000, Appendix A) for further analysis of issues in applying theoretical confidence intervals to real data, and (Bengio and Mariéthoz, 2004b) for methods to compute confidence intervals for aggregate error rate measures such as DCF and HTER.

2.6 Corpora

To develop and evaluate practical ASV systems, example speech data are usually needed. For reproducible results, this speech data are preferably well documented and packaged into a *speech database*, or a *speech corpus*, by which we mean a finite

[11] or rather a better approximation of the binomial with corrections for skewness

[12] coverage indicates how often a confidence interval covered the "true" parameter value

collection of speech data in non-volatile storage. A speaker verification corpus is a speech corpus suitable for experiments in speaker verification. The main difference between a corpus suitable for speaker verification and one targeted for speech recognition is the need for intra-speaker variability coverage. The recordings should preferably be spread over time to capture both long-time changes and colds, sore throats, mood, and other sources of short-time variation in speakers. Alternatively, or in addition, other variations may be covered depending on the focus of research, e.g. handset variation (Reynolds, 1997b).

Depending on the purpose of ASV experiments, there may exist a suitable public corpus, or a dedicated corpus need to be collected. In general, the closer to application deployment in a development process, the smaller chance there exists a suitable corpus. Hence, generic research in algorithmic aspects of ASV can usually be done with existing, more or less generic, corpora, while tuning of an ASV system before application deployment must usually be done with bootstrap data collected from (an initial version of) the application itself.

Many speaker verification corpora exist, covering many languages. Some of the corpora are publicly available through the major corpus distribution agencies LDC[13] and ELRA[14], or from elsewhere. See Melin (1999); Campbell and Reynolds (1999); Ortega-Garcia and Bousono-Crespo (2005) for overviews of existing corpora.

[13] http://www.ldc.upenn.edu/

[14] http://www.elra.info/

Part II

Systems and applications

Chapter 3

The speaker verification systems

3.1 Introduction

This chapter describes the speaker verification research systems used in this thesis: a text-dependent (sub-)system based on word-level hidden Markov models (HMM), a text-independent (sub-)system based on Gaussian mixture models (GMM), and a score level combination of the two. The HMM system, and variants of it, are used stand-alone in Chapter 8 that compares different prompting strategies, and Chapter 9 that looks at variance estimation techniques for HMMs. The combined system is used as a component in the PER system and is used for experiments in Chapter 7 on robust error estimation techniques, and Chapter 10 with various results from the PER system and data collected with it.

All speaker verification research systems used in this thesis are built on GIVES, a generic framework for speaker verification systems developed by the author at KTH Center for Speech Technology (CTT). This framework is shortly described.

In addition to the research systems described in this chapter, a commercial speaker verification system has also been used. Results for this system are included in Chapters 7 and 10 in addition to results from the research systems. For reasons of proprietary interests the design of this commercial system cannot be described here, nor may the identity of the system be disclosed.

3.2 Notation and common features

The HMM and GMM subsystems share several features. These features are described in this section together with some notation used later. In the following, the letter ξ will be used to refer to a subsystem, with $\xi = \mathrm{H}$ for the HMM subsystem and $\xi = \mathrm{G}$ for the GMM subsystem.

3.2.1 Feature extraction

The input signal[1] is pre-emphasized and divided into one 25.6 ms frame every 10 ms. A Hamming window is applied to each frame. 12-element[2] mel-frequency cepstral coefficient (MFCC) vectors are then computed for each frame using a 24-channel, FFT-based, mel-warped, log-amplitude filter bank between 300-3400 Hz followed by a cosine transform and cepstral liftering. Both subsystems use these MFCC vectors, while their use of energy terms, delta features and feature post-processing differ, mainly as a result of the subsystems having been optimized rather independently during separate threads of development (see also Section 10.2.1).

3.2.2 Classifier units

Classifiers in both subsystems share a basic classifier unit structure. Refer to a classifier unit in subsystem ξ as $\psi = \xi_u$, where u is an index that uniquely identifies the classifier unit within the subsystem. This classifier unit has one target model and two gender-dependent background models. Target models represent the voices of particular speakers (legitimate users of the system, or *clients*), while background models represent the voices of universal groups of speakers, in this case male and female speakers. Background models are used for two purposes: as seed models during the training phase, and for score normalization during the verification test phase.

Each classifier unit ψ defines one or more likelihood functions $P^{\psi}(\mathbf{O}|\boldsymbol{\lambda})$ used to evaluate the similarity between an observation sequence $\mathbf{O}$ and the model $\boldsymbol{\lambda}$. In the following, $\boldsymbol{\lambda}_{\psi}$ will denote parameters of the target model for a particular target speaker (the client whose identity is claimed during an enrollment or test session) while $\boldsymbol{\lambda}_{\psi}^{\text{male}}$ and $\boldsymbol{\lambda}_{\psi}^{\text{female}}$ will denote parameters of the two background models in classifier unit ψ.

The data and operation of classifier units within the system are independent of each other during both the training and verification test phases: units share no model parameters and the data processing within one unit takes no input from the processing in other units. Units may operate on the same part of input speech, though.

3.2.2.1 Training phase

Assume that all relevant word repetitions and their boundary locations in the enrollment speech are known from the output of an automatic speech recognizer[3].

[1] at 8 kHz sampling rate (see Section 5.2 (p. 79) for particulars about the on-site PER system)

[2] does not include the 0'th cepstral coefficient

[3] cf. Section 5.6 (p. 85) for the procedure used in the PER system

Denote all valid[4] enrollment data from a given enrollee

$$\overline{\mathbf{O}}^{\text{enroll}} = \bigcup_{w \in \mathbf{W}} \{\mathbf{O}_{w,1}, \ldots, \mathbf{O}_{w,R_w}\}$$

where w is a word in the application vocabulary $\mathbf{W}$[5], $\mathbf{O}_{w,r} = \{\mathbf{o}_1^{(w,r)} \ldots \mathbf{o}_{N_{w,r}}^{(w,r)}\}$ is the observation sequence corresponding to the r'th valid repetition of word w (a word segment), $N_{w,r}$ is the length of that observation sequence, and R_w is the number of valid repetitions of word w.

Since classifier units may be trained on different subsets of the data, introduce $\overline{\mathbf{O}}_{\psi}^{\text{enroll}}$ to denote the subset of $\overline{\mathbf{O}}^{\text{enroll}}$ used to train unit ψ. Rather than training a target model directly from this data, *an adaptation procedure is used.* While the actual adaptation method depends on the implementation of the classifier unit, the first step in the adaptation procedure is the same for all classifier units. Based on the enrollment data, one of the two background models is selected as a seed model $\boldsymbol{\lambda}_{\psi}^{g_{\psi}^{\text{seed}}}$, using an automatic gender detector

$$g_{\psi}^{\text{seed}} = \underset{g \in \{\text{male,female}\}}{\arg\max} \; P^{\psi}(\overline{\mathbf{O}}_{\psi}^{\text{enroll}} | \boldsymbol{\lambda}_{\psi}^{g}). \tag{3.1}$$

That is, if the male model fits better to the data, the male model is chosen, otherwise the female model is chosen. Note that no *a priori* information about the gender of the enrollee is used in this selection, and that gender selection in one classifier unit is independent of other classifier units in the system.

The seed model is then used as a basis for target model adaptation as described for each of the two subsystems below.

3.2.2.2 Verification test phase

To test a claim for a given target identity put forward by a claimant speaker, a test utterance is first collected. Again assuming all relevant word repetitions and their boundary locations are known from the output of an automatic speech recognizer, a test utterance with L words is denoted $\overline{\mathbf{O}}^{\text{test}} = \{\mathbf{O}_1 \ldots \mathbf{O}_L\}$, where $\mathbf{O}_i = \{\mathbf{o}_1^{(i)} \ldots \mathbf{o}_{N_i}^{(i)}\}$ is the vector sequence corresponding to the i'th word segment in the utterance. Denote as $w(i)$ the word spoken in segment i. The exact function used by classifier units to score a test utterance given an identity claim varies between units, but it always has the form

$$z_{\psi} = \mathcal{F}\left(\overline{\mathbf{O}}^{\text{test}} | \boldsymbol{\lambda}_{\psi}, g_{\psi}(\overline{\mathbf{O}}^{\text{test}})\right), \tag{3.2}$$

[4] assuming the application somehow checks collected utterances for validity; see for example Section 5.5 (p. 84) for the procedure used in the PER system

[5] for example $\mathbf{W} = \{0, \ldots, 9, \text{name}\}$ as in the PER system

where $\boldsymbol{\lambda}_\psi$ is the model created for the target identity from the target's enrollment data, and

$$g_\psi(\overline{\mathbf{O}}^{\text{test}}) = \operatorname*{arg\,max}_{g \in \{\text{male,female}\}} P^\psi(\overline{\mathbf{O}}^{\text{test}} | \boldsymbol{\lambda}_\psi^g) \tag{3.3}$$

is a gender detector like the one used in the training phase, but it uses test data instead of enrollment data to make the gender selection.

This method of selecting a background model has been referred to as an unconstrained cohort by other authors (Ariyaeeinia and Sivakumaran, 1997). It differs from the traditional cohort method (Higgins et al., 1991; Rosenberg et al., 1992) in that the selection is based on similarity to a test segment rather than to enrollment data. However, our method differs slightly from both the traditional cohort method and the unconstrained cohort method in that the competing models are only two and represent groups of speakers (genders) rather than individual speakers.

3.3 The text-dependent HMM system

The HMM subsystem is text-dependent and operates in a prompted mode with digit string utterances only[6]. Except for how background models are selected during the test phase, the system is the same as the baseline system described and tested in (Melin et al., 1998) and (Melin and Lindberg, 1999b). In this section, the design of the HMM subsystem is described relative to the common subsystem features described in Section 3.2. The modified background model selection method is described and evaluated.

3.3.1 Feature extraction

The basic 12-element MFCC vector (Section 3.2.1) is extended with the 0'th cepstral coefficient (frame energy). Cepstral mean subtraction is applied to this 13-element static feature vector, and first and second order deltas are appended. The total vector dimension is 39.

3.3.2 Classifier units

The HMM subsystem contains ten classifier units $\psi = \mathrm{H}_0 \ldots \mathrm{H}_9$, one classifier unit per digit word. Models are continuous word-level left-to-right HMMs with 16 Gaussian terms per phoneme in the represented word distributed on two states per phoneme[7] with an eight-component Gaussian mixture observation probability density function (pdf) per state. Gaussian components have diagonal covariance matrices. The choice of 16 terms per phoneme is based on development experiments on Gandalf data in preparation for previous work (Melin et al., 1998), while their partitioning into two states with eight terms each is somewhat arbitrary as shown

[6] it ignores the name parts of enrollment and test data in the PER case

[7] Swedish digit words have between two and four phonemes per word.

by Bimbot et al. (2000). Denote the target HMM in classifier unit H_w for a given client as $\boldsymbol{\lambda}_{\mathrm{H}_w} = \{\mathbf{c}_w, \mathbf{m}_w, \boldsymbol{\sigma}_w^2, \mathbf{A}_w\}$ where $\mathbf{c}_w$, $\mathbf{m}_w$ and $\boldsymbol{\sigma}_w^2$ are vectors of all mixture weights, mean values and variance values, respectively, and $\mathbf{A}_w$ is the matrix of transition probabilities.

3.3.2.1 Training phase

A target model $\boldsymbol{\lambda}_{\mathrm{H}_w}$ for the word w and a given client is trained on all R_w valid examples of the word spoken during the client's enrollment session[8]. That is, training data $\overline{\mathbf{O}}_{\mathrm{H}_w}^{\mathrm{enroll}}$ for classifier unit H_w is a subset of $\overline{\mathbf{O}}^{\mathrm{enroll}}$ such that $\overline{\mathbf{O}}_{\mathrm{H}_w}^{\mathrm{enroll}} = \{\mathbf{O}_{w,1}, \ldots, \mathbf{O}_{w,R_w}\}$, where observations are 39-dimensional feature vectors as described in the previous section.

Given the training data, one of the gender-dependent background models is first selected as a seed model using a gender detector (Eq. 3.1). The seed model is then used as a basis for target model training: transition probabilities and variance vectors are left as they are, while mean vectors and mixture weights are trained from the data. Training is performed with the Expectation Maximization (EM) algorithm to optimize the Maximum Likelihood (ML) criterion

$$(\hat{\mathbf{c}}_w, \hat{\mathbf{m}}_w) = \underset{(\mathbf{c}_w, \mathbf{m}_w)}{\arg\max} P(\overline{\mathbf{O}}_{\mathrm{H}_w}^{\mathrm{enroll}} | \mathbf{c}_w, \mathbf{m}_w, {\boldsymbol{\sigma}_w^{\mathrm{seed}}}^2, \mathbf{A}_w^{\mathrm{seed}}), \tag{3.4}$$

where ${\boldsymbol{\sigma}_w^{\mathrm{seed}}}^2$ and $\mathbf{A}_w^{\mathrm{seed}}$ are the fixed variance and transition probabilities taken verbatim from the seed model $\boldsymbol{\lambda}_{\mathrm{H}_w}^{g_{\mathrm{H}_w}^{\mathrm{seed}}}$. The seed means and mixture weights are used as starting values in the first iteration of the EM algorithm (Rosenberg et al., 1991).

Background models were trained with the EM-algorithm and the ML criterion. After initializing models with a single Gaussian per state, Gaussians were split into $2 \rightarrow 4 \rightarrow 6 \rightarrow 8$ Gaussians per state and re-estimated with up to 20 EM iterations after each splitting operation. A fixed variance floor of 0.01 was used, but only 0.1% of all variance parameters received a value less than twice the floor.

3.3.2.2 Verification test phase

The likelihood function implemented by the classifier unit (during the verification test phase) is the Viterbi approximation of the probability of observation data given a model, i.e. the probability of observations given the model and the most likely path:

$$P^{\mathrm{H}_w}(\mathbf{O}|\boldsymbol{\lambda}) = \max_{\mathbf{S} \in \boldsymbol{\Omega}} P(\mathbf{O}|\boldsymbol{\lambda}, \mathbf{S}) \tag{3.5}$$

where $\mathbf{S}$ is a certain path through the HMM $\boldsymbol{\lambda}$ and $\boldsymbol{\Omega}$ is the set of all possible paths. The notation $P^{\mathrm{H}_w}(\mathbf{O}|\boldsymbol{\lambda})$ is used to indicate this is the likelihood function used in classifier unit $\psi = \mathrm{H}_w$ (cf. Section 3.2.2).

[8] in the PER-system, $R_w = 5$ (for digits)

Given a target model and a test utterance, a classifier unit produces an output score value s_{H_w} for each word segment i for which $w(i) = w$:

$$s_{\mathrm{H}_w}(i) = \frac{1}{N_i}\left(\log P^{\mathrm{H}_w}(\mathbf{O}_i|\boldsymbol{\lambda}_{\mathrm{H}_w}) - \log P^{\mathrm{H}_w}(\mathbf{O}_i|\boldsymbol{\lambda}^g_{\mathrm{H}_w})\right) \tag{3.6}$$

and for the entire test utterance

$$z_{\mathrm{H}_w} = \begin{cases} \frac{1}{L_{\mathrm{H}_w}} \sum_{i:w(i)=w} s_{\mathrm{H}_w}(i), & L_{\mathrm{H}_w} > 0 \\ 0, & L_{\mathrm{H}_w} = 0 \end{cases} \tag{3.7}$$

where

$$g = g_{\mathrm{H}_w}\left(\overline{\mathbf{O}}^{\mathrm{test}}_{\mathrm{H}_w}\right)$$

is the gender detected for the test utterance in the same classifier unit, Eq. (3.3), and N_i is the number of observation vectors in word segment i. $\overline{\mathbf{O}}^{\mathrm{test}}_{\mathrm{H}_w} = \{\mathbf{O}_i : w(i) = w\}$ is the subset of the test utterance where word w is spoken, and L_{H_w} the number of word segments in this subset (i.e. the number of repetitions of word w).

The score output value z_{H} from the entire HMM subsystem for a test utterance $\overline{\mathbf{O}}^{\mathrm{test}}_{\mathrm{H}} = \big\{\mathbf{O}_i : w(i) \in \{0\ldots 9\}\big\}$ (the subset of $\overline{\mathbf{O}}^{\mathrm{test}}$ where a digit word is spoken) is

$$z_{\mathrm{H}} = \frac{1}{L_{\mathrm{H}}}\sum_{u=0}^{9} L_{\mathrm{H}_u} z_{\mathrm{H}_u} = \frac{1}{L_{\mathrm{H}}} \sum_{i:w(i)=\{0\ldots 9\}} s_{\mathrm{H}_w}(i), \tag{3.8}$$

where L_{H} is the number of word segments in $\overline{\mathbf{O}}^{\mathrm{test}}_{\mathrm{H}}$.

3.3.3 Background model selection

The background model selection method in this system is different from the one used in our previous publications (Melin et al., 1998) and (Melin and Lindberg, 1999b), where the background model was chosen based on similarity to enrollment data like in the traditional cohort method. The purpose of selecting a background model based on similarity to the test segment is to circumvent a well-known problem with traditional cohorts and dissimilar impostors. If the background model is trained on data “close” to the target speaker, then both the target model and the background model will be poor models in regions of the sample space “far away” from the target speaker. Hence, the likelihood ratio test will not be a good test for dissimilar speakers, such as cross-sex impostors. By selecting the background model that is closer to the test segment, the likelihood ratio test is more likely to reject a dissimilar impostor. The advantage of the used method is evident from Figure 3.1, where same-sex (on the left) and cross-sex (on the right) DET curves are shown for both methods. These curves are from experiments on the Gandalf corpus with identical enrollment and test sets as were used in (Melin and Lindberg, 1999b). Results show that the unconstrained cohort method reduces cross-sex imposture rate considerably, at no loss in same-sex imposture rate.

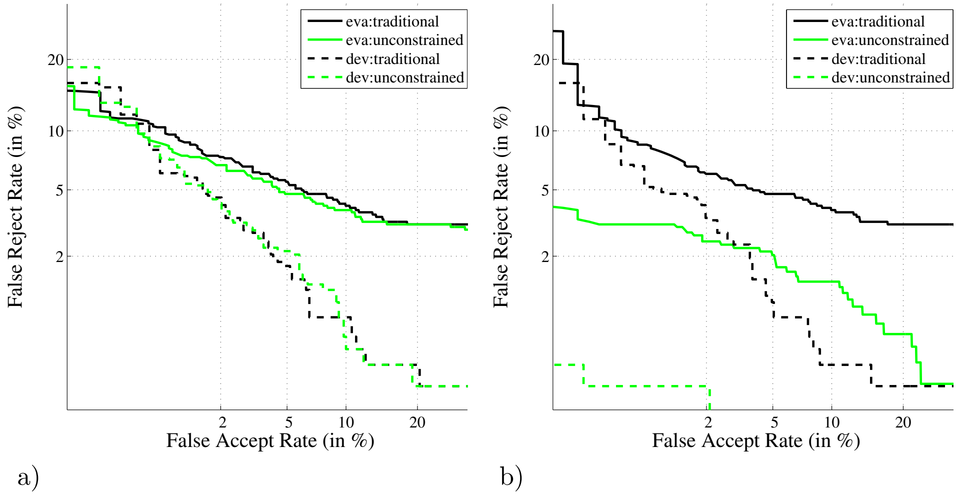


Figure 3.1: DET plots for the HMM subsystem with two different methods for selecting from one of two gender-dependent background models: by similarity to enrollment data (traditional cohort) or by similarity to test data (unconstrained cohort). Test data is from the Gandalf corpus with single-session, one-minute enrollment and two four-digit test utterances. DET curves are shown for both the development (dev) and the evaluation (eva) sets. Curves in a) are based on same-sex impostor attempts, while curves in b) are based on cross-sex impostor attempts. True-speaker tests are the same in a) and b).

3.4 The text-independent GMM system

The GMM subsystem is inherently text-independent, though in this thesis it is used in a prompted, text-dependent way in the sense that enrollment and test utterances are always composed of words from the same vocabulary[9]. Background models are still used text-independently, however. The GMM-specific modules for the GIVES framework were initially developed as part of a student project (Neiberg, 2001), and then extended by Neiberg in conjunction with CTT's participation in the 1-speaker detection cellular task in the NIST 2002 Speaker Recognition Evaluation (NIST, 2002). Experiments on a PER development set of Gandalf data (Section 10.2.1, p. 193) were then used as the basis for selecting the particular configuration of the GMM subsystem used in this work. This section describes the design and configuration of the subsystem in detail. It is included for completeness since the GMM system is used in the thesis and because not all parts of the description were published elsewhere.

[9] proper name and digits in the PER case

3.4.1 Feature extraction

The basic 12-element MFCC vectors (Section 3.2.1) are RASTA-filtered (Hermansky et al., 1991) and first order deltas are appended. The total vector dimension is 24.

3.4.2 Classifier unit

The GMM subsystem contains a single classifier unit $\psi = \mathrm{G}_0$, where target and background models are 512-component Gaussian mixture pdfs with diagonal covariance matrices, also known as GMMs (Rose and Reynolds, 1990; Reynolds, 1995). Denote the parameters of the target GMM in the classifier unit as $\boldsymbol{\lambda}_{\mathrm{G}_0} = \{c_k, \mathbf{m}_k, \boldsymbol{\sigma}_k^2\}_{k=1}^K$, where c_k is the weight and $\mathbf{m}_k$ and $\boldsymbol{\sigma}_k^2$ the vectors of mean and variance values of mixture term k, and $K = 512$ is the number of terms[10] in the model

$$p(\mathbf{o}|\boldsymbol{\lambda}_{\mathrm{G}_0}) = \sum_{k=1}^{K} c_k \phi\Big(\mathbf{o}|\mathbf{m}_k, \boldsymbol{\sigma}_k^2\Big). \tag{3.9}$$

$\phi()$ denotes the multivariate normal density function.

3.4.2.1 Training phase

A target model $\boldsymbol{\lambda}_{\mathrm{G}_0}$ for a given client is trained on all valid enrollment data from the client, i.e. $\overline{\mathbf{O}}_{\mathrm{G}_0}^{\mathrm{enroll}} = \overline{\mathbf{O}}^{\mathrm{enroll}}$ with observation vectors being 24-dimensional feature vectors as described above. Note that observation vectors from non-speech segments are not included in training data (provided word boundaries are correctly estimated).

Given the training data, one of the gender-dependent background models is first selected as a seed model using a gender detector (Eq. 3.1). The target model is then created from the seed model using the following maximum a posteriori (MAP) like update formulas (Reynolds et al., 2000):

$$\begin{aligned} c_k &= \big(\alpha_k \eta_k / N + (1-\alpha_k) c_k^g\big)\gamma && (3.10)\\ \mathbf{m}_k &= \alpha_k E_k(\mathbf{o}) + (1-\alpha_k)\mathbf{m}_k^g && (3.11)\\ \boldsymbol{\sigma}_k^2 &= \alpha_k E_k(\mathbf{o}^2) + (1-\alpha_k)\big({\boldsymbol{\sigma}_k^g}^2 + {\mathbf{m}_k^g}^2\big) - \mathbf{m}_k^2 && (3.12) \end{aligned}$$

where

$$\alpha_k = \frac{\eta_k}{\eta_k + r} \tag{3.13}$$

is a data-dependent adaptation coefficient with relevance factor $r = 16$,

$$\gamma = \frac{1}{\sum_{k=1}^K c_k} \tag{3.14}$$

[10] "term" and "component" are used interchangeably in this thesis when referring to the terms of the sum in (3.9)

assures target model weights sum to unity, and[11]

$$\eta_k = \sum_{n=1}^{N} \eta_{k,n} \tag{3.15}$$

$$E_k(\mathbf{o}) = \frac{1}{\eta_k} \sum_{n=1}^{N} \eta_{k,n} \mathbf{o}_n \tag{3.16}$$

$$E_k(\mathbf{o}^2) = \frac{1}{\eta_k} \sum_{n=1}^{N} \eta_{k,n} \mathbf{o}_n^2, \tag{3.17}$$

where

$$\eta_{k,n} = \frac{c_k \phi(\mathbf{o}_n | \mathbf{m}_k^g, \boldsymbol{\sigma}_k^{g\,2})}{\sum_{l=1}^{K} c_l \phi(\mathbf{o}_n | \mathbf{m}_l^g, \boldsymbol{\sigma}_l^{g\,2})} \tag{3.18}$$

is the *a posteriori* weight of mixture term k given an observation vector $\mathbf{o}_n$ and the seed model. Training data $\overline{\mathbf{O}}_{\mathrm{G}_0}^{\mathrm{enroll}}$ have here been viewed as a single vector sequence $\mathbf{O} = \{\mathbf{o}_1 \ldots \mathbf{o}_N\}$ with

$$N = \sum_{w \in \mathbf{W}} \sum_{r=1}^{R_w} N_{w,r} \tag{3.19}$$

where $N_{w,r}$ is the length of the observation sequence from the r'th valid repetition of word w, and $\mathbf{W}$ is the vocabulary (cf. Section 3.2.2.1).

Background models were trained with the EM-algorithm and the ML criterion. First a gender-independent "root" GMM was initialized from a VQ codebook and then trained on pooled male and female data with eight EM iterations. Centroids of the VQ codebook were initialized from 512 equidistant (in time) training vectors and then trained with the generalized Lloyd algorithm (e.g. Gersho and Gray, 1992) using the Mahanalobis distance measure. The root GMM was then used as the starting point for training a male GMM on male data and a female GMM on female data with three iterations for each gender model.

3.4.2.2 Verification test phase

The classifier unit is tested on all available speech segments in a test utterance, i.e. $\overline{\mathbf{O}}_{\mathrm{G}_0}^{\mathrm{test}} = \overline{\mathbf{O}}^{\mathrm{test}}$. Given a target model and a test utterance, a classifier unit produces an output score value

$$z_{\mathrm{G}_0} = \frac{1}{N_{\mathrm{G}_0}^{\mathrm{test}}} \left(\log P^{\mathrm{G}_0}\left(\overline{\mathbf{O}}_{\mathrm{G}_0}^{\mathrm{test}} | \boldsymbol{\lambda}_{\mathrm{G}_0}\right) - \log P^{\mathrm{G}_0,\mathrm{gps}}\left(\overline{\mathbf{O}}_{\mathrm{G}_0}^{\mathrm{test}} | \boldsymbol{\lambda}_{\mathrm{G}_0}^{g}\right)\right) \tag{3.20}$$

where $N_{\mathrm{G}_0}^{\mathrm{test}}$ is the number of observation vectors in the test utterance and

$$g = g_{\mathrm{G}_0}\left(\overline{\mathbf{O}}_{\mathrm{G}_0}^{\mathrm{test}}\right)$$

[11] $\mathbf{o}^2$ is a shorthand for $\mathrm{diag}(\mathbf{o}\mathbf{o}^{\mathrm{T}})$ (from Reynolds et al. (2000))

is the gender detected for the test utterance in the same classifier unit, Eq. (3.3). z_{G_0} is also used as the output score value of the GMM subsystem, i.e. $z_{\mathrm{G}} = z_{\mathrm{G}_0}$.

The likelihood function $P^{\mathrm{G}_0}\left(\overline{\mathbf{O}}^{\mathrm{test}}_{\mathrm{G}_0}|\boldsymbol{\lambda}_{\mathrm{G}_0}\right)$ used with the target model is the probability of test data given the model, i.e.

$$\log P^{\mathrm{G}_0}\left(\overline{\mathbf{O}}^{\mathrm{test}}_{\mathrm{G}_0}|\boldsymbol{\lambda}_{\mathrm{G}_0}\right) = \sum_{i=1}^{L}\sum_{n=1}^{N_i} \log\left(\sum_{k=1}^{K} c_k \phi\left(\mathbf{o}_n^{(i)}|\mathbf{m}_k, \boldsymbol{\sigma}_k^2\right)\right) \tag{3.21}$$

where $\mathbf{o}_n^{(i)}$ is an observation vector in the i'th word segment $\mathbf{O}_i$ in the test utterance (cf. Section 3.2.2.2).

A modified likelihood function $P^{\mathrm{G}_0,\mathrm{gps}}\left(\overline{\mathbf{O}}^{\mathrm{test}}_{\mathrm{G}_0}|\boldsymbol{\lambda}_{\mathrm{G}_0}\right)$ is used with background models in (3.20). It uses a Gaussian pre-selection (gps) method to reduce the number of calculations relative to (3.21). Each time (3.21) is evaluated for an observation vector in segment $\mathbf{O}_i$, the index k of the $C = 6$ top contributing mixture terms for that observation vector is stored into an N by C matrix $\boldsymbol{\kappa}^{(i)}$, and the likelihood for a background model is calculated as

$$\log P^{\mathrm{G}_0,\mathrm{gps}}\left(\overline{\mathbf{O}}^{\mathrm{test}}_{\mathrm{G}_0}|\boldsymbol{\lambda}^g_{\mathrm{G}_0}\right) = \sum_{i=1}^{L}\sum_{n=1}^{N_i} \log\left(\sum_{j=1}^{C} c^g_{\kappa^{(i)}(n,j)} \phi\left(\mathbf{o}_n^{(i)}|\mathbf{m}^g_{\kappa^{(i)}(n,j)}, {\boldsymbol{\sigma}^g_{\kappa^{(i)}(n,j)}}^2\right)\right) \tag{3.22}$$

where

$$\boldsymbol{\lambda}^g_{\mathrm{G}_0} = \left\{c^g_k, \mathbf{m}^g_k, {\sigma^g_k}^2\right\}_{k=1}^{K}$$

are the parameters of the background model for gender g.

This gps-method is a modified version of a method suggested by Reynolds (1997a) based on the assumption that a mixture term of an adapted GMM has a relation to the corresponding term in the GMM it was adapted from (the *parent* model), such that the two terms are "close" compared to other terms. Call this a parent relation. While Reynolds evaluated all mixture terms of a (single) background model and only selected terms in the target model, we used a variant where all terms of the target model are evaluated and only selected terms of the two background models. A similar variant was previously tested with a single background model by Navrátil et al. (2001), who showed that the modification results in a clock-wise rotation of the DET curve relative to the original method, i.e. reduced false accept rates at low false reject rates.

With our use of two background models in gender-detection during the test phase (Eq. 3.3), evaluating all terms in the target model and only a few in background models is a logical choice, since more computations are saved compared to fully evaluating both background models. To allow this, we create both background models through the adaptation of a common (gender-independent) "root" model as described above. Analogous to the mentioned parent relation, such background models have a *sibling relation*. We assume that siblings have a similar kind of closeness relation as parent-child, though weaker in strength. The sibling relation between background models is needed to use indexes of top-scoring mixture terms in a target model to pick mixture terms for Eq. (3.22) with both background models, because the target model has a parent relation to (was adapted from) only one of the background models (Eq. 3.1).

With our use of Gaussian pre-selection, the number of evaluated mixture terms is $K + 2C$ per observation vector compared to $3K$ for a full evaluation of target model and both background models, a reduction of 66%.

3.5 The combined system

3.5.1 Score fusion

The HMM and GMM subsystems are fused at the score level. The system output score value, or decision variable, z for a test utterance is a linear combination of subsystem score values

$$z = \omega_{\mathrm{H}} z_{\mathrm{H}} + \omega_{\mathrm{G}} z_{\mathrm{G}}. \tag{3.23}$$

Combination weights ω_ξ are computed as

$$\omega_\xi = \frac{1}{\sum_{\zeta \in \{\mathrm{H},\mathrm{G}\}} (1 - \epsilon_\zeta)/\sigma_\zeta} \cdot \frac{1 - \epsilon_\xi}{\sigma_\xi} \tag{3.24}$$

where ϵ_ξ and σ_ξ are determined empirically through a development experiment [12] with the individual subsystems, as their respective equal error rate and standard deviation of observed values for z_ξ. The rationale for (3.24) is that scores from each of the subsystems are first scaled to have unit variance (on development data) and are then weighted such that the subsystem with lower EER gets a higher weight.

3.5.2 Classification

The actual classifier decision is taken by comparing the value of the decision variable z to a speaker-independent threshold θ:

$$z \quad \underset{\text{reject}}{\overset{\text{accept}}{\underset{\leq}{>}}} \quad \theta. \tag{3.25}$$

The value of the threshold is also determined empirically from a development experiment[12].

[12] cf. Section 10.2, p. 192 for the PER system

3.6 The GIVES framework

GIVES (General Identity VErification System) is a C++ software package built for research in automatic speaker verification. It has a general purpose kernel which can be used both in simulations on speech corpora and in real-time demonstrators. Tools for running off-line simulations on speech corpora are included in the package, together with a server interface using SVAPI (cf. Section 4.2.2.5, p. 65).

The central part of GIVES is a platform with the aim of implementing as much as possible of abstract functionality in a speaker verification system, for example:

- how pieces of the system interact
- how speech data are input to the system
- how multiple speaker models are trained or evaluated in parallel
- generic support for implementing different kinds of building blocks (modules).

Actual methods, such as an MFCC feature extractor or an HMM classifier, are implemented as *modules* which can be put on the platform at runtime. Modules can then easily be tested, combined and compared. Figure 3.2 illustrates the composition of the GIVES framework.

The *speaker model* is central to the construction of a speaker verification system in GIVES. Six types of speaker model-related modules have been defined: stream operator, speaker model element, score operator, normalization operator, decision maker and threshold setter. By defining a speaker model configuration using implementations of these module types, the entire ASV system is defined from an algorithmic point of view. The configuration is specified in a *speaker model template file* which is fed to GIVES tools to train and test speaker models on speech data. Once a speaker model has been created for a particular target speaker, the configuration of the ASV system is stored together with data. This solution is ineffective in terms of storage requirements, but allows for flexible design of ASV systems. For example, an ASV system configuration can be modified by editing an already trained model, and different system configurations could be used for different targets.

Each of the speaker model-related module types will now be shortly introduced. Their typical use will then be exemplified through the description of how the above speaker verification systems were constructed using the GIVES framework.

- **stream operators** are organized into a *parameterization tree* to compute one or more speech feature streams. Speech waveform data, possibly together with associated external segmentation information, are fed through the root of the tree and each unique path of connected stream operators define a *speech feature type* that can be read by a speaker model through a leaf in the tree. Data passed between operators in the tree may be scalar values, vectors or segment labels. Data is computed on-demand in the sense that

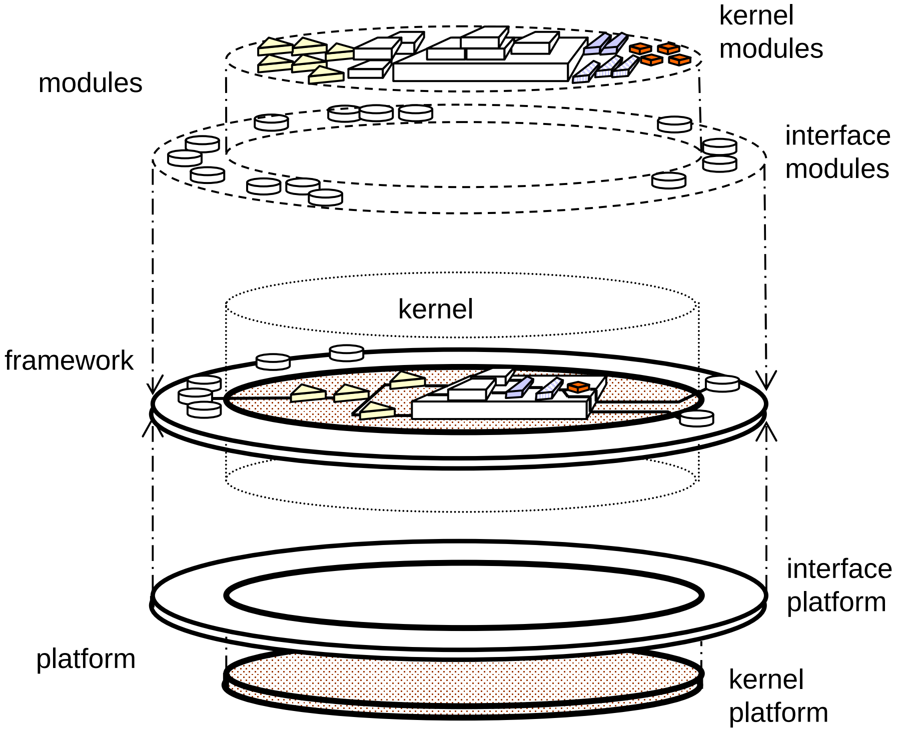


Figure 3.2: Exploded view of the conceptual composition of the GIVES framework. The platform is fixed in the sense that an ASV system designer cannot change it at runtime. The kernel part of the platform is always used, while the interface part is different with off-line simulation tools and with the server interface. An ASV system is constructed at runtime by connecting a selection of kernel module instances on the platform. Interface modules are selected to configure a database tool or server towards input and output file formats, etc.

only feature data requested by a speaker model element are computed by the parameterization tree. The tree itself is also *grown* on-demand, given the list of speech feature types specified by the set of loaded speaker models.

- **speaker model elements** are organized into hierarchical speaker models. A speaker model may contain one or more subordinate elements, each of those elements in turn may contain zero or more subordinate elements, etc. A speaker model element both contains the actual model data (for example the parameter values of a GMM) and defines the algorithms to train and evaluate (score) the model on a speech feature data stream. Model elements may specify two speech feature types from the parameterization tree[13]. One is used as the model element's *data stream*, and the other as its *information stream*. The information stream must contain segment label data and may be used by the model element to implement token-dependency, such that the element is applied only to tokens in the data stream indicated by segment labels in the info stream that match a given expression. Tokens may be for example word or phoneme segments, and the token-dependency mechanism be used to implement word- or phoneme-dependent models.

- **score operators** are associated with speaker model elements and are used to compute score values from subordinate model elements within a speaker model.

- **normalization operators** are also associated with speaker model elements. Their purpose is to compute a normalized output score based on scores from the model element itself and scores from model elements in other speaker models, for example background models or pseudo-impostor models.

- **decision makers** make verification decisions based on the score values from their associated speaker model elements, or from decisions from subordinate model elements. Possible output of a decision maker is *accept*, *reject* or *no decision*.

- **threshold setters** can be used to determine target-dependent thresholds, for example by performing experiments on enrollment and/or pseudo-impostor data.

In addition to the module types related to the speaker model, several system-related module types have been defined, such as modules for reading audio waveform files and for writing log and result files. While the mechanisms for reading audio samples or sending logging and result information from and to such modules are part of the generic GIVES platform, actual module implementations are selected to support a particular audio waveform file or a logging file format.

[13] or name a pre-specified node in the tree

The GIVES platform and the majority of the existing modules were developed by the author during 1995-1999, and were used in research that resulted in several papers and much of the work presented in this thesis. The framework has later been used also by colleagues at KTH, Center for Speech Technology, within several research and Master of Science (MSc) projects. Many projects involved developing new modules for the framework (Neiberg, 2001; Garcia Torcelly, 2002; Thumernicht, 2002; Olsson, 2002; Öhlin, 2004), while others were users of the framework for speaker verification experiments (Armerén, 1999; Gustafsson, 2000; Ihse, 2000; Elenius, 2001; Lindblom, 2003; Zetterholm et al., 2004).

Other efforts to create more or less generic software frameworks for speaker verification include BECARS (Blouet et al., 2004) and ALIZE (Bonastre et al., 2005). Independently of any of the mentioned generic platform initiatives, a fairly simple ASV system was created within the framework of the European COST250 project to serve as a publicly available reference system (Melin et al., 1999). It uses LPCC features and a VQ classifier and was implemented as building blocks glued together using the Tcl[14] scripting language. The system is distributed[15] freely for research and education purposes under the title "The COST250 Speaker Recognition Reference System".

3.6.1 Examples

We will now shortly describe how GIVES was used to construct the speaker verification systems described in this chapter.

3.6.1.1 HMM-based system

The HMM-based system in Section 3.3 was realized in the following way. Eight stream operators produce the stream of MFCC plus delta and acceleration feature vectors at point A as illustrated in Figure 3.3. Word-level segment labels provided by an external speech recognizer are simply forwarded to point C for use as an information stream. The information stream is used by the speaker model illustrated in Figure 3.4 to implement word-dependent models. Ten word-dependent HMM model elements (LRHMM) have one normalization operator each (COHORT), and are held by a container model element (HMM_SUBMODEL). A score operator (MEAN) associated with the container merges scores from each word-dependent model element. A decision maker module (DECISION) completes the classifier. Each of the three submodel elements shown in the figure are located in their own speaker model top level containers (omitted in the figure).

[14] http://www.tcl.tk/

[15] http://www.speech.kth.se/cost250/

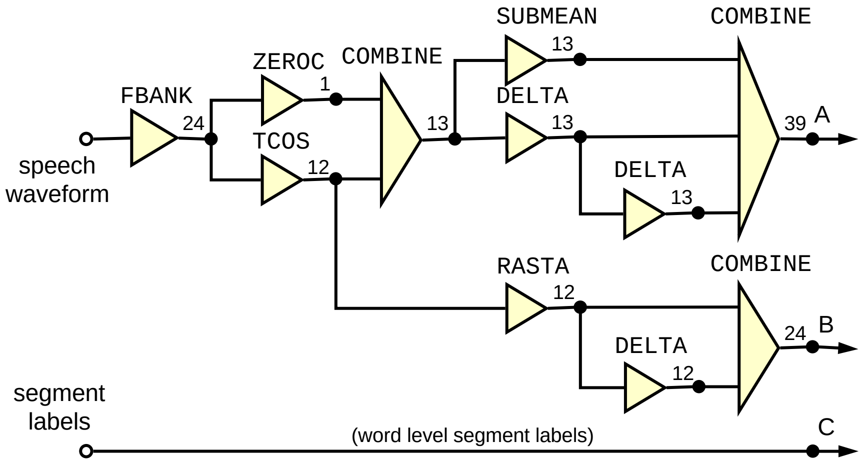


Figure 3.3: Example of a GIVES parameterization tree configuration: the configuration used with the research ASV systems. Data stream at points A and B are used by the HMM and GMM systems respectively. Point C provides the information stream for both subsystems. Numbers indicate sample dimension in nodes of the tree.

3.6.1.2 GMM-based system

The GMM-based system in Section 3.4 is illustrated in Figure 3.5. Compared to the structure of the HMM system, the GMM system has only one model element in each of the target and background models and lacks the intermediate submodel container level of the HMM system. Speech feature data is taken from point B in the parameterization tree (Figure 3.3), but the information stream is the same (point C). In the GMM system, models are made "speech-dependent" rather than word-dependent by applying models to all segments but non-speech segments.

3.6.1.3 Combined system

The fused system used in the PER application, and described in Section 3.5, is a combination of the HMM and GMM systems at the score level. A speaker model top level container (SPMODEL) contains the submodels of the HMM and GMM systems and a score operator (MEAN) to combine scores using weights specified as parameters of the score operator. The combined system is illustrated in Figure 3.6.

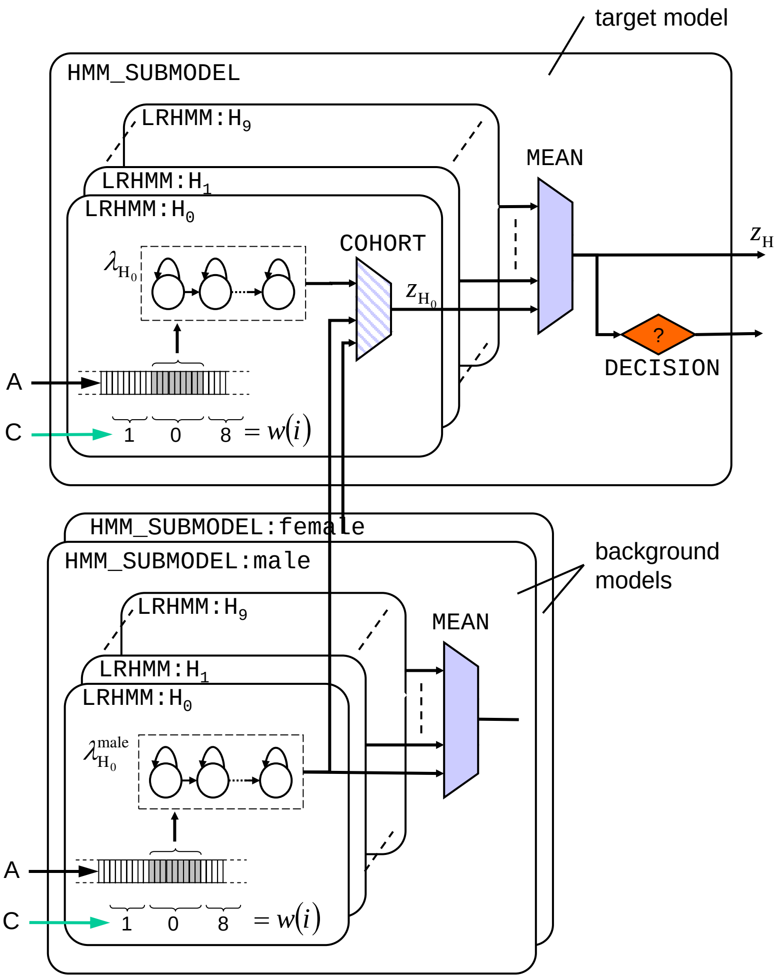


Figure 3.4: The GIVES setup of the HMM system. The three speaker model top level containers are hidden in the figure to reduce cluttering. Points A and C are connections to the parameterization tree, where A is used as data streams and C as info streams.

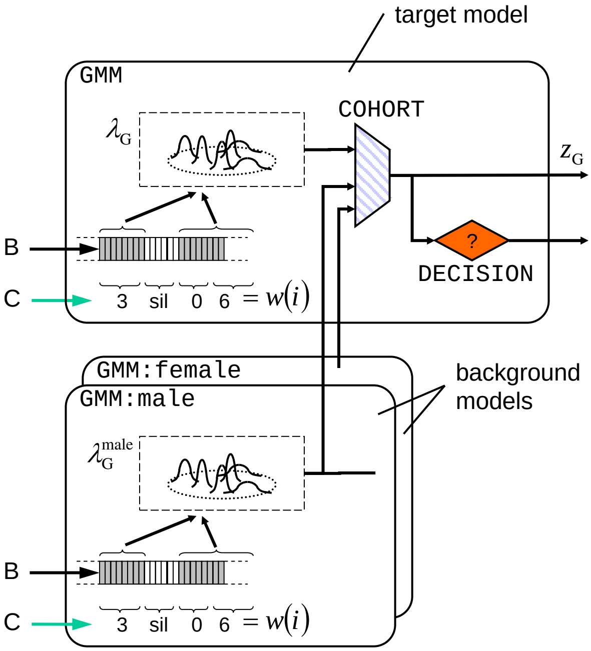


Figure 3.5: The GIVES setup of the GMM system. The three speaker model top level containers are hidden in the figure to reduce cluttering. Points B and C are connections to the parameterization tree, where B is used as data streams and C as info streams.

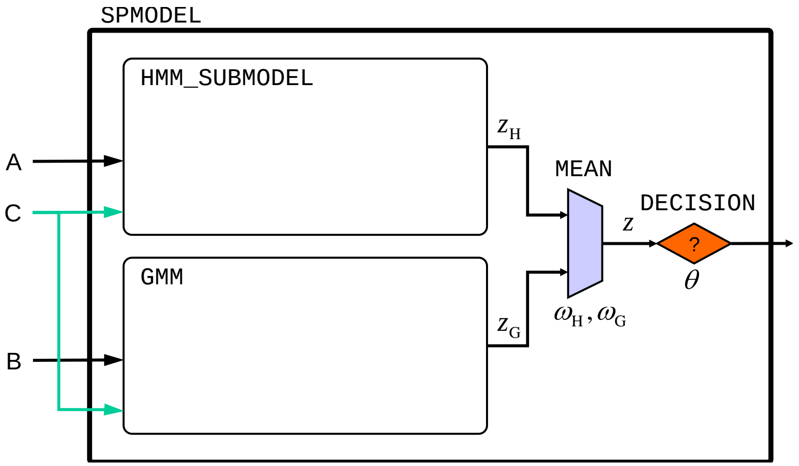


Figure 3.6: The GIVES setup of the combined HMM and GMM system. The insides of the submodels are shown in Figures 3.4 and 3.5. Background models are not shown to reduce cluttering. Points A, B and C are connections to the parameterization tree, where A and B are used as data streams and C as info streams.

Chapter 4

Building prototype applications

4.1 Introduction

Human-machine interface design and usability issues are fundamental for the success of speech technology, and a demonstration system can be useful in studies on these topics. A demonstration system can also be useful in collecting speech data to support evaluation of for instance speech recognition and speaker verification systems.

With the growing commercial interest in speech technology based applications, and an increasing demand on research labs to do industry relevant research, it is also becoming more and more valuable to show practical examples of research advances. This often means live demonstrations of the technology in useful applications. Demonstration systems typically include several speech technology components, such as speech generation, text-to-speech synthesis, speech recognition, speech understanding, speaker recognition, and dialog management. The components require complex interaction with each other and with audio devices, and the components are themselves complex. As a result, a demonstration system is often a complex system. To prevent system building itself to take too much effort away from the more research oriented tasks, such as improving basic speech technology components, it is important to have an efficient framework for building demonstration systems. A framework can be defined by for instance a suitable programming language, a good system architecture and reusable software components. It is important that such a framework is flexible enough to allow researchers to test new ideas, and that it evolves with state-of-the-art in speech technology. This requirement is challenging, because it somewhat opposes the requirement for efficiency. A framework that is efficient and easy to use when building small demonstration applications may not be flexible enough when building for example state-of-the-art conversant dialog systems.

Several publications have reported on efforts in creating frameworks for speech technology applications. A well-known platform is Galaxy-II (Seneff et al., 1998).

It was developed at MIT and has been used successfully in several applications such as the Jupiter, Voyager and Orion systems. It has also been designated as the first reference architecture for the DARPA Communicator Program (Bayer et al., 2001), called Galaxy Communicator. Galaxy-II is a client-server architecture where all interactions between servers are mediated by a programmable hub and managed by a hub script. Galaxy Communicator and a collection of servers, wrappers and application examples (The Open Source Toolkit (OSTK)) is available through an open source project[1].

Jaspis[2] (Turunen and Hakulinen, 2000) is an agent based architecture designed with special focus on multi-linguality and user and environment adaptivity. Sutton et al. (1998) describe the OGI CSLU Toolkit[3] that includes several ready to use speech technology components and a Rapid Application Developer tool. Potamianos et al. (1999) review efforts in defining design principles and creating tools for building dialog systems, including architectural issues.

Several commercial companies offer platforms for developing applications with speech technology. Nuance provides SpeechObjects (Nuance, 2000) as “a set of open, reusable components that encapsulate the best practices of voice interface design”. Philips[4] marketed[5] SpeechPearl and SpeechMania as speech recognition and speech understanding-centric product families. SpeechPearl included SpeechBlocks, in concept very similar to Nuance’ SpeechObjects.

Related to the creation of generic platforms are also several standardization activities. The World Wide Web consortium[6] (W3C) specifies markup languages for voice dialogs (VoiceXML), speech recognition grammars, speech synthesis markup, reusable dialog components, etc. ECTF[7] defines standards for interoperability in the Computer Telephony (CT) industry.

4.2 The ATLAS framework

This section presents an effort to create a framework for multi-modal and multi-lingual speech technology applications. The framework is called ATLAS and is a Java software library that includes a set of application programming interfaces (APIs) for speech technology components. The aim has been to code much of application invariant, low-level functionality in ATLAS and to provide application programmers with a powerful, easy-to-use speech technology API. ATLAS thereby defines a multi-layered system architecture that encourages software reuse. The framework is intended for building demonstration systems in a research environ-

[1] http://communicator.sourceforge.net/

[2] http://www.cs.uta.fi/research/hci/spi/Jaspis/

[3] http://cslu.cse.ogi.edu/toolkit

[4] http://www.speech.philips.com/

[5] In 2002, this part of Philips was acquired by ScanSoft, now Nuance Communications.

[6] http://www.w3.org/voice/

[7] http://www.ectf.org/

ment. A particular effort has been made on developing support for the use of automatic speaker verification.

The ATLAS framework was mainly developed within the CTT-projects PER and CTT-bank. PER is the automated entrance receptionist described in detail in the next chapter, while CTT-bank is a demonstration of a speech controlled telephone banking system (Ihse, 2000; Melin et al., 2001). The framework has then been used when building applications within several other projects, most of them Master of Science (MSc) student projects. Applications and projects include (in chronological order):

- The Picasso Impostor Trainer – impostor training with presentation of targets' authentic recordings and/or feedback of ASV scores. Developed through a MSc project (Elenius, 2001) within the EU-funded PICASSO project and used with "lay" impersonators. It has later also been used in several experiments with professional impersonators, e.g. (Zetterholm et al., 2004).

- ReMember – a "code hotel" sales demonstration application developed at JustDirect, one of CTT's participating companies.

- The Hörstöd project – visualizing telephone speech through the output of phoneme recognition (Johansson, 2002; Johansson et al., 2002; Angelidis, 2003) (MSc and BSc projects within the framework of a CTT project)

- Tilltalad – an application-independent speaker adaptation service (Söderquist, 2002) (MSc project)

- Simon – a voice interface for vehicle inspectors (Gyllensvärd, 2003) (MSc project)

- MultiSense – voice-modality in multi-modal system for pre-operative planning within orthopedic surgery; A MSc project initially developed a stand-alone voice-only application (Svanfeldt, 2003) within the EU-funded MultiSense project. ATLAS was then used in the core project to add the voice modality to the multi-modal system (Testi et al., 2005).

- SesaME – interaction manager for personalized interfaces on top of ATLAS (Pakucs, 2003) (PhD project)

- Metro times – telling when the next metro leaves from KTH; simple application used to demonstrate the addition of a bridge to a commercial API for text-to-speech components (Lundgren, 2003) (BSc project)

- Remote – remote control of home appliances by voice (Bjurling, 2004) (MSc project)

4.2.1 The system model

ATLAS has been designed with the layered system model shown in Figure 4.1 in mind. The model has an application-dependent layer on top, a resource layer in the bottom, and an application-independent layer, *the middleware*, in between. The upper side of the middleware is a powerful speech technology application programming interface (API), and the lower side (as seen from above) is a collection of APIs to speech technology components in the resource layer.

The middleware is itself layered. Each layer adds more powerful functionality and abstraction to the set of primitives that are offered in the component APIs. For retained flexibility, the lower layers are always made available to the application through the API.

ATLAS is first of all an implementation of the middleware illustrated in Figure 4.1, but it also contains foundation classes for the application layer.

4.2.1.1 Terminology and notation

When describing software structures in the following sections, we borrow terms from the object-oriented programming paradigm as used with the Java programming language. In this terminology, a *class* is a collection of data and methods that operate on that data. A class is usually created to specify the contents and capabilities of some kind of object. An object created from its class specification is called an *instance*, or simply an *object*. A *method* is the object-oriented term for what is sometimes called a procedure or a function. For example, a circle may be defined by a radius, a location, and a color. What we would like to do with a circle is perhaps to draw it, move it and calculate its area. With an object-oriented programming language we can then define the class *Circle* with attributes (data) *radius*, *location* and *color*, and methods *draw*, *move* and *getArea*. Once we have the class Circle we can create instances of it, i.e. create circle objects. Each circle object has its own radius, location and color, and can be drawn or moved individually.

In this chapter the word *interface* is used both in its general sense (for example: a human-machine interface, an application programming interface (API)) and in the object-orientation sense. In the latter case, an interface is a collection of methods and usually represent a certain aspect shared between classes. A class often implements several interfaces. In our example, the Circle class would perhaps implement interfaces *Drawable*, specifying the method draw, and *Movable*, specifying the method move.

New concepts, especially method names, are set in italics when introduced in the text.

4.2.2 The middleware

In this section we exemplify the contents of the various layers of the middleware as implemented in ATLAS and illustrated in Figure 4.1. We start at the top with the Dialog Components Layer and proceed towards the Components API Layer.

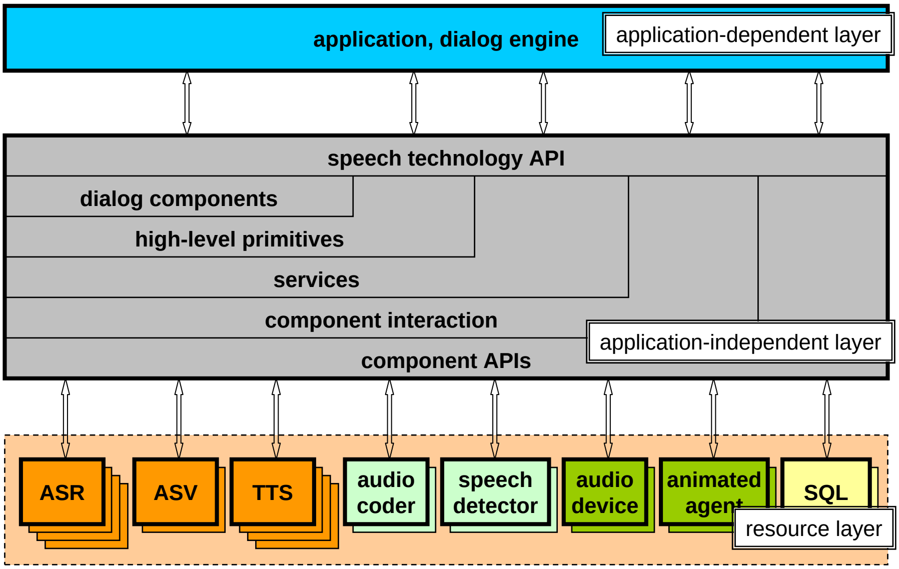


Figure 4.1: The system model behind the ATLAS design. It is layered with an application-dependent layer on top, a resource layer in the bottom, and an application-independent layer in between. ATLAS is an implementation of the middle layer, and provides some support for implementing the top layer.

4.2.2.1 Dialog components

A dialog component is meant to be a powerful object that can solve a specific task within a dialog with the user. The task can be to make a secure user login, to get the name of an existing bank account from the user, or to ask for a money amount. To solve such a task, a dialog component must have some task-specific domain knowledge, such as knowing which customers exist and what accounts they have. The domain knowledge is often supported by an external database. Dialog components should also be able to detect and recover from errors. An error may be an invalid response from the user such as the name of a non-existing account. If the user gives no response at all, or if he asks for help, the dialog component should be able to provide useful help. As part of error recovery, the dialog component may repeat or re-formulate a previously asked question.

The purpose of the Dialog Component Layer is to allow a dialog engine or the application programmer to delegate a well-defined task to an existing component, and allow the re-use of components within and between applications. If no suitable component exists for a given task, the programmer may modify an existing component, create a new one, or choose to solve the problem in some other way. In creating modified or new dialog components, the programmer has access to all the layers in ATLAS. Dialog components are in concept very similar to Nuance' SpeechObjects (Nuance, 2000) and Philips' SpeechBlocks. They also seem to be similar to dialog agents in Jaspis (Turunen and Hakulinen, 2000).

ATLAS itself currently contains three types of dialog components: *login procedures*, *enrollment procedures* and *complex questions*.

The task of a login procedure is to find out who the user claims to be, and then make sure the claim is valid. A login procedure is built from a set of *login operations*, each of which implements a part of the login procedure. The login procedure used in a normal CTT-bank session, for example, contains two login operations. The first is an identification operation that asks the user for his name and ID-number and then looks for a matching customer identity in a database. The second is a verification operation that prompts the user to utter a randomized passphrase and checks the answer for the correct text and for the voice characteristics associated with the claimed identity. The login procedure used in the registration call to CTT-bank, on the other hand, contains a single login operation that performs both the identification and the verification function. This operation asks the user for a unique digit sequence issued to him when he was asked to make the registration call.

While login operations implement the details of login, the login procedure itself adds procedural aspects, such as giving the user a certain number of attempts at a given operation. It also provides a single API to the dialog engine or the application. An important point here is that it is easy for the application programmer to exchange one login procedure for another: it is just a matter of selecting another object for the task.

The task of an enrollment procedure is to elicit speech from a customer, build a representation of the customer's speech, and store the representation in a database. In a CTT-bank registration call, a login procedure is first used to establish the caller's identity as a valid customer. An enrollment procedure is then used to collect ten utterances from the customer. The procedure checks that each utterance is spoken correctly and asks for a repetition if needed. When ten valid utterances have been collected, the procedure trains a speaker model for the customer's voice and stores it in a database. The same enrollment procedure is re-used in the PER system, only modified to exploit a graphical display for showing the user what utterances to speak.

ComplexQuestion is the base class of question dialog components. It was originally created in the CTT-bank application (Ihse, 2000) to implement dialog components for tasks like getting a money amount, the name of a valid account, or the answer to a yes/no-question. It was then generalized and moved into ATLAS together with the yes/no-question class. Classes for confirmation questions (deriving from the yes/no-question) were then added, and a generic list selection question class.

4.2.2.2 High-level primitives

The High-Level Primitives Layer currently contains an *ask* method and a *simplified ask* method. Both methods present an optional prompt from a given prompt text and record and process the answer using a set of *audio processors* (defined in the next section). They normally depend on methods in the Services Layer for their implementation such as *say* and *listen* (also defined and described in the next section). The simplified ask method returns the top-scoring text hypothesis for the spoken answer, while the ordinary ask method gives access to the results of all participating audio processors including multiple text hypotheses and speaker information.

4.2.2.3 Services

The Services Layer provides speech and media input and output capabilities through *play*, *say* and *listen* methods, plus specialized retrieval methods for speech technology components (resources) of pre-defined types.

Speech and media output The *play* method loads media data from file, sends it to one or more media devices, and makes the media devices render it. The *say* method takes a text argument and sends it to a text-to-speech (TTS) component to generate a media stream. It then sends the generated media stream to one or more media devices like the play method. Note that both the play and the say methods can handle multi-modal media output devices, such as speech with face animation. In this case the generated media stream contains two channels, an audio channel and a channel with parameter data for face animation.

Speech and media input The *listen* method is more complex than the play and say methods. Its task is to record a segment of audio from a media device and process it. The processing is done by an optional speech detector and zero or more *audio processors*. An audio processor is a speech recognizer, speaker verifier, or any other object that inputs audio and outputs a result. The configuration of speech detector and audio processors to be used by the listen method is defined by a *listener profile*, central to the design of the speech input mechanism. The listener profile can specify dependencies between audio processors, such that one processor may wait for the output of another processor and use it as input to its own processing. For example, a speaker verifier A may need the output of a particular speech recognizer B to segment an utterance and another speech recognizer C for deriving an identity claim (in the case when a single utterance is used both for identification and verification of an identity). A's dependencies on B and C are then specified in the listener profile as A(B,C). In addition to being sent to audio processors given by the listener profile, the recorded audio segment can be saved to a file.

The listen method is supported by three other methods: A preparatory method sets up media streams and prepares audio processors for a new utterance according to a listener profile. A call to the preparatory method is followed by a call to the listen method itself, that triggers the start of the actual recording (the "listening"). A group of methods can then be used to retrieve results from one or more of the audio processors. When asked for results from multiple audio processors, these methods do some data fusion. Result retrieval methods normally block until results from all audio processors are available. A *maximum processing time* can be specified, however. After this time has elapsed, a method will return with the results available at the time. When all the results have been retrieved, a clean-up method should be called to release resources allocated for the listen operation.

Resource retrieval Specialized retrieval methods are provided for speech technology components (resources) of each pre-defined type. Pre-defined types are currently *speech recognition engine*, *speaker verification engine*, *speech detector*, *text-to-speech engine*, *sound coder*, *media stream player*, *media stream recorder*, *graphical display*, *SQL database connection*, *database monitor* and *file-oriented database.* Additional and more specialized types of media stream players and recorders have also been defined, including *telephony device*, *desktop audio device*, and *audio-visual agent.* Each resource retrieval method comes in two versions: one to retrieve the default resource of a given type and one to retrieve a named resource.

4.2.2.4 Component interaction

The Component Interaction Layer contains resource handling, media stream connections, logging, and several structures for representing various types of information.

In resource handling, all components attached to ATLAS via a component API are abstracted to a resource, and are collected in a resource bundle. The life of a resource starts when it is created and ends when it is closed. While alive, its operation may be monitored to detect if the functionality is lost (the resource is down). Whenever the application or an object within ATLAS needs access to an attached component, it retrieves a handle to the component's API through the component's resource interface. This layer handles all resource types in the same way, while the Services Layer provides specialized retrieval methods for each resource type.

A media stream consists of one or more TCP/IP-based media channels. The end-point of a channel is a TCP socket. By convention, the media producer connects to a server socket opened by the media consumer. When the connection has been established, the producer starts transmitting data in a format specified by the consumer. In most cases, the media stream has a single channel containing audio data. The only current example of a multi-channel stream in ATLAS is the stream from a text-to-speech synthesizer to an audio-visual agent, where a second channel contains parameter data for the face animation. Media streams are created on a per-utterance basis.

Several types of information are passed between components, the ATLAS layers, and the application. The Component Interaction Layer provides data structures to hold such information. An example is the *utterance information* structure that holds information about the contents of a spoken utterance. This may be the output of a speech recognizer and may be used as input by the application itself or by another audio processor, such as a speaker verifier or a parser. Currently the utterance information structure supports scored text hypotheses, word timing information, and speaker information, but could be extended to support for instance syntactic and semantic information.

Calls and events in an ATLAS application can be logged and stored in XML log files. The root node of a log file corresponds either to an application run or to the life of a single session. Objects and methods in all layers add nodes and leaves to the XML tree structure to log information such as start times, duration, input values and results. A Document Type Definition (DTD) for ATLAS log files is included in Appendix G. It specifies the structure of the XML log files and indicates what information is logged.

4.2.2.5 Component APIs

A component API has been defined for each of the pre-defined resource types listed above (Section 4.2.2.3). Some of the APIs are complex in that they are represented by several interfaces. The speech recognizer API, for instance, consists of a recognizer *factory*, a recognizer *engine* and a recognizer *utterance*. They are related in such a way that a factory creates engines, and engines process utterances (segments of audio data). Furthermore, the recognizer utterance interface uses the utterance information structures defined in the Component Interaction Layer to represent its recognition results. The recognizer engine interface also extends the audio processor interface (described in Section 4.2.2.3). Similarly, the speaker verification API includes a verifier engine and a verification utterance. These are based on the SVAPI[8] standard speaker verification API. Besides the functionality covered by SVAPI, the speaker verification API in ATLAS has been extended to handle ATLAS-type media streams, and to have the verifier engine extend the audio processor interface.

The TTS API also contains a factory interface and an engine interface. Utterances are handled with a method call in the TTS engine, rather than with a dedicated utterance object. The synthesis method and language are specified when a TTS engine is created and cannot be changed later. Voice properties for the selec-

[8] SVAPI was a result of collaborative efforts of many companies, including Novell, Dialogic, IBM, Motorola and many others. Unfortunately, the SVAPI initiative was closed down. Motorola CipherVOX claim to support SVAPI (Fette et al., 2000), but otherwise there are few or no products that actually support SVAPI. BioAPI is an API that is currently being actively developed for more biometric modalities than speech (`http://www.bioapi.org`).

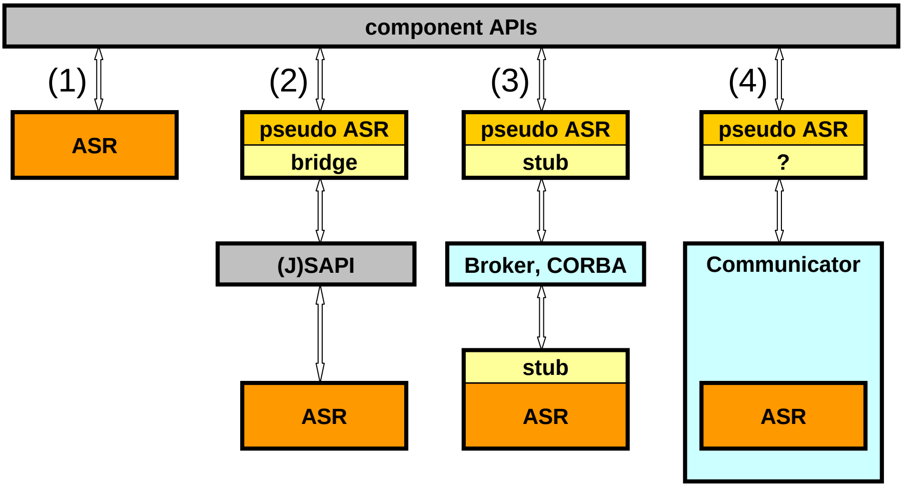


Figure 4.2: Four examples of how a speech technology component, in this case an automatic speech recognition (ASR) engine, can be connected to an ATLAS component API.

ted synthesis method, such as pitch level, can be changed, however. An application can change voice or language by creating multiple TTS engines and switch between them.

4.2.3 The resource layer

As already mentioned, the resource layer refers to a collection of (speech technology) components used by an application. In this section we first elaborate on how components can be connected to ATLAS, and then list what components are currently available. Let us emphasize that the components themselves are not part of ATLAS, and that ATLAS is rather useless without a set of good components.

4.2.3.1 Component implementation

A component API, at the lower side of ATLAS, specifies how an application or an ATLAS layer can interact with a component, while leaving a lot of freedom for how the component is actually implemented. Since ATLAS is implemented in Java and the component APIs are defined in terms of Java APIs, the component as such must be a Java object. But what if we already have a speech recognizer engine written in, for instance, C++? Then we can create a pseudo-implementation of the engine in Java that uses the existing C++ code to do the actual work. Figure 4.2 illustrates four examples of how a speech recognition engine (labeled ASR in the figure) can be connected to ATLAS through the component API.

In the first example, the engine already has a Java implementation of the component API. Either the engine is coded in Java or it is coded in a native language (C/C++) but has a Java wrapper using the Java Native Interface, JNI.

In the second example, the engine supports another API than ATLAS' component API. This may be an industry standard API, such as Microsoft's SAPI[9] or Sun's JSAPI[10], or an engine vendor-specific API. Provided the ATLAS API can be mapped to the other API, a pseudo-implementation of the ATLAS API could be created that operates as a bridge between the two APIs. Such a bridge can possibly be used with other engines that support the same standard API.

In the previous two examples, the engine is likely to execute in the same process as ATLAS itself, while in the remaining two examples the engine may be implemented as a server in a separate process. The third example illustrates a plain server implementation, where a small pseudo-implementation of the ATLAS API communicates with the server through some inter-process communication (IPC) mechanism, such as the Common Object Request Broker Architecture[11] (CORBA), Java Remote Method Invocation[12] (RMI), or the CTT Broker (described in Section 4.2.3.3).

The fourth and final example in Figure 4.2 indicates the possibility to interface to an engine that is integrated into another speech technology system, such as the Galaxy Communicator[13]. This could include interfacing several other Communicator engines (text-to-speech engine, parser, etc.) at the same time through a single bridging mechanism. Alternatively, each single engine could be attached directly, like in examples two and three.

4.2.3.2 Available components

In this section we list the currently available components that implement an ATLAS component API and thus can be used with ATLAS. We pay special attention to how each component is connected to ATLAS and give references for the underlying technology, but otherwise keep descriptions very brief. More detailed descriptions for some of the components can be found in (Melin et al., 2001).

Five components are available as internal resources executing in the same virtual machine as ATLAS (Figure 4.2, example 1):

- the Starlite speech recognizer (Ström, 1996); acoustic triphone models trained on SpeechDat databases (Höge et al., 1997; Elenius, 2000) are available for Swedish and English (Salvi, 1998; Lindberg et al., 2000)
- the ACE speech recognizer (Seward, 2000); the same acoustic models are available as for Starlite

[9] http://www.microsoft.com/speech/
[10] http://java.sun.com/products/java-media/speech/
[11] http://www.corba.org/
[12] http://java.sun.com/products/jdk/rmi/
[13] http://communicator.sourceforge.net/

- an energy and zero-crossing rate based speech detector
- a sound coder; performs audio format conversion, speech parameterization for speech recognizers and can fork audio streams.
- a file-oriented database.

(All components except the speech detector and the ACE recognizer are also available as CTT Broker servers, see below.)

One component uses a bridge to an industry standard API (Figure 4.2, example 2) to connect commercial TTS engines to ATLAS (Lundgren, 2003). Actually, two standard APIs and two bridges are used. The first bridge maps the ATLAS TTS engine API to JSAPI. A commercially available bridge, TalkingJava SDK from CloudGarden[14], then maps JSAPI to Microsoft's SAPI. TalkingJava SDK features are used to connect ATLAS media streams to the TTS engine. This component can be used[15] to connect to for example Acapela's[16] BrightSpeech TTS engine (using non-uniform unit selection technology) and the Swedish voices Emma and Erik.

The remaining components are implemented using a server and some IPC mechanism to communicate with the server (Figure 4.2, example 3). The two first are SQL database components that interface MySQL[17] and Borland InterBase[18] databases, respectively. Since JDBC was chosen to be the component API for SQL databases in ATLAS, these components simply use standard JDBC drivers to communicate with database servers. JDBC drivers implement their own (proprietary) IPC mechanism. Note that these ATLAS components add very little on top of the JDBC driver itself; they merely define the name of a driver and load the driver into the virtual machine.

The remaining IPC-based components are implemented as clients to CTT Broker servers. Available components are:

- a text-to-speech component using RULSYS (Carlson et al., 1982) for text-to-phone conversion plus GLOVE (Carlson et al., 1991) or MBROLA (Dutoit et al., 1996) synthesizers. Several Swedish and English voices are currently available, including Lukas (Filipsson and Bruce, 1997) and the Acapela voices Ingmar, Annmarie and Roger. It can generate media streams for multi-modal output (face and voice) (Beskow, 1995).
- a Starlite speech recognizer (Ström, 1996); also available as an internal component.

[14] http://www.cloudgarden.com/

[15] This component is still experimental and its source code is available on a development branch of ATLAS only.

[16] http://www.acapela-group.com/

[17] http://www.mysql.com/

[18] http://www.borland.com/interbase/

- a speaker verifier based on GIVES (Chapter 3). Text-dependent modes for Swedish and English are available together with text-independent modes for Swedish.
- a media device with animated agent output and audio-only input (half-duplex mode) (Beskow, 1995; Gustafson et al., 1999).
- a media device, "digitizer", with desktop (half-duplex) audio output and input based on the Snack toolkit[19] (Sjölander and Beskow, 2000). Includes an optional WaveSurfer-based[20] GUI.
- an ISDN media device with telephony call handling and (half-duplex) audio output and input.
- a sound coder component; also available as an internal component.
- a file-oriented database; also available as an internal component.

In addition to the above CTT Broker-based components, a registry component interfaces a registry in the Broker that keeps track of host-specific servers.

The text-to-speech, speech recognition, agent, digitizer, sound coder and file database Broker servers were originally developed as part of other projects at CTT. They were later improved and adapted to work well with ATLAS.

4.2.3.3 The CTT Broker

The CTT Broker[21] created by Erland Lewin in 1997 is an architecture for interprocess communication that is helpful when building modular and distributed systems. It was initially developed within the ENABL (Bickley and Hunnicutt, 2000) and August (Gustafson et al., 1999) projects. The Broker can be used with ATLAS for communicating with several speech technology components implemented as Broker servers, as indicated above. It has also been used recently in the AdApt (Gustafson et al., 2000), Higgins (Edlund et al., 2004) and KTH Connector (Edlund and Hjalmarsson, 2005) dialog systems. (In the latter three system, ATLAS was not used.)

The primary function of the Broker is to pass message strings between servers through TCP ports. To manage this, it also keeps track of what servers are connected. The basic, lightweight protocol uses a short header for the Broker's own use attached to the actual message string. The header includes a message type indicator and address information, where message types exist to connect and disconnect a server, to send procedure or function calls to a server, and to send a return or error value in response to a function call. It is up to each server to define syntax and semantics for the actual message strings - the Broker simply passes this

[19] http://www.speech.kth.se/snack

[20] http://www.speech.kth.se/wavesurfer

[21] http://www.speech.kth.se/broker/

string from sender to receiver without interpreting or altering it. The string based message protocol and the use of TCP port based connections make the operation of the Broker platform-independent, in that servers can run in any programming environment and operating system that supports TCP connections. The Broker itself is implemented in Java and can therefore run on any platform that supports Java.

A secondary function of the Broker is to start servers on demand, and to detect when a server is closed. It uses a database of *startable servers* that defines what servers can be started and how to start them.

To aid the creation of servers, software libraries have been created for several programming languages, including Java, C, C++, Tcl, Perl, Prolog and Mozart/Oz[22]. With these libraries a server can register itself with the Broker and make calls to a remote server using constructs in the used programming language, rather than handling low-level TCP connections directly. For example, with the Java library a server creates an instance of the *BrokerClient* class and calls the instance's *connect* method. It can then make remote calls to another server by calling a method *callFunc*, giving the name of the remote server and the message string as arguments. The callFunc method blocks until the Broker sends a reply and then either returns a value or throws an exception.

In addition to the basic call functionality, some of the libraries (currently Java, C++ and Tcl) provide a parser for the contents of a message string that can route calls to *classes*, *instances* and *attributes* inside a server. Language constructs are also available to represent classes and instances in the server. With this mechanism, the concept of remote objects is supported. The remote object concept, the parser, and the corresponding message structure are entirely optional, but are used by all servers currently available through the ATLAS platform.

Using the remote objects concept, an event mechanism has been implemented using a publication metaphor. A server creates a publication for publishing certain information, and servers subscribing to the publication gets an update message every time new information is published. This event mechanism is for instance used by the Broker itself to make server connection status information available. By subscribing to such a publication, an application can for instance know when a server is lost. ATLAS uses this feature with all Broker-server based resources. When a Broker-server based resource is created, ATLAS automatically subscribes to status information for the corresponding server connection. ATLAS is then notified if the server is lost, and can take measures to for instance re-create the resource. The event mechanism is currently available in the Java and C++ libraries only.

The CTT Broker architecture has similarities with other inter-process communication architectures[23]. The Galaxy-II hub, for instance, also organizes servers in a star topology where all server-to-server messages pass through the hub. The hub has a programmable controller function, however, that the CTT Broker has not.

[22] `http://www.mozart-oz.org`

[23] URL references for CORBA, Java RMI and Galaxy-II were given on p. 67.

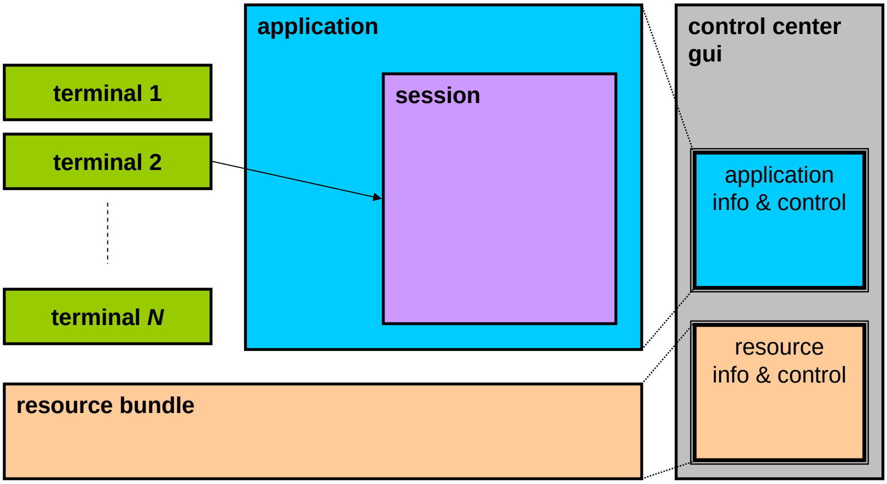


Figure 4.3: Application structure implemented by ATLAS application support classes. The application object creates sessions upon incoming calls from a terminal. The Control Center GUI is optional and any application can run without it.

CORBA and Java RMI provides support for manipulating remote objects almost as if they were local objects. Similar functionality can be achieved with the Broker and its support libraries. It is left to the Broker server designer, however, to provide client-side APIs that allow remote objects to be manipulated as if they were local objects. Such client-side APIs (stubs) are generated automatically with CORBA and Java RMI.

Audio and other binary streams are usually not sent through the CTT Broker. Instead, servers communicate through the Broker to setup direct connections where binary data is transmitted. This is the same as in Galaxy-II.

4.2.4 Applications

4.2.4.1 Application support

Apart from providing an implementation of the middleware illustrated in Figure 4.1, ATLAS provides a set of support classes for the application-dependent layer of a system. This includes interfaces and super classes for *application*, *session* and *terminal* classes.

The provided super classes can be used to create applications with the structure illustrated in Figure 4.3. The idea is that the application corresponds to an object that is created once. The application can then create session objects whose lives correspond to physical sessions of interaction with a user through a terminal. It is usually the session object that does something interesting using the speech technology API in ATLAS. The current implementation limits the number of concurrent sessions to one, for reasons of simplicity. We believe this to be sufficient for most research situations.

Each session object is connected to a single terminal. A terminal may be telephony based, in which case the session naturally corresponds to a telephone call, or desktop based. With a desktop-based terminal the session metaphor does not come as naturally as with a telephony-based terminal. It is left to the implementation of the terminal object to decide when a session should start, and to the terminal or the session to decide when a session should be terminated. Common to all terminals is that they provide a means for initiating a session with the application, and that they are associated with an audio output and an audio input device. Optionally, a terminal may also have a means to close a session and may be associated with a display.

One of the key features of ATLAS and the arrangement illustrated in Figure 4.3 is that applications and sessions can interact with any type of terminal transparently (as long as they do not require particular properties of specific terminals). In CTT-bank, for example, a session normally interacts with the user via a telephony-based terminal, but it can also use a desktop-based terminal. In fact, this was often exploited during the development phase of the project. The desktop terminal could even be extended with output through an audio-visual agent. To take full advantage of multi-modal output, however, the session code needs to be modified to send requests for animated gestures to the agent, to make the agent look more alive. With such modifications the session would still run with a telephony-based terminal, since gesture requests are simply discarded if the terminal cannot visualize them.

Beside sending audio through the audio output device associated with the terminal connected to a given session, the application or the session may choose to add other output devices.

ATLAS has been internationalized[24] with respect to the language spoken within the application. That is, assuming the application-dependent part of an application is also internationalized, the application can be localized[25] to a new language. Localization in this case involves translating text elements related to generating system prompts and interpreting user responses, and adding resources for the new language (or making sure the existing ones support the new language). All such text elements within ATLAS, i.e. in its dialog components, have been localized to Swedish and English, and both languages are supported among speech technology components listed in Section 4.2.3.2. Basically, internationalization means separating text elements from program code. Text elements are stored in text files that are read by the program at runtime. There is one or more text files per language,

[24] Internationalization is the process of designing an application so that it can be adapted to various languages and regions without engineering changes.

[25] Localization is the process of adapting software for a specific region or language by adding locale-specific components and translating text.

and text elements are localized by creating a corresponding set of text files for the new language.

An important additional advantage of separating text elements from program code is that changing system prompts, and especially hand tuning prompts for optimal synthesis output, requires no knowledge of the programming language used to code the application. Having all prompt texts collected in one place also provides for a good overview.

A graphical user interface (GUI) to the application and the resource bundle is provided. It is called the Control Center and is entirely optional - any application can run without it. The application part of the GUI provides a possibility to start and stop the application and to view the current application or session log. It also has a message pane that shows when the application was started or stopped, and when sessions are created and ended. The resource bundle part of the GUI provides possibilities to select the current language and to change default resources of each type for each language. The latter facility enables the operator to, for instance, select a TTS engine with another voice to be the default. This effectively changes the voice of the application, given that the application is coded to use the default TTS engine. The resource bundle GUI also has a message pane that logs resource status information.

The application class supports recording of audio and video during sessions through a video capable network camera[26]. This may be useful for example when studying subject behavior during interaction with the application. In PER, a camera was used to record who was actually talking to the system.

4.2.4.2 Examples

In this section we shortly describe four systems that use ATLAS, as examples of how the platform can be used. For each system, we explain its task, what has been coded in its application-dependent layer, which ATLAS layers are used, and what speech technology components are included in the resource layer.

All four systems have in common that their application-dependent layer is coded in Java and that it uses ATLAS application support classes to implement application and session classes.

CTT-bank CTT-bank is a speech controlled telephone banking system (Ihse, 2000; Melin et al., 2001). Customers identify themselves to the system by saying their name and a short digit sequence. The digit sequence is chosen by the customer during registration, and is used to make the identification phrase unique. After claiming an identity, he verifies the claim by repeating a four-digit passphrase generated by the system. Once allowed access to the system, the user can check account balance, list recent transactions, and transfer funds between accounts.

[26] Network cameras from Axis Communications are currently supported.

The application-dependent layer defines several dialog components to implement the banking services and part of the registration dialog. Dialog components use methods and objects in various ATLAS layers for their implementation. ATLAS dialog components for enrollment and login are extended and specialized, and used to implement registration and user authentication dialogs. Specialization includes using an error-correcting code with a seven-digit registration number used to authenticate the user during the registration call, and changing prompt texts to fit the application.

The resource bundle contains a speaker verification engine, several speech recognition engines, several text-to-speech engines, a speech detector, two ISDN terminals, a desktop-based terminal, a sound coder, a file-oriented database and a MySQL database driver. The multitude of speech recognition engines is needed because the used speech recognizer does not support on-line grammar modification. One engine is therefore created for each specialized grammar used in the application (Melin et al., 2001).

PER The PER system (described in detail in the next chapter) is installed at the central entrance of the Department of Speech, Music and Hearing. The system is basically a voice-operated lock: employees at the department may open the door by saying their name followed by a random digit sequence displayed on a screen. Speech recognition and speaker verification is used to authenticate the user, and an animated agent gives the user feedback by greeting him or asking him to try again. The physical installation includes a screen, a high-quality microphone, a relay to unlock the door, and several sensor devices to detect the presence of a person.

Most of the current dialog is implemented by ATLAS dialog components for enrollment and login. The current application has been localized to Swedish and English. It is not a prioritized task, however, to keep system extensions, such as more advanced language understanding and dialog control, bilingual.

The resource bundle contains one speaker verification engine, two speech recognition engines, and several text-to-speech engines per supported language. It also contains an animated agent, a graphical display, a speech detector, two ISDN terminals, one terminal object per detector, a sound coder, a file-oriented database and a MySQL database driver. Several of the resources were created especially for this application, including the display that presents the random passphrase on the screen, and detectors which signal state changes in physical sensor devices.

Regarding the session metaphor used in ATLAS application support classes, each terminal object (except the telephone-based) uses a detector to decide when to trigger the start of a new session in the application. It is then up to the session logic to decide when the session has finished, i.e. when a user has left.

Picasso Impostor Trainer The Picasso Impostor Trainer system was developed for a study on speakers' ability to imitate other speakers (Elenius, 2001). A subject calls the system while sitting in front of a computer. He speaks a list of digit sequences to provide a sample of his normal voice, and the system compares the voice to a list of speaker models and selects two target speakers for imitation. The subject can then interactively practice to imitate a target speaker under controlled conditions: by listening to recordings of the target speaker, by watching a display with scores from the speaker verification system for his own practice utterances tested against the target speaker's model, and combinations of the previous two. After each training round, the subject speaks ten utterances without feedback to test if he is able to alter his voice to get "closer" to the target speaker in the sense of the speaker verification system.

The application-dependent layer uses no dialog components. Instead, the application uses the listen method in the ATLAS Services Layer to input and process utterances, and the play method to play back pre-recorded samples of target speakers. It has an elaborate GUI for system feedback to the user and for mouse input.

The resource bundle contains a speaker verification engine, a speech recognition engine, a speech detector, a sound coder, a media file database, and an ISDN terminal.

Hörstöd This system was developed for investigating if a hearing impaired person can be aided by transcriptions produced by a phoneme recognizer in understanding speech during a telephone conversation (Johansson, 2002; Johansson et al., 2002; Angelidis, 2003).

The application-dependent layer is fairly similar in content to the Picasso Impostor Trainer. It uses a telephony terminal for audio input and a GUI for graphical output. It uses ATLAS High-Level Primitives Layer to input and process utterances, and no dialog components are used. The resource bundle contains the same resources as in Impostor Trainer, except that no speaker verifier or speech detector are included.

4.2.5 Discussion

The main difficulty in creating a generic application platform is to make it both efficient and flexible. For a given application, the platform should be efficient to use to minimize development costs. But it should at the same time be flexible enough to be efficient for another type of application, and to allow adaptation to new types of applications. We believe that the layered structure employed in ATLAS is powerful in this regard. By providing low-level APIs and structures, where little is assumed about the overall application structure, we allow very diverse applications to share at least the same speech technology components. In the higher, more powerful layers, we assume more and more about application structure. These layers are efficient to use with applications for which these assumptions are valid, and may simply be ignored by applications for which they are not. To develop the platform to provide powerful layers also to new and diverse applications, we can either adapt the existing layer implementations and generalize them, or we can create parallel implementations with other assumptions regarding system structure. As a development process, we suggest to first develop new applications with whatever parts of ATLAS are useful, then to analyze the application-specific code to see what is general and what is specific to the particular application, and finally to successively move the general parts into ATLAS. This is how ATLAS can evolve with research advances.

VoiceXML[27,28] is a standard markup language for representing human-computer dialogs. To relate ATLAS and VoiceXML to each other, we first try to describe the latter in the context of the system model illustrated in Figure 4.1. The VoiceXML standard primarily defines a specification for the interface between the application-dependent layer and a voice browser. The application-dependent layer in a VoiceXML application is very "thin" and is represented by a set of XML documents. The voice browser is an application-independent engine that implements dialogs according to given VoiceXML documents. It thus includes the functionality of the middle layer and the resource layer of Figure 4.1 (though it may have an entirely different structure). We therefore suggest that VoiceXML corresponds to the speech-technology API in ATLAS.

While ATLAS gives the application programmer access to all its internal layers for retained flexibility, VoiceXML provides access to rather high-level functionality in the voice browser, but not to low-level details. For instance, a VoiceXML application can tell the browser to ask a multiple-choice question, but cannot manipulate the voice browser's speech recognizer directly (except possibly through a browser vendor's proprietary features). VoiceXML has been created based on some assumptions about system structure and capabilities, and as a standard it also imposes corresponding constraints on what applications can be created. It is therefore efficient for those kinds of applications. Because VoiceXML is a standard, a VoiceXML application can also be executed in a standard-compliant voice browser from any vendor[29].

VoiceXML does not have (standard) support for biometric user authentication. Rua et al. (2004) have suggested how support for identity verification tasks based on multi-modal biometrics could be added.

Creating media streams on a per-utterance basis allows demands for real-time performance in some parts of the system to be reduced, compared to if a continuous (unbuffered) media stream on a central bus were used. This is an advantage in a research system since it allows, for instance, an experimental speech recognizer to take the time it needs to prepare for and process an utterance. A slow component need not risk that other components involved in the processing of the same utterance looses any samples. We also believe per-utterance streams make system programming somewhat easier. It has a couple of disadvantages, however. Setting up streams for every new dialog turn or utterance takes more time than simply telling a device to start listening on an already connected media bus, possibly res-

[27] http://www.w3.org/voice/

[28] http://www.voicexml.org/

[29] Several vendors market voice browsers that implement VoiceXML, including Tellme, Motorola, Nuance and Pipebeach (now HP).

ulting in a slower system. It may also make it more difficult to use full duplex input/output streams, to implement barge-in, etc. Most standard APIs related to audio and speech also tend to assume a central media bus. Thus we consider using continuous, buffered media streams for the future.

For ATLAS component APIs we have in general not used public standard APIs (the exceptions are JDBC for SQL database connections and SVAPI for the speaker verification). This is because, first of all, for most current component implementations we use in-house technology developed before ATLAS was conceived, and we chose to design APIs that match the abilities of the current technology. Implementing a standard API, such as the Java Speech API (JSAPI), for the TTS for instance, would have resulted in overhead work at this stage. Second, the main candidate for a standard API for speech engines in ATLAS would be JSAPI, and there were not (in year 2001) many engines available that supported JSAPI. Furthermore, JSAPI in its current state does not integrate well with the corresponding Java Sound API and Java Telephony API that would enable us to maintain the audio device independence of ATLAS. Third, as we outlined in Section 4.2.3.1, an ATLAS component API can be mapped to a standard API via a bridge to enable the use of engines with a standard API. Today there is at least one TTS engine that natively supports JSAPI (FreeTTS[30]), and with the CloudGarden TalkingJAVA SDK to bridge JSAPI to SAPI, many SAPI engines could be used as well.

A corresponding division between internal and standard APIs is seen in Jaspis (Turunen and Hakulinen, 2000). In its input/output architecture, *virtual devices* are abstract units that represent more concrete *engines*. Virtual devices serve as the interface between engines (below) and *agents* and the *communication manager* (above) and partly correspond to ATLAS component APIs. Below the virtual device level in Jaspis are the *client*, *server* and *engine* levels, and standard APIs are employed between the server and engine levels (cf. Figure 4.2, example 2).

4.2.6 Conclusions

ATLAS has been presented as a framework for building demonstration applications with speech technology. So far it has proved useful for research in two EU-projects and several CTT projects. The CTT-bank system has been used both in a usability study and for collecting data for evaluation of speech recognition performance. The Hörstöd system has likewise been used in a usability study and to test the performance of a phoneme recognizer (Johansson et al., 2002). The Picasso Impostor Trainer was used to test how speakers are able to imitate other people's voices. The PER system has been used to test speaker verification performance. ATLAS also enabled one of CTT's associated companies to build a demonstration application (ReMember).

The high-level speech technology API and the application support classes in ATLAS make application building easier compared to programming with speech

[30] `http://freetts.sourceforge.net/`

technology components directly. Eight of the current ATLAS applications were created as part of student projects. The platform has thus proved to be useful also for educational purposes.

Chapter 5

The PER system

5.1 Introduction

The PER (Prototype Entrance Receptionist) system is an application of speaker verification created at KTH Department of Speech, Music and Hearing. Its primary, on-site version is essentially a voice-actuated door lock that provides staff and students working at the Department on a regular basis with a means to unlock the central gate to their workplace. The speaker verification system is text-dependent. Users are authenticated using a single repetition of their proper name and a visually prompted, random sequence of digits. The automatic collection of enrollment and test utterances is governed by the system through the use of speech recognition, multi-modal speech synthesis and a graphical display.

In addition to the on-site version of PER, a telephone version was created to support the collection of parallel on-site and telephone data. The speaker verification and speech recognition system components are the same for both system versions, including background/acoustic models and the choice of feature representations. They were initially developed using landline telephone data, and were expected to perform better with landline telephone calls than with calls from mobile telephones and in the on-site version of the system.

The first (on-site) version of PER was installed in 1999 as part of a student project (Armerén, 1999), and the system and its components has since then been improved successively until May 2003 when data collection for an evaluation started. The corpus resulting from this collection is described in Section 6.3 (p. 101) and results from the evaluation are presented in Chapter 10 (p. 191). This chapter focuses on the description of the two versions of the PER system.

5.2 On-site version

The on-site version of the system, depicted in Figure 5.1, serves an entrance where users are physically present. It uses a graphical display to prompt claimants visually

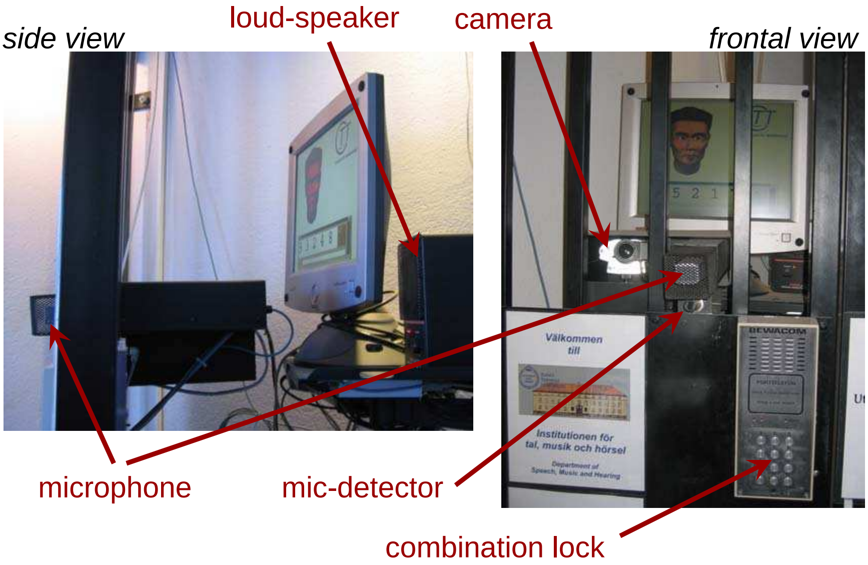


Figure 5.1: The on-site version of PER from side and frontal views. Photos by Botond Pakucs.

for five-digit passphrases. A new passphrase is generated every 10 seconds and the display is updated correspondingly until the system detects the presence of a person. Presence detection is implemented through a diffuse reflection type photoelectric sensor (Diell MS6/00-1H). The sensor is mounted just below the microphone (cf. Figure 5.1) and triggers the start of a session when an object is within approximately 20 cm from the microphone. To allow claimants to start speaking immediately upon arriving at the microphone, the idle system is set in a stand-by mode where it is continually recording audio into a one-second circular buffer. When the presence of a person is detected, input processing starts from the first sample stored in the buffer at that time.

Speech input is recorded with a Sennheiser MD441U directed microphone and sampled at 16 kHz with 16 bits per sample via a SoundBlaster Live! sound card and an external pre-amplifier (M-Audio AudioBuddy). This sample stream is stored to file for future wide-band experiments, while it is decimated to 8 kHz and compressed to 8 bits/sample using A-law coding for use in the on-line processing by system components described below. The decimated and compressed sample stream is also stored to file for off-line experiments. It was used for internal processing because our available development data were telephone data, and both acoustic models for speech recognition and speaker verification background models were trained on such data (cf. Section 10.2.1, p. 193).

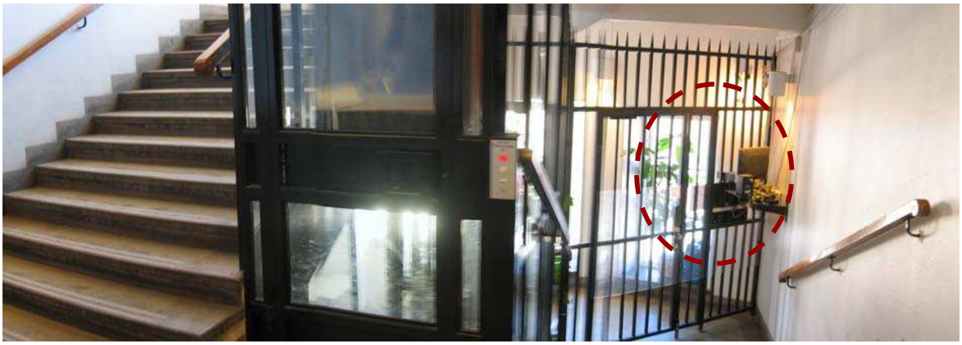


Figure 5.2: Panorama view of the bar gate with the on-site version of PER. Photo by Botond Pakucs.

A video camera (Axis 2120 Network Camera) is installed next to the microphone to capture close-up images of claimants (cf. Figure 5.1). The purpose of the camera is to help annotators decide if a claimant is the target speaker or not, to annotate sessions from the same (unknown) impostor speaker with a single identity label, and to speed up annotation work in general (cf. Section 6.3.3.1, p. 107). The camera is not used for automatic face recognition, while this would have been an obvious possibility for this on-site application. Figure A.3 (p. 259) shows examples of images captured by the camera.

The gate where PER is installed[1] is an iron-bar gate located in a spacious stairwell just below the two floors housing the Department. The stairwell is a reverberant room with stone floor and bare concrete walls. It spans three floors of the building and contains several potential sources of transient background noise such as doors and talking people. Figure 5.2 shows a picture of the stairwell with the PER system to the right of the bar gate as seen from inside the Department. Figure A.2 (p. 258) shows a closer picture of the gate and PER.

The reverberation time (T60) of the stairwell was measured by Nordqvist and Leijon (2004) to 2.4 s at 500 Hz and 2.1 s at 2000 Hz, while both corresponding values for a typical office in the Department were measured to 0.7 s.

The PER system provided one of three possible ways for employees to unlock the gate, the other two being a combination lock and a regular door lock.

5.3 Telephone version

The main differences introduced in the telephone version of the system with respect to the on-site version are listed in Table 5.1. Except for the choice of authentication

[1] describes the location where evaluation data were collected; the Department (and PER) has moved since then (Figure A.1 on p. 257 shows the new installation)

method during enrollment explained in Section 5.5, differences are all motivated by the limited number of available choices of output modalities in the telephone case, or by the standardization in the telephony system. With (traditional) telephones the speech and audio modality is the only available one, while for on-site applications, any modality could be used. (However, in this work on-site output was limited to the (multi-modal) speech and graphics modalities.)

As shown in the table, the prompting method differs between the two versions of the system. In the on-site case the graphical display is used to prompt the digit string visually. This has several advantages over aural prompting like in the telephone case:

A. No aural prompt is needed to initiate the first attempt from the claimant, allowing for a short time from session start to system decision.

B. Longer digit sequences can be used, allowing for reduced speaker verification error rates. In a previous study (presented in Chapter 8) it was found that using five digits with aural prompting introduced a lot more errors and disfluencies in user responses compared to using only four digits, suggesting that four digits is an upper limit for practical use with aural prompting. Preliminary observations with PER indicated that visual prompts with five digits caused no difficulties for users.

C. With visual prompts it is quite possible to collect the name and the digit sequence in a single utterance, again allowing for a short time to system decision. With aural prompts we believe this would be very difficult for users because of an increased cognitive load and limitations in short-term memory in users, so we chose to collect name and digits separately in a two-step procedure.

5.4 Web interface

A web interface to an SQL database used by PER is provided through the Department's intranet server. The interface serve several purposes. The two first are related to normal use of an access control system:

- Regular PER users (clients) visit their personal page to enable gate or telephone enrollment, and to generate enrollment sheets for telephone enrollment. They can also customize PER's greetings to them.

- The system administrators use a privileged part of the interface to add new users to the database, or delete users. Statistics on system use is also provided, such as enrollment status of users, enrollment durations, the number of sessions, access times, the number of attempts required for access and a list of error messages.

Table 5.1: Differences between the on-site and telephone versions of the system.

Property	On-site	Telephone
Transducer	directed, high-quality microphone	telephone instruments (landline/cellular)
Sampling	16 kHz, 16 bits/sample, then decimated to 8 kHz, 8 bits/sample (A-law)	8 kHz, 8 bits/sample (A-law)
Passphrase prompting method	visual prompts	aural prompts (synthetic speech)
Number of digits	5	4
Collection of name and digit sequence during test	single utterance	separate utterances
Turn-taking	no system prompt before first attempt; graphical indication of when system expects user input	system prompt before every expected user utterance
Session start	optical sensor	telephone call
Authentication during enrollment	30 minute time window from activation via web page	7-digit code and two hour time window

Other purposes are related to the data collection for scientific purposes:

- Client and impostor subjects can see how many sessions are expected from them and how many they have completed, together with statistics on recorded sessions. Impostor subjects are provided with a list of possible target speakers. For further encouragement subjects could also access group statistics of all users (like that provided to system administrators).

- Data collection supervisors are given an overview on what subjects have not yet completed their expected number of sessions, etc., in addition to all the statistics provided to system administrators. This function was very useful during data collection for issuing reminders to subjects. Statistics and results of annotation is also given.

Table 5.2: Digit sequences collected from each subject during enrollment.

Item	Sequence	Item	Sequence
1	3 5 6 0 2	6	6 9 4 1 3
2	7 6 3 2 4	7	2 1 8 5 7
3	9 3 0 4 6	8	0 8 1 7 5
4	8 7 2 9 0	9	4 0 9 6 8
5	1 2 5 8 9	10	5 4 7 3 1

5.5 Enrollment

The use of speaker verification requires clients to enroll. During an enrollment session, the PER system collects speech from the enrolling client (the enrollee) and creates a target model for that client.

The system is set to collect one valid repetition of each of ten items per subject with a proper name and five digits in each item. The digit sequences, listed in Table 5.2, are the same across subjects. They were designed such that each digit occur exactly five times; exactly once in every position within the sequence; and never more than once in a given left-context or right-context. A repetition of an item is deemed *valid* if the on-line speech recognition includes the expected name and digits for that item in its N-best output. The system asks clients to repeat the same item until a valid repetition is found and before moving on to the next item.

To avoid users being held up by repeating a particular item an unreasonable number of times in the event that the speech recognizer repeatedly fails to produce the correct hypothesis, they are offered to skip an item after every sixth attempt on the same item. A user is allowed to skip up to two of the ten items in this way. This skip-possibility was utilized by a few clients as presented in Section 10.3.1 (p. 195), thus the number of collected items per client varied between eight and ten.

Before using PER for the first time, clients have to enable their enrollment via the system's web interface. By doing so, they are given a time window for enrollment of 30 minutes for gate enrollment and 2 hours for telephone enrollment. Access to the intranet is protected by standard user name plus password login that constitutes the main mechanism for authenticating clients at enrollment. For the gate version of the system, it is the only authentication mechanism. It was judged as sufficient since the client also have to be physically present by the gate within the allocated time window. To enroll with the telephone version of PER, clients also have to enter a seven-digit authorization code at the beginning of the call. The code is issued by the web interface and presented to the client on an enrollment sheet that also includes the ten items to speak during enrollment. An enrollment sheet is not needed with gate enrollment since enrollment items are presented on the display.

5.6 Speech recognition

The automatic speech recognition (ASR) component of the system is based on the Starlite decoder (Ström, 1996) and acoustic models trained on the Swedish landline FDB5000 SpeechDat database (Elenius, 2000).

Acoustic models are state-tied triphone models created using the COST249 reference recognizer framework (Lindberg et al., 2000). Each triphone is modeled by three states and a mixture of eight Gaussian terms per state, with a total of 7623 states. The data set used for training the acoustic models is described in Section 10.2.1.

Input features are specified by the reference recognizer framework. They are 12-element MFCCs plus the 0'th cepstral coefficient and their first and second order deltas. MFCCs are similar to those used in the speaker verification system except that the filter bank has 26 filters spaced between 0-4000 Hz (cf. Section 3.2.1). Energy normalization and cepstral mean subtraction are not used.

The decoder uses a two-pass search strategy: a Viterbi beam-search followed by an A* stack-decoding search. A number of class-pair grammars are used to simulate a dialog-state dependent finite-state grammar. Output is an N-best list with up to 10 hypotheses, each specified by a word sequence and start and end times of each word segment. The application selects one hypothesis per utterance to be used as the segmentation of the utterance by the speaker verification system. The hypothesis is selected based on the knowledge of what the claimant is supposed to say, as the hypothesis with the highest score whose text matches the expected text. If there is no hypothesis with the expected text, the dialog system rejects the utterance and prompts the claimant for a new one. The system always knows what digits to expect from the claimant, since during both enrollment and test, digit sequences are prompted to the user.

5.7 Speaker verification

The speaker verification component of the PER system is a score-level combination of an HMM-based subsystem and a GMM-based subsystem. The entire speaker verification system was described in detail in Chapter 3.

Part III

Assessment methods

Chapter 6

Corpora

6.1 Introduction

This chapter describes two speaker verification corpora created by the author: the Gandalf and the PER corpora. Descriptions include the presentation of design criteria, recording procedures, text material and statistics on subjects and sessions. Enrollment and test sets used in this thesis are defined. Some of the more detailed information on the corpora, including additional enrollment and test sets of potential interest for future users, are presented in appendices and serve as documentation of the corpora for future reference.

The author also contributed to the creation of the Polycost and the VeriVox corpora. The scientifically interesting part of these contributions consisted in defining data sets for speaker verification experiments, and running such experiments. Some of this work is also summarized in this chapter. As a follow-up to our work on the public Polycost corpus, an overview on published results world-wide where this corpus has been used is given.

6.2 The Gandalf corpus

6.2.1 Introduction

The Gandalf corpus (Melin, 1996) was designed for research on speaker verification in telephony applications in the Swedish language and was recorded during 1995-1996. Before Gandalf, there was no large-scale speech corpus available that was recorded in Swedish and was suitable for experiments in speaker verification. The main ASV corpora publicly available[1] were YOHO, KING and SPIDRE, and they were all American English corpora.

The three main design criteria for Gandalf were

[1] available from the Linguistic Data Consortium, LDC, http://www.ldc.upenn.edu/

1. to include both telephone line variation and intra-speaker variation,
2. to allow for a comparison between ASV-systems with different text dependence, and
3. to enable an analysis of the significance of effects from different sources of variation in the speech signal.

The motivation for these criteria will now be outlined.

Some ASV corpora have been designed to cover either (long-term) variations in the speaker (e.g. YOHO (Campbell, 1995)) or telephone handset variation (e.g. HTIMIT and LLHDB (Reynolds, 1997b)). Reasons for including only one of the two are to isolate one source of variation in experiments performed on the data, and to limit the size of the corpus. Other corpora have been designed to cover both types of variation, e.g. a LIMSI/CNET corpus (Gauvain et al., 1995; Lamel and Gauvain, 1998), the CSLU Speaker Recognition Corpus (Cole et al., 1998), and the AHUMADA/Gaudi corpus (Ortega-Garcia et al., 2000).

The first half of Gandalf includes both types of variation, while the second half focuses on long-term speaker variation. In order to enable a separation of variations due to speaker and handset in the first half, subjects make half of the calls (every second call) from a "favorite handset", and the rest of the calls from different handsets in different environments. In the second half of the corpus, all calls (with some exceptions as detailed below) were made from the favorite handset.

Systems with different text dependency may differ in many aspects, such as user acceptance, system complexity, resistance against impostor attempts with e.g. tape recordings, and pure verification performance, where the last aspect is the main target for experiments with a speech corpus. To enable a comparison between these systems, the corpus should allow for testing of each system during equivalent conditions, which is made easier if the respective kinds of text are available in each call.

Most investigations on ASV-systems have been quantitative. The general characteristics of a corpus are described and results are summarized in overall error rates. Such experiments will not answer questions about what happens if a user gets a cold, if he calls from a mobile phone, if there is significant background noise, etc. In order to take the analysis further, a more detailed assessment of the ASV-system is needed. Such an assessment is possible if more sources of variation are documented in the corpus, and the results from system tests are correlated with this information. An example of such detailed analysis is that made during the NIST speaker recognition evaluations 1999, where results were correlated with pitch differences between enrollment and test, same-line vs. different-line tests, and same vs. different transducer type (Martin and Przybocki, 2001).

The Gandalf corpus has been used by the author in several papers (Melin, 1996; Lindberg and Melin, 1997; Melin, 1998; Melin et al., 1998; Melin and Lindberg, 1999b) and in this thesis (mainly Chapters 8 and 9 and Section 10.2), by Jesper Olsen at Center for Personkommunication at Aalborg University in Denmark

(Olsen, 1997, 1998a,b,c) and in a number of MSc thesis projects at KTH (Gustafsson, 2000; Neiberg, 2001; Elenius, 2001; Olsson, 2002; Lindblom, 2003).

6.2.2 Data collection

This section describes the recording procedure and provides statistics on subjects, calls and texts included in the corpus. A description for an initial subset of the corpus was previously published in (Melin, 1996). This section includes most of that description, but covers the entire corpus and adds statistics on more aspects of the content. Some additional, more detailed, statistics are presented separately in Appendix B (p. 261).

All calls were recorded with an automatic procedure through an ISDN-line and stored as one utterance per file in the format used in the Euro-ISDN network. Thus, the sampling frequency is 8 kHz, samples are A-law coded and stored with 8 bits per sample.

6.2.2.1 Subjects

Subjects in the corpus were recorded as either clients or non-clients. Clients made multiple calls while non-clients made two calls each. Later in this chapter, when defining test sets on Gandalf data, each non-client subject is assigned a function as either impostor or background speaker.

The client subjects were mainly recruited among employees at KTH and Telia Research AB, and from their friends and families. Subjects for the non-client part were mainly recruited through the client subjects with the request that people with some similarity be recruited, such as blood relatives or someone with the same particular dialect.

The client part contains 86 subjects (48 male and 38 female) and the non-client part 83 subjects (51 male and 32 female) with age distributions as displayed in Figure 6.1. Age distributions within the two subject groups are similar to each other, but they both clearly differ from that of the Swedish population, included in the same figure. In particular, age intervals 21–30 and 46–50 are over-represented in the corpus.

The dialect distribution among client subjects is heavily biased towards the Stockholm region, with 52 of the subjects coming from Stockholm and 12 coming from the nearby city of Eskilstuna. The remaining 22 subjects speak various dialects as shown in Table 6.1.

Table 6.2 shows statistics on smoking habits in subjects as stated by subjects themselves on a subject response sheet (at the beginning of the collection period). Two client subjects (`M045` and `F074`) and three non-client subjects (`M145`, `M164` and `M191`) indicated they smoked more than 10 times/day. Corresponding statistics for the Swedish population, also included in the table, shows that daily smokers are under-represented in the corpus.

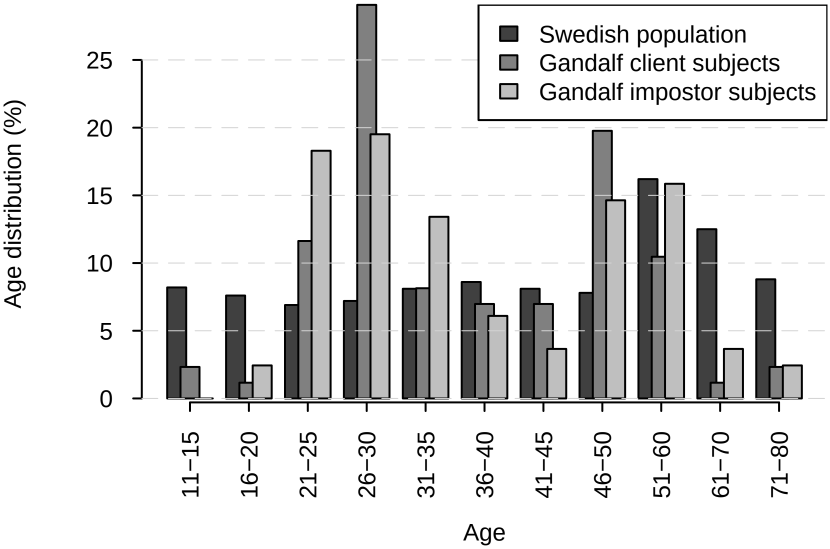


Figure 6.1: Age distribution, at the start of the recording period, among the 86 client subjects and 82 of the 83 non-client subjects (age information is missing for one female non-client subject). Distribution for the Swedish population between ages 11 to 80 is also included (Statistics Sweden (SCB), 2004). Note that the three right-most age intervals span 10 years each while the others span 5 years per interval (this is the resolution at which ages are known for Gandalf subjects).

Table 6.1: Dialectal distribution in the client subject group as judged by listening. Most of the subjects lived in the Stockholm region by the time of the recording period.

Region/dialect	No. of subjects
Stockholm	52
Eskilstuna	12
Norrland	7
Småland	4
Finland-Swedish	4
Göteborg	3
Foreign accent	3
Skåne	1

Table 6.2: Smoking habits among client and non-client subjects as indicated on subject response sheets, together with the corresponding statistics for the Swedish population year 2004 (ages 16–84; Statistics Sweden (SCB), *Undersökningarna av levnadsförhållanden (ULF)*).

Smoking	Subjects		Swedish
	client	non-client	population
No	81%	80%	73%
Occasionally	13%	12%	11%
Daily	5.8%	7.3%	16%

Table 6.3: Educational level among client and non-client subjects as indicated on subject response sheets, together with the corresponding statistics for the Swedish population year 2004 (Statistics Sweden (SCB)).

Education	Subjects		Swedish
	client	non-client	population
Compulsory school	7.0%	2.4%	25%
+ upper secondary school (or similar)	12%	23%	46%
+ university (or similar)	81%	74%	29%

Table 6.3 shows statistics on the educational level among subjects together with corresponding statistics on the Swedish population. A comparison shows that persons with university-level education are heavily over-represented in the corpus.

Table 6.4 shows how many groups of subjects in the corpus are related by blood. Table B.5 (p. 266) lists the exact relations in terms of subject numbers. To get an idea of how subjects perceive their own and their relative's voices, each subject was asked *"Do you think that your voices sound alike (for example, does someone usually mistake you for the other when you talk on the phone)?"* and had to respond by checking one of "no", "maybe a little" or "yes, a lot". Subject groups where at least one of the subjects responded by the third option have been underlined in the Table B.5 and were counted separately in Table 6.4.

6.2.2.2 Client calls

Two types of calls were recorded: enrollment calls and test calls. Enrollment calls were longer in order to collect enough speech material for enrollment into an ASV-system. The duration of the entire call was about 7 minutes per enrollment call and 2.5 minutes per test call.

The recording of client subjects were done in three parts.

Part 1 included three enrollment calls and 14 test calls; one call per week during four months. The series of calls started with two enrollment calls (denoted as `1` and `2`), one from the favorite handset and one from another handset. Then the test calls (calls `3`–`16`) were made with every second call from the favorite handset

Table 6.4: Number of groups with various types of relation by blood among subjects in the Gandalf corpus. In groups in the "alike" column, at least one of the subject indicated they have similar voices. Table B.5 (p. 266) lists the exact relations in terms of subject numbers.

Relation	Subject groups	
	"alike"	other
Identical twins	2	0
Siblings	5	9
Parent–child (same gender)	6	17
Parent–child (different gender)	0	9
Cousins (same gender)	0	8

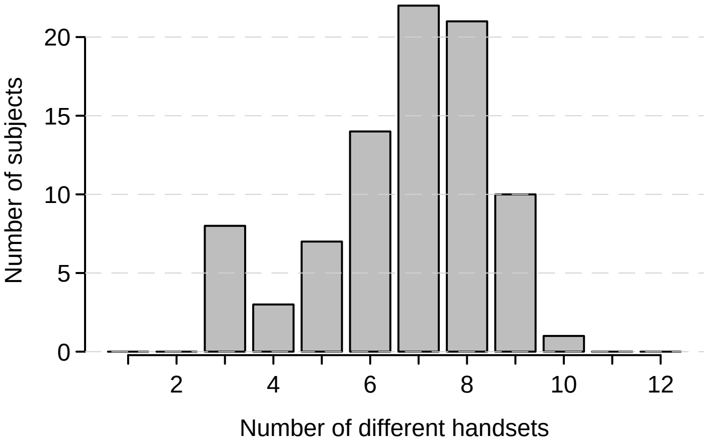


Figure 6.2: Histogram showing how many client subjects called from a certain number of distinct handsets (including the favorite handset).

and the rest from other handsets. Finally, the third enrollment call (denoted as 99) was made, approximately four months after the first enrollment call and again from the favorite handset. Figure 6.2 shows statistics on the number of handsets used by subjects, and Table 6.5 shows the distribution of those handsets over handset type. As many as 82 of the subjects completed all the 17 calls (remaining subjects completed 4, 7, 14, and 16 calls each). The total number of calls in Part 1 is 1435, corresponding to approximately 30 hours of speech.

Part 2 included seven test calls (denoted as 17–23) from the favorite handset with a one-month interval between the calls. 67 of the subjects from Part 1 volunteered to Part 2, and 59 of them (35 male and 24 female subjects) completed the seven calls. With Part 2, intra-speaker variation during up to 12 months has been included in the corpus. Part 2 contains 439 calls with 8 hours of speech.

Table 6.5: The number of handsets of each type used by client and non-client subjects, and the total number of calls made from a handset of the respective type in the client non-favorite handset section and the non-client call section.

	client calls			non-client calls	
	favorite	non-favorite		enroll	test
Type	handsets	handsets	calls	calls	calls
Stationary, button	84	259	394	69	40
Stationary, dial	1	43	63	10	18
Cordless	1	27	44	1	10
Mobile, GSM		58	81	1	8
Mobile, NMT		17	25		4
Pay phone, card		59	61	1	2
Pay phone, coin		12	14		
Speaker phone		6	7		
ISDN-phone		6	7		

Part 3 included five test calls (denoted as 24–28) also from the favorite handset and with a one-month interval between the calls. 49 subjects volunteered to Part 3, and 42 of them (25 male and 17 female subjects) completed the five calls. With Part 3, intra-speaker variation during up to 18 months has been included in the corpus. Part 3 contains 232 calls.

While Part 1 included calls both from the favorite handset and from other handsets to allow mixed studies of speaker and handset variability, Parts 2 and 3 focused on the speaker variability by including calls from the favorite handset only. However, eleven subjects stopped using the handset designated as their favorite handset during the collection of Parts 2 or 3 because it was no longer available to them for some reason. Table B.6 (p. 266) lists those subjects and during which calls they used another handset. The issue here is that these calls were not recorded from the same handset used during subjects' first enrollment call as they were meant to be to focus on speaker variability rather than handset variability. A total of 70 calls were made from a different handset than the original favorite handset.

With all three parts taken together, the client part contains 256 enrollment and 1850 test calls, with a total of approximately 40 hours of speech. Figure B.1 (p. 267) shows a complete histogram on how many client subjects recorded how many calls.

6.2.2.3 Non-client calls

Non-client subjects were recorded in two calls: one enrollment call and one test call (denoted as 101 and 102 respectively). The two calls were made from two distinct handsets and no specific time was requested for the interval between the calls. Normally, a non-client subject would have to be recorded only once, and the recording would be used only to make impostor attempts on the target identities of client subjects. However, recording one extra call per subject involves little additional effort, gives the possibility of making limited true-speaker tests with the non-client subjects, and allows each non-client subject to be recorded from two different handsets. The 83 non-client subjects recorded 165 calls (female non-client subject F157 only completed the enrollment call) with approximately 7 hours of speech.

6.2.2.4 More call statistics

Appendix B presents additional, more detailed statistics on the Gandalf corpus, including statistics on calling locations, international calls, illnesses, and (perceived) background noise.

6.2.2.5 Texts and prompting methods

The recorded phrases include short sentences and digit strings of various length. Some of the phrases are the same across calls while some are different. Some of the phrases are read from a script (visual prompting) and some are given aurally to the subjects by a voice prompt at recording time. In client calls 19–28, subjects were also asked to speak freely for 15 seconds. Table 6.6 shows the exact composition of phrases in each call. The scripted phrases are the same across subjects (except the 7-digit identification number).

Aural prompts were selected randomly at recording time from pools of prefabricated prompts. Two pools with 20 candidates each were used with 4-digit utterances, the first pool during client calls 3–18 and the second pool during client calls 19–28. With 5-digit utterances, which were collected only during client calls 19–28, a single pool of 20 candidates was used.

The first aural sentence prompt (for file RS01) in each call was selected from two pools of 20 "short" sentences each. Like with 4-digit utterances, the first pool was used during client calls 3-18 and the second pool during later client calls. The second sentence prompt (for file RL01) was selected from a pool of 20 "long" sentences during client calls 3–18. During client calls 19–28 two such "long" sentences were collected, now from a new pool.

All digit and sentence prompts except for the third prompted sentence of each call (for file RL02), were created using a rule-based formant synthesis TTS system (Carlson et al., 1982, 1991). Prompts for the third sentence were recordings of a human voice from a male speaker. Hence, prompts for the second and third sentences (RL01 and RL02) were created from the same sentence pool but with different voices.

Prompts created with the synthetic voice for sentences in the "short" pools were 2.3 s on average and contained on average 5.4 words per sentence. Sentences in the "long" pool were 2.6 s on average, with 6.6 words per sentence on average.

Non-client call 102 was designed to be as similar to early client calls as possible, but to also include content added to later client calls. Therefore, 4-digit sequences

Table 6.6: The number and types of phrases in different calls. PxR in a Call-ID column denotes R repetitions of P phrases.

Presentation			Call-ID								
Phrase	Tag[a]	TD[b]	1-2	3-16	99	17-18	19-23	24-28	101	102	Example
Visual prompt (manuscript)											
7-digit id.	ID	x	1x2	2	1x5	1	1	1	1x5	1	0 8 8 0 3 2 5
call No.	ID		1	1	1	1	1	1	1	1	det tredje
fixed sent.	FS	x	2x5	2	2x5	2	2	2	2x5	2	Öppna dörren innan jag fryser ihjäl.
varied sent.	VS		10	4	10	4	4	4	10	4	Filmen med badbilderna har fastnat i tullen.
1-digit	D1		50								2
3-digit seq.	D3	x		6		6	6	6		6	6 5 8
4-digit seq.	D4	x		4	4x5	4	4	4	4x5	4	7 9 4 1
5-digit seq.	D5	x	25		25				25		1 4 4 7 2
Aural prompt											
4-digit seq.	R4			2		2	2	2		2	2 9 5 4
5-digit seq.	R5						2[c]	2[c]		2[c]	3 4 0 8 9
"short" sent.	RS			1		1	1	1		1	Affären gick med stor förlust.
"long" sent.	RL			1		1	2[d]	2[d]		2[d]	Han såg mannen på höjden med kikaren.[e]
Free											
free speech	SP						1	1		1	
		Type[f]	E	T	E	T	T	T	E	T	
		Part	Part 1			Part 2		Part 3			
		Subject	Client						Non-client		

[a] The tag indicates the first two letters in names of recorded files.

[b] An x in this column indicates that the same text is used in every call, which allows for text-dependent tests.

[c] The prompt is presented once for file R501 and twice for file R502.

[d] One file (RL01) is prompted with a synthetic voice and the other (RL02) with a human voice.

[e] This particular sentence (*He saw the man on the hill with the binoculars*) was included in four versions with different combinations of word stress on the three nouns (mannen, höjden, kikaren).

[f] E and T denotes *enrollment* and *test* call, respectively.

(R401 and R402), the “short” sentence (RS01) and the first “long” sentence (RL01) were randomly selected from the corresponding first pools used with client calls. The second “long” sentence (RL02) was then picked from the second pool of “long” sentences while first 5-digit sequence (R501) was picked from the first 10 candidates of the 5-digit sequence pool and the second 5-digit sequence (R502) from the next 10 candidates in the same pool.

We finally want to mention four points in the design of sentence pools for aural prompting that have been useful, or may become useful in future experiments, for studying how the prompting method influences speaker responses:

The first point is that the strings used for scripted 4-digit sequences (“7 9 4 1”, “2 2 3 9”, “7 6 8 9”, and “0 3 5 1” recorded in files D401 through D404) were included in the first pool for picking aural prompts for 4-digit sequences. This allows for experimental studies comparing visual vs. aural prompting, such as these presented in Chapter 8.

Second, since we believed that repeating five aurally prompted digits would be more difficult than repeating four digits, and that playing 5-digit prompts twice would (partly) compensate for this difficulty, we collected the first 5-digit sequence (R501) with a single presentation of the prompt, and the second sequence (R502) with the prompt played twice[6]. Hence, comparing the two sets of recorded files shows whether our hypothesis was correct. One such comparison is made with respect to rates of generated speaking and recording errors in Table 8.2 (p. 161).

Third, as was already mentioned above, sentences were collected from a single pool using synthetic (RL01) and human (RL02) prompts. This design addresses questions like: Are subjects more likely to be influenced by a human voice than a synthetic voice? Does the reproducibility of a synthetic voice result in a normalization of the subjects’ speech?

Fourth, the phrase “han såg mannen på höjden med kikaren” (*he saw the man on the hill with the binoculars*) was included in the first pool of “long” sentences in four versions with different word emphasis patterns. This addresses the question of how much subjects replicate prosodic cues in aural prompts.

6.2.3 Meta information

Apart from audio files, the Gandalf corpus also contains a relational database (RD). The primary function of the RD application is to store all the available information about subjects and calls. It also has the potential for being used as a tool for analyzing results from ASV tests, though this possibility has not been exploited in practice.

6.2.3.1 Stored information

Information stored in the RD has been collected from the subjects through subject response sheets, call response sheets and through post-processing of the recordings.

[6] The manuscript indicated where a prompt would be played once or twice.

Subjects filled out one subject response sheet (SRS) plus one call response sheet (CRS) for each call. The SRS had questions about fixed characteristics, such as gender, age, dialect, smoking habits, and education, while the CRS had questions about call-specific conditions, such as the handset, the room where the call was made, background noise, and illness and other conditions that potentially could affect the voice. (Statistics on most of this information were given above or in Appendix B.) The response rate for the SRS was 100% for client subjects and 99% for non-client subjects (one sheet missing). Response rates for the CRS relative to the actually recorded calls were 99.5% (only 10 missing) for client calls and 99.4% for non-client calls (1 missing).

The post-processing involved manually checking the recorded files for correctly spoken phrases. Files where there is some deviation from a normal pronunciation of the text were sorted out for further annotation. Examples of such deviations are: repeated, extra, missing or mispronounced words.

6.2.3.2 Analysis tool

Given that many types of information about subjects and calls are stored in the RD, the RD application could be used for automated, detailed analysis of results from ASV simulations on the data. This possibility has not been exploited more than in terms of preliminary experiments reported in (Melin, 1996). The basic idea was to develop a report generator in the RD application that took as input score values from an ASV simulation on a list of verification tests. Reports would be based on correlations between information about subjects and calls on the one hand, and the simulation results on the other, and would address questions like: What happens to the false reject rate when a client always calls from his favorite handset compared to when he uses different handsets? Is a system robust to a particular type of background noise? How much larger is the probability of a false rejection when the client has a cold? How does the false reject rate change with time from enrollment? Is the probability of a successful impostor attempt greater if the impostor is a close relative?

The statistical significance of the answers to questions like these will depend on the number of occurrences of an investigated phenomenon and the number of errors made by the system under test. Hence, different questions will have answers with different statistical significance. For instance, the number of calls where a client subject has a severe illness[7] is low (51 calls, or 2.4% of client calls), and hence, conclusions on the influence of a cold will be relatively weak.

6.2.4 Data sets

With respect to subjects, the corpus was split into three disjoint sets: a development set, an evaluation set and a background set, by assigning each subject to one of

[7] see Appendix B, p. 264 for our definition

Table 6.7: Test set sizes for the Gandalf corpus with subject-split according to Division-1.

	Development set		Evaluation set	
	male	female	male	female
Targets	22	18	24	18
Impostors	23	18	58	32
True-speaker tests	508	419	543	342
Impostor tests				
same-sex	484	306	1368	558
cross-sex[a]	396	414	768	1044

[a] *male*-column shows number of cross-sex attempts against male targets, etc.

the three sets. This particular split is referred to as *Division-1*. Details on how the assignment was made and lists of subjects in each group are given in Appendix B (p. 261). Division-1 was used in (Melin, 1998) (background and development sets only), (Melin et al., 1998; Melin and Lindberg, 1999b; Gustafsson, 2000; Neiberg, 2001; Olsson, 2002; Lindblom, 2003) and in the present thesis (development and evaluation sets).

At the session level, one session per subject (the first test call; i.e. call 3) was used for impostor tests against all other targets, and all available test calls from a target were used for true-speaker tests. With one test drawn from each such session, we get the number of tests shown in Table 6.7. While the test set specification includes both same-sex and cross-sex impostor attempts, results should preferably be presented for same-sex tests only like in (Melin et al., 1998; Melin and Lindberg, 1999b).

For enrollment, there are three sessions to work with. Call 1 is from the favorite handset, call 2 from another handset, and call 99 is again from the favorite handset. The latter call was nominally recorded after call 16, i.e. after several test sessions. These sessions can be combined to form enrollment sets with various properties. Call an enrollment set NsMh*t, where NsMh indicates the composition of the enrollment data and t indicates the approximate amount of speech included in it, in minutes. The composition is defined by the number of sessions (N) from which data are drawn, and the number of distinct handsets (M) used in those sessions. Each enrollment session includes 25 five-digit sequences, 10 varied sentences and 5 repetitions of two fixed sentences (see Table 6.6). The digits correspond to approximately one minute of speech (0.5 s per digit) while the varied sentences correspond to approximately half a minute (3 s per sentence). The three enrollment sessions can hence be used to compose many enrollment sets. The sets used in this thesis are listed Table 6.8, while Table B.2 (p. 263) lists all Division-1 enrollment sets that have been defined.

At the file level, several test sets can be combined from the files included in each test session (Table 6.6). Test sets used in this thesis are given in Tables 6.9. An

Table 6.8: Gandalf enrollment sets used in this thesis. Sets are defined on five-digit sequences (d5) and fixed sentence (fs0n, with $n \in \{1, 2\}$). Files are specified on the format session/filename. Refer to Table B.2 (p. 263) for a complete list of defined Division-1 enrollment sets.

Composition	Phrase	t	Files	Comment
1s1h	d5	1	001/D5{01-25}	
		0.5	001/D5{01-12}	
		0.3	001/D5{01-07}	
2s2h	d5+fs0x	1	001/D5{01-05}, 002/D5{06-10}, 001/FS0*n*{01-05}), 002/FS0*n*{01-05})[a]	used for dedicated PER development

[a] $n = 1$ for half the targets and $n = 2$ for the other half

Table 6.9: Gandalf test sets used in this thesis. For an extensive list of test sets defined for Division-1, refer to Tables B.3 and B.4 (p. 264).

Test set	Text	Prompt	Files	Comment
2d4	8 digits	visual	(D401, D402) or (D402, D403) or (D403, D404) or (D404, D401)	the same digit sequences for a given target across all calls
1fs+1r4-fs0x	1 sent., 4 digits	visual, aural	(FS01, R401) or (FS02, R402)	the same sentence across calls; digit sequences are picked at random; designed as development set for PER experiments

extensive list of test sets defined in Division-1 are given in appendix as Tables B.3 and B.4 (p. 264).

6.3 The PER corpus

6.3.1 Introduction

The PER corpus is the result of data collection during 2003–2004 from actual use of a speaker verification system, namely that in the PER system described in Chapter 5. It consists of recordings of proper names and digit sequences spoken in Swedish and is suitable for experiments in text-dependent speaker verification.

The main design criteria for the corpus were to support an evaluation of the performance of ASV in the PER application, and to allow a fair comparison between speaker verification in on-site vs. telephone use.

The first criterion followed from our goal to evaluate the PER system during live use. New speech data were needed to this end because we had no suitable data before. The Gandalf corpus contained only telephone data, while PER used a wide-band microphone mounted in a reverberant room. It contained visually and aurally prompted digits, but no proper names, and the available text-dependent non-digit phrases were sentences shared by all subjects. Finally, Gandalf was recorded using a tape-recorder metaphor, where subjects were speaking to a machine without any feedback on how they were speaking, and we felt the difference to talking to a live ASV system could be important.

The second criteria came from a wish to relate results from on-site use of ASV to our previous research on its telephone use. Furthermore, we wanted to experiment with cross-condition enrollment and testing, where a client is supposed to enroll *once*, say by telephone, and then be ready to use his voice for authentication *anywhere*, be it a telephone or on-site application.

To allow a fair comparison between speaker verification in on-site vs. telephone use, the data collection was designed to include telephone data in parallel to on-site data. The telephone version of PER was thus created and a part of the test group was asked to make telephone calls in conjunction with their entry through the gate protected by the on-site version. Differences introduced in the telephone version with respect to the on-site version are motivated by the dissimilar prerequisites of the two cases as outlined in Section 5.3.

To allow for even better comparison between on-site data and the rather broad class of "telephone" data, and after noting that calling from a telephone rather than talking to the system on site potentially involves a change of room in addition to the recording transducer and channel, we decided to collect data in four separate *conditions*: through the microphone at the gate in the hall (stairwell), through a mobile telephone in the same hall, through the same mobile telephone from an office, and through a landline telephone from the same office. In this way, there is a change in one dimension at the time in a *room–transducer space* between the four conditions, and it should be possible to determine whether a difference in ASV error rate between any two conditions is mainly due to a change in the room (with associated background noise) or to the transducer (and associated transmission effects). The four conditions in the room–transducer space are labeled gate/hall, mobile/hall, mobile/office and landline/office in this thesis, and sometimes abbreviated G8[8], MH, MO and LO. They are illustrated in Figure 6.3.

While the PER system versions used for collecting data were not optimized for this particular application and the respective condition, a set of separate speakers were collected to serve as development data (background data) for creating optim-

[8] G8 for the downsampled 8 kHz version of gate data

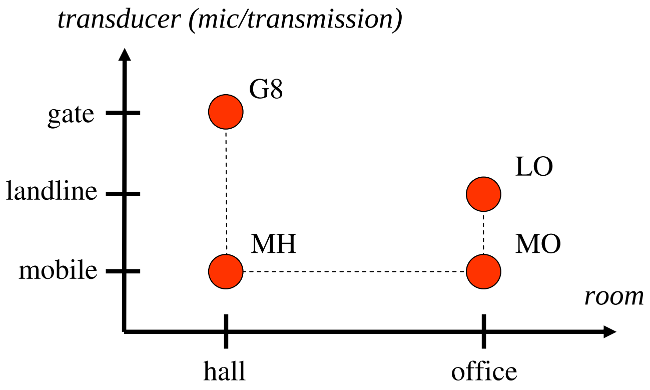


Figure 6.3: The four recording conditions in the PER corpus: gate/hall (G8), mobile/hall (MH), mobile/office (MO) and landline/office (LO).

ized systems for later simulation experiments. The so called background speakers were recorded in each of the four conditions.

6.3.2 Data collection

This section describes the recording procedure and provides statistics on subjects and sessions in the corpus.

Given the two main design criteria for the corpus, two somewhat conflicting goals were pursued in data collection: to collect as much data in the primary gate/hall condition as possible, and to collect as many parallel data as possible from all four conditions. To resolve this conflict, client subjects were divided into two groups, where one of the groups provided data in the primary condition only, while the other group provided data in all four conditions.

Subjects were interacting with the on-site or telephone versions of the fully automated PER system described in Chapter 5, that uses automatic speech recognition and speaker verification to recognize the content of spoken utterances and to verify users' claimed identities. If an utterance (or pair of utterances in the telephone case) was not found to contain a valid claim the system prompted the user to try again. Each system session allowed up to three attempts, but there was no limit on how many consecutive sessions a user was allowed to initiate. If a valid claim was found, the on-site version of the system physically unlocked the gate, while the telephone version did not. Both versions welcomed the user verbally. If no valid claim was found after three attempts, the system informed the user of this verbally.

Audio data from the telephone version of PER were recorded through an ISDN-line and stored as one utterance per file in the format used in the Euro-ISDN network. Thus, the sampling rate is 8 kHz, samples are A-law coded and stored with 8 bits per sample, the same format used in the Gandalf corpus. Audio data from the on-site version were recorded at 16 kHz sampling rate with 16 bits per sample (linear amplitude scale) and stored as one utterance per file. The same data was also decimated to 8 kHz sampling rate as described in Section 5.2 and stored in the same format as telephone data.

6.3.2.1 Subjects

Subjects are divided into two disjoint groups: the *test group* and the *background speakers group*. Those in the test group have been assigned one or both of the functions *client* and *impostor*. As clients they are further divided into *group L* (limited) and *group E* (extended) with respect to how much effort they were willing to spend as subjects. The main difference between the tasks of client subjects in the two groups is that group E provides data in all four conditions and group L only in the gate/hall condition.

Out of 56 subjects who volunteered to the client group and attempted to enroll to the system, 54 (16 female and 38 male) succeeded to enroll[9]. They were all students or staff from the Department, with the age distribution shown in Figure 6.4 together with the age distribution of the Swedish population. Like in the Gandalf corpus, the age distribution in client subjects in the PER corpus has two pronounced peaks at around 30 and 50 years of age. Among both male and female client subjects (who succeeded to enroll), half had previously been assigned to group L and half to group E.

Background speakers were recruited mainly from students and staff outside of the Department. 51 male and 28 female background speakers were recorded. While subjects in the test group used their own names, background speakers were assigned alias names. Alias names were chosen through the following procedure with the goal of including the most common Swedish names based on name statistics from Statistics Sweden (SCB) as of December 2002.

Starting from a list of the 100 most common family names in Sweden, 21 redundant names were removed. They were either homophones (e.g. Carlsson-Karlsson), phonetically similar (e.g. Jonsson-Jansson, Peterson-Petterson, Jonasson-Johansson), or substrings of other names (e.g. Ström-Strömberg). The order of the remaining 79 names was then randomized. The first 50 were then combined with a male first name and the remaining 29 with a female first name as described below. The 79 family names cover 31% of the Swedish population.

First names were processed similarly to family names, except they were used in frequency order. Starting from the 100 most common male (female) first names, 10 (13) were removed because they were homophones of other names in the list, or phonetically similar. The frequency count of a removed name was added to the frequency count of the similar name kept in the list. Names were then re-ordered according to adjusted frequency counts and assigned to subjects in that order.

[9] for results related to the enrollment process, refer to Section 10.3.1 on p. 195

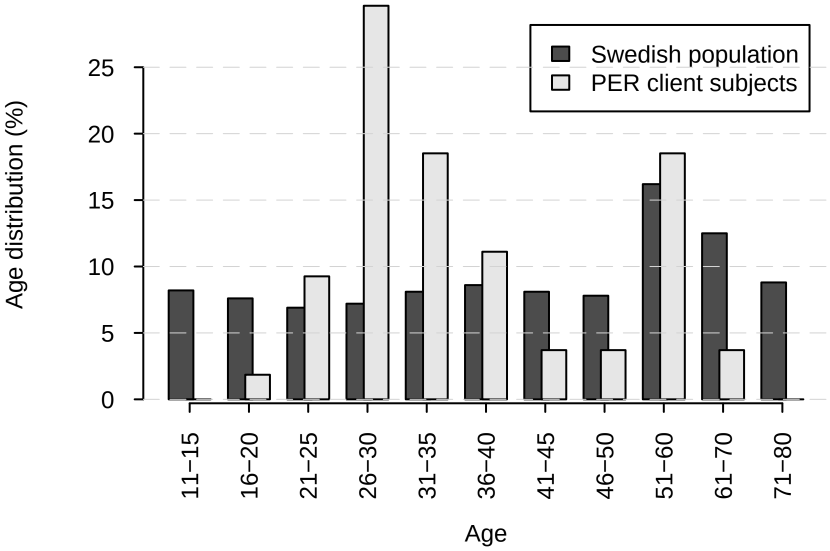


Figure 6.4: Age distribution, at the start of the recording period, among the 54 client subjects together with the distribution for the Swedish population between ages 11 to 80 (Statistics Sweden (SCB), 2004). Note that the three right-most age intervals span 10 years each while the others span 5 years per interval (for compatibility with the corresponding figure for the Gandalf database, Figure 6.1, p. 92).

The 51 male names used by a background speaker and seven corresponding similar names cover 51% of the Swedish male population, while the 28 female names plus two similar names cover 30% of the female population. More male first names than female first names were used since male subjects spoke only male first names, and vice versa, and more male background speakers than female ones were collected.

6.3.2.2 Recording conditions

As introduced above and illustrated in Figure 6.3, speech data were collected in four different conditions, referred to as gate/hall, landline/office, mobile/office and mobile/hall.

The primary condition was gate/hall. It was also the most naturally occurring condition of the four in that users had to pass through the gate to enter into the Department, and the PER system provided one of the three possible ways for employees to unlock the gate. The three telephone conditions were more artificial because the telephone version of the system did not give access to anything.

To allow comparison between the four conditions, parallel data were collected in *series* of sessions. A series consists of one session per condition recorded within a short time period with the same claimant speaker and the same claimed identity (target speaker) in all sessions. Subjects were asked to record sessions in a series as close as possible in time, preferably in immediate succession, and at the least to record them within the same day. They were also asked to vary the order of conditions between series.

In the gate/hall condition, every session by all claimants against any target is recorded through the same channel, i.e. with a single microphone, fixed amplifier gain, fixed recording level, etc. To establish a corresponding same-channel situation for telephone conditions (a single channel per target, but different channels for different targets), each client was asked always to use the same landline phone and the same mobile phone, and impostors were instructed to use the exact same telephone instruments as their target (to borrow phones from their target). These instructions were also followed in practice, with the exception that some subjects in the test group obtained a new mobile phone during the collection period and did not keep the old one. In these cases impostor attempts were made with the new telephone resulting in different channels between true-speaker and impostor tests, since most impostor sessions were recorded after the corresponding true-speaker sessions.

All telephone calls were made to a toll-free number to allow subjects to use their own mobile phone without being billed for their calls.

To balance out a potential bias from learning effects during enrollment in comparison between the gate/hall and landline/office conditions, half of the subjects within the test group and the background speakers group made their first enrollment session in the gate/hall condition and the second in landline/office, while the other half started with enrollment in landline/office. Enrollment in mobile conditions was always made after the other two enrollment sessions, however, allowing for a bias between mobile conditions on the one hand and gate/hall and landline/office on the other.

6.3.2.3 Client sessions

Clients in group L provided enrollment and test sessions in the gate/hall condition, plus an enrollment session in the landline/office condition. They were asked to provide at least 20 test sessions in the gate/hall condition during at least 15 different days. Clients in group E provided enrollment and test sessions in all four conditions. They were asked to provide 30 series of one session per each of the four conditions (for the definition of such series, see above) during at least 15 different days and then continue with at least 20 gate/hall sessions during different days.

During an enrollment session, the PER system collected one valid repetition of between eight and ten items per client and condition as described in Section 5.5 (p. 84). Each item consisted of the client's name and a string of five digits.

6.3.2.4 Impostor sessions

Since clients use their own name for verification, dedicated impostor sessions had to be collected. Impostor sessions against targets in client group L were collected in the gate/hall condition only, while impostor sessions against targets in client group E were collected as series of sessions in all four conditions. Impostor subjects were mostly the same people that also participated as clients. They knew most of their targets and were allowed to imitate the target's voice, though from listening through the recordings during annotation work, it turned out not many imitations were made in practice. Only same-sex impostor attempts were collected.

6.3.2.5 Background sessions

Background speakers made one complete enrollment session in each of the four conditions using office telephones and mobile phones mostly not used by subjects in the test group.

In each session they provided similar data as subjects in the test group (except they spoke their assigned alias name instead of their own name), plus five sentence items. The first sentence item was the same for all background subjects, "öppna dörren innan jag fryser ihjäl" (open the door before I freeze to death), while the remaining four were selected from a pool of 114 sentences such that one or two subjects of the same gender spoke the same sentence. Each subject spoke the same sentences in all four conditions. Sentences were 5 to 14 words long (average 7.4 words) and between 21 and 49 phonemes long using a prototypical transcription (average 33 phonemes).

6.3.3 Annotation

Recorded data were manually annotated on session and file level, where a file is intended to contain a single utterance. Annotations were made with a graphical tool dedicated to this task. Wherever possible, the tool provided initial values for annotation fields that the annotator could confirm or change to the appropriate value. Initial values were taken from output saved by the PER system during data collection into a relational database and XML session log files (cf. Appendix G for a specification of log file contents).

6.3.3.1 Session level annotation

Sessions were annotated with *claimant identity*, *claimed identity* and *session status*. Figure 6.5 shows an example screen shot of the graphical tool used for this purpose.

The claimant identity was determined by comparing audio and video data recorded during the session to reference audio and video data for known identities. The annotation tool provided an *identity browser* where the annotator could traverse a list of known identities and inspect reference data for each of them, and a session browser where the annotator could listen to recorded files and view recorded images from the selected session. The annotator could create new identity entries as new subjects were encountered.

The claimed identity was determined by listening to one or more audio files for the spoken name. An instance of the identity browser was used for this field too, mainly to provide the annotator visual feedback on the currently selected identity.

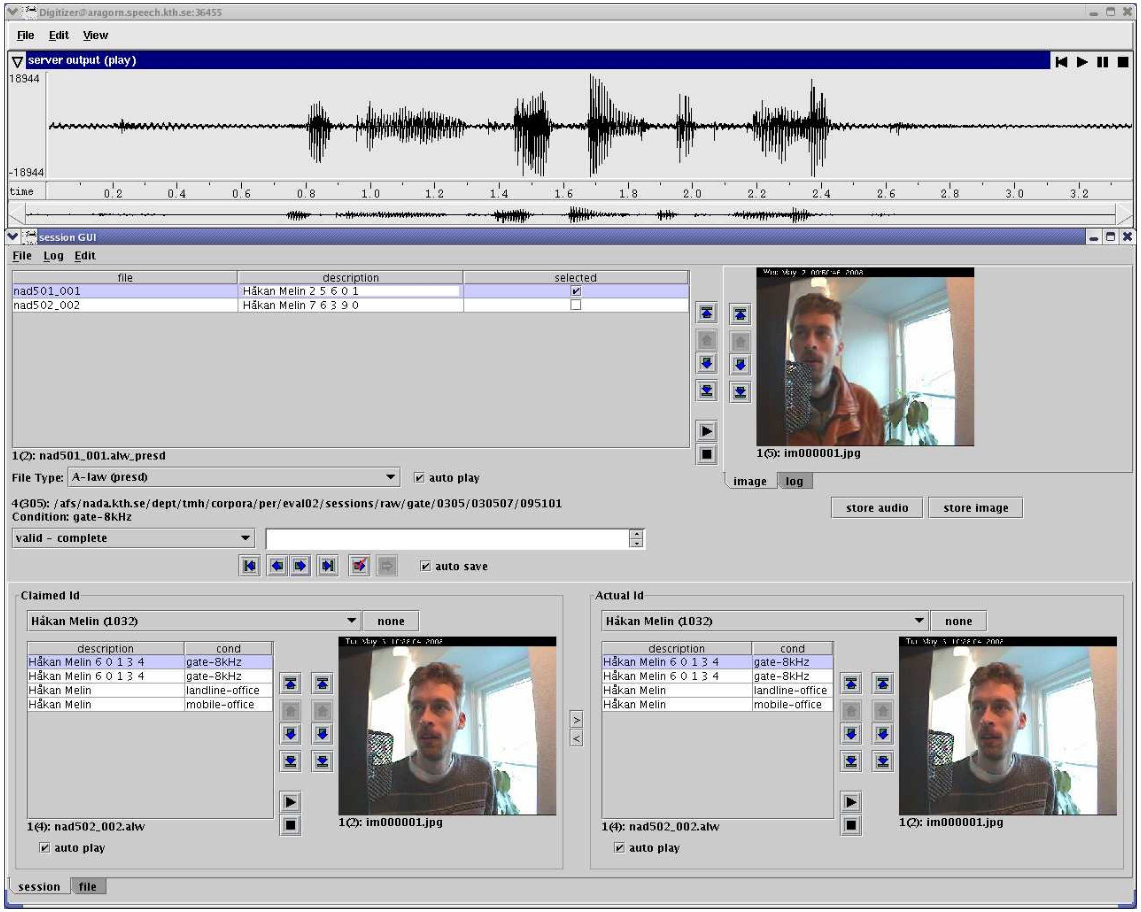
The default value selected for both identity fields when the annotator loaded a new session was the identity corresponding to the name recognized by the PER system, *a priori* assuming the session was a true-speaker session. Since the annotator tool showed three images simultaneously (one from the selected session, one for the currently selected claimant identity and one for the currently selected claimed identity) the annotator could very quickly verify current selections by comparing the three images and listening to one or more audio files.

Session status is a categorization with the main categories "valid" and "invalid". Valid sessions were further sub-categorized as "complete" or "incomplete". A session is considered valid and complete if it contains at least one file (pair of files for telephone sessions) with a name and the requested number of digits (five for the gate/hall sessions and four for telephone sessions). Remaining sessions were classified as valid but incomplete if a user was trying to make a (serious) attempt but failed to record at least one complete attempt (e.g. user spoke very slowly and the last words were truncated in the recording, or in a telephone session, the spoken name was never recognized as the name of an enrolled client); or invalid if a person was not judged by the annotator to make a serious attempt to use the system, if an unregistered identity was claimed, or if there was no recorded speech (a session was triggered by mistake).

6.3.3.2 File level annotation

Files were annotated with a *graphical transcription* and an optional *free-text comment*. If the speaker in a file was different from that selected for the session-level claimant identity (i.e. the speaker changed during the session), that file was also annotated with a *file-level claimant identity*.

Graphical transcriptions were based on SpeechDat conventions for transcription (Senia and van Velden, 1997). Standard conventions used were markers for stationary noise (`[sta]`), intermittent noise (`[int]`), speaker noise (`[spk]`), mobile phone-specific noise (`%word`), truncated signals (`~word` or `word~`), and mispronounced or truncated words (`*word`). To these were added a weaker marker for pronunciation errors used specifically with names (`&name`), and variants of intermittent noise for the particular noise occurring when the bar gate was opened (`[igo]`) or closed (`[igc]`). The `&`-marker was used with impostor attempts where the impostor pronounced a target's name in a different way than the target himself, and the difference was distinct enough to be captured by a phonemic transcription with word accent markers, but not so much different as to merit a `*`-marker for an incorrect pronunciation. Words labeled with `*` or `&` in the graphical transcription were transcribed phonemically in the comment field. The comment field was also



file	description	selected
nad501_001	Håkan Melin 2 5 6 0 1	☑
nad502_002	Håkan Melin 7 6 3 9 0	□

1(2): nad501_001.alw_presd

File Type: A-law (presd) ☑ auto play

1(5): im000001.jpg

4(305): /afs/nada.kth.se/dept/tmh/corpora/per/eval02/sessions/raw/gate/0305/030507/095101
Condition: gate-8kHz

valid - complete

☑ auto save

Claimed Id: Håkan Melin (1032)

description	cond
Håkan Melin 6 0 1 3 4	gate-8kHz
Håkan Melin 6 0 1 3 4	gate-8kHz
Håkan Melin	landline-office
Håkan Melin	mobile-office

1(4): nad502_002.alw

☑ auto play

1(2): im000001.jpg

Actual Id: Håkan Melin (1032)

description	cond
Håkan Melin 6 0 1 3 4	gate-8kHz
Håkan Melin 6 0 1 3 4	gate-8kHz
Håkan Melin	landline-office
Håkan Melin	mobile-office

1(4): nad502_002.alw

☑ auto play

1(2): im000001.jpg

Figure 6.5: The session annotation tool showing a true-speaker session. The "Session GUI" window contains the session browser (upper half), the identity browser for selecting the claimed identity (lower left), and the identity browser for selecting the actual identity of the claimant (lower right). The upper window holds a WaveSurfer widget for listening to the audio file select in the session browser, or looking at some graphical representation of it.

used to note cases of clearly altered voices in the speaker, such as a whispering or high-pitched voice.

6.3.4 Data sets

This section describes the enrollment and test sets used in this thesis (some additional data sets not used in the thesis are defined in Appendix C). A data set is defined by rules to select claimants, target speakers, sessions and files.

6.3.4.1 Notation

Data sets are denoted tix_c, where parameter t is E for enrollment sets, T for true-speaker test sets, I for impostor test sets and S for complete test sets (combined true-speaker and impostor test sets). Parameter c indicates the recording condition and takes values {G8,LO,MO,MH}. Parameter x indicates how files are selected from a given session. It is referred to as "accepted text status" in the set definitions below, and takes values {a,b} meaning

a: "accepted text status" means that both the target's name and the prompted digits were included in one of the hypotheses produced by the speech recognizer during collection;

b: "accepted text status" means that both the target's name and the displayed digits were spoken, as indicated by a manually verified transcription. To be more specific, the following conditions must be met by the transcription of a file (pair of files in telephone conditions): the complete name is included, but no modifier-labeled (~, *, %) repetitions of any part of the name; and the prompted digits are included in the given order, without modifiers, and with no other words in between. Note that names with the &-modifier are allowed, but not names with the *-modifier. Noise markers are allowed anywhere in the transcription.

Parameter i is simply an index number.

The notation introduced here is more general than required to cover the data sets actually used in this thesis. Specifically, we have always used text acceptance rule a in enrollment sets, while b was used in all test sets, and we have only used one single-condition test set and one condition-parallel test set per condition, all with index number 2. We have chosen to keep this notation for consistency with unpublished results and potential future experiments using other data sets. Other data sets, not used in this thesis, are defined in Appendix C.

6.3.4.2 Client enrollment sets

Based on data collected during enrollment sessions, two enrollment sets per condition c were defined using text acceptance rule a:

- E1a_c: (half session) the first five items from the last recorded and complete enrollment session from each client speaker under condition c; the first repetition of each item with accepted text status (approximately 15 seconds of speech per speaker).

- E2a_c: (full session) all ten items from the last recorded and complete enrollment session from each client speaker under condition c; the first repetition of each item with accepted text status (approximately 30 seconds of speech per speaker).

Sets E2a_c use the exact same data as was used during on-line enrollment into the collection system, while sets E1a_c can be used to simulate enrollment with only half of the speech data actually collected.

The G8 and LO client enrollment sets include 38 male and 16 female clients, while the MO and MH sets include 19 male and 10 female clients.

Corresponding enrollment sets have been defined on background speaker data as presented in Appendix C. Background enrollment sets also include sets with pooled speakers for training multi-speaker background models.

6.3.4.3 Single-condition test sets

Separate true-speaker and impostor test sets T2b_c and I2b_c are first defined for each recording condition c. Those are then combined condition-wise into the complete test sets S2b_c. In this thesis only a single complete single-condition test set per condition is used, and only for the gate/hall and landline/office conditions.

Common to both true-speaker and impostor test sets is that they contain no more than one attempt from any given session, and only from login sessions annotated as valid and complete that contain at least one attempt whose file level transcription meet the conditions of the b-criterion for "accepted text status". The true-speaker test sets include one attempt per session from all such true-speaker login sessions (no limit on the number of sessions per day or per target speaker). The impostor test sets include one attempt per combination of impostor speaker and target where the impostor speaker has recorded at least one session where (s)he claimed the given target identity. If there is more than one such session, the first one is used. Only same-sex impostor tests are used.

In sessions where more than one attempt satisfies the b-criterion for "accepted text status", the first attempt is used. Note that this selection depends on the manual transcription of recorded files only, and is independent of the results of speech recognition and speaker verification in the automatic PER system that collected the data. Table 6.11 below shows examples of files that were *not* included in the S2b_G8 test set because they did not satisfy the file-level selection criteria defined by the b-selection rule.

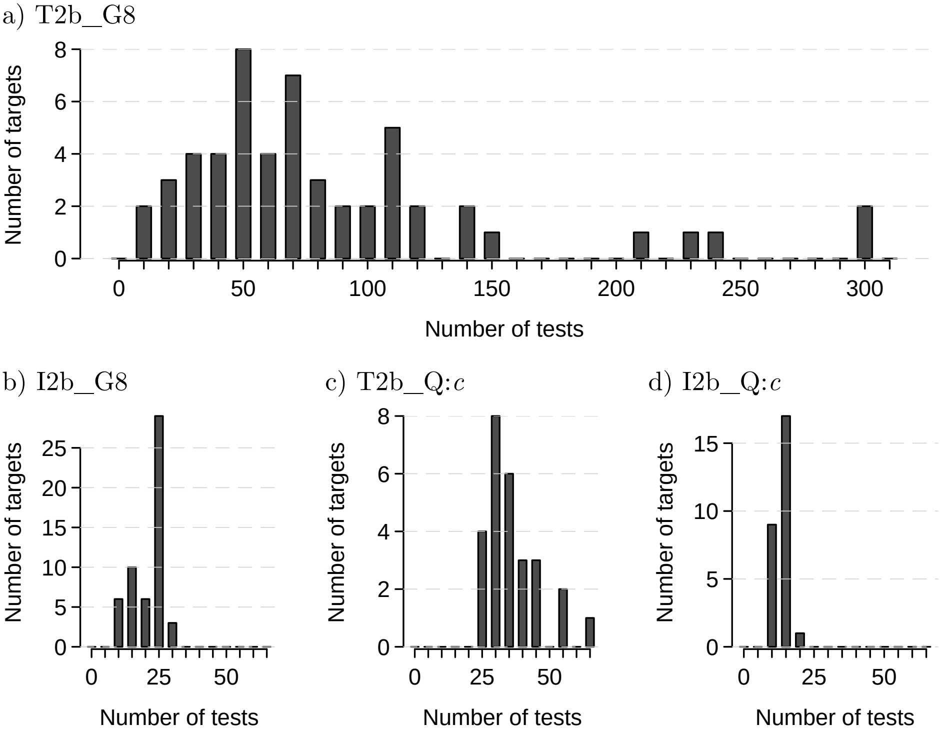
Table 6.10 shows the number of speakers and tests included in the PER test sets, including condition-parallel test sets defined below, while Figure 6.6 shows how tests are distributed over targets in the G8 true-speaker and impostor test sets.

Table 6.10: Test set sizes for the PER corpus. Number of subjects are specified as number of male / number of female subjects. All impostor attempts are same-sex attempts.

	Test set	S2b_G8	S2b_LO	S2b_Q:c
Test group	Targets	38 / 16	24[a] / 9[b]	19 / 8
	Impostors	76 / 22	37 / 16	35 / 16
	True-speaker tests	4643	1228	977
	Impostor tests	1121	422	393
Background speakers group	Speakers	51 / 28		

[a] Three of the 24 targets (M1014, M1023, M1101) have enrollment data but only a single true-speaker test each, and no impostor attempts. Two additional targets (M1122, M1127) have 7 and 20 true-speaker tests each, but no impostor tests.

[b] Target F1124 has 20 true-speaker tests but no impostor tests.


a) T2b_G8

b) I2b_G8

c) T2b_Q:c

d) I2b_Q:c

Figure 6.6: Histograms showing how many targets have how many tests in the gate/hall true-speaker and impostor test sets T2b_G8 and I2b_G8, and in the condition-parallel test sets T2b_Q:c and I2b_Q:c (per condition).

6.3.4.4 Condition-parallel test sets

To feature a comparison between conditions, a quadruple of condition-parallel test sets have been defined, each denoted S2b_Q:*c*, where Q is a short-hand notation for a list of the included conditions, Q={G8,LO,MO,MH}, and *c* is one of the four conditions. A condition-parallel test set is constructed such that there is always exactly one test from each of the listed conditions that correspond to each other in the sense that they were recorded near each other in time. Such a group of one test per condition is called "a series", like it was during the data collection. The file selection criterion specified in the single-condition test set is applied to each condition individually. If there is no selectable file for one or more conditions, no corresponding series is constructed. Sessions in the various conditions to be grouped into a series should have been recorded as close as possible to each other in time. They must at least have been recorded during the same day.

6.3.4.5 Test set statistics

Handset use Impostor subjects in the test group were asked to make calls in telephone conditions from the same telephone instruments used by the target speakers during their enrollment, and this request was well responded to. A comparison between A-numbers in test calls included in the true-speaker-part of the condition parallel test sets (S2b_Q:*c*) and the corresponding enrollment calls shows that 6.0% of calls in the landline/office condition and only 0.1% of calls from each of the mobile conditions were made from a different number. All but one of the different-number calls in landline/office were made by two subjects who changed their number and phone shortly after enrollment because they moved to other offices. `M1151` changed after the 8th of 31 calls and `F1160` after the first of 27 calls. After the change they consistently called from the same numbers, though the new numbers were different from the enrollment numbers. The remaining different-number calls (one per condition) were made by one female subject.

The corresponding proportions of impostor calls from a different number than the enrollment number are 23.9% in landline/office, 24.9% in mobile/office and 25.4% in mobile/hall. Most of these calls were made from a different number because either the target had left the Department before (four targets, 63% of different-number calls) or at the end of (two targets, 8% of different-number calls) the impostor data collection period, or because targets replaced their mobile phone during the same period (five targets, 15% of landline/office and 25% of mobile different-number calls). In all these cases, the enrollment mobile phone of the target was not available. Impostor subjects were then instructed to use non-enrollment phones in all telephone conditions. Moreover, in the landline/office condition, 9% of the different-number calls were made against target `F1160` from the same telephone that the target herself used in most of her true-speaker attempts. The remaining different-number calls (2-5 calls per condition) were made for other reasons, such as by mistake or by curiosity from impostor subject.

Note that a check for the same A-number in two calls doesn't guarantee that the same telephone instrument was used, and vice versa, but it is our belief there is a very good correspondence between telephone number and telephone instrument in our data.

Test file selection As an illustration of what definition b of "accepted text status" means in practice, Table 6.11 shows a categorization of transcription patterns for files that were skipped when selecting files for the S2b_G8 test set. The categories show what mistakes made by system or user caused files to be omitted from the test set. The majority of cases (62%) are omitted because one or more of the expected words are missing from the recorded file. This may be due to a speaker forgetting to say the digits (e.g. after the system responded to the previous attempt that it didn't perceive a name, then the subject often responded with only the name), speakers saying only their first name instead of the full name, or the system failed to record the entire (complete or incomplete) spoken utterance either because the speech detector pre-maturely signaled the end of the utterance or a programmed maximum recording time came to an end before the speaker finished the utterance. In many of these cases, missing words are due to a combination of a lacking capability in the system and unexpected behavior from the subject (e.g. a speech detector unaware of grammatical constrains in combination with very slow speech, a late start, or long pauses between words), and therefore the division of cases between "user mistake, not repaired" and "system mistake" is somewhat arbitrary.

Table 6.11: A categorization of transcription patterns for files that were skipped when selecting files for the S2b_G8 test set according to "accepted text status" alternative b. The total number of cases is 142 from 131 different sessions. The total number of sessions included in the S2b_G8 test set is 5764. Transcription patterns are constructed by replacing the first and last names of the target speaker with F and L, digits with D, markers for extralinguistic sounds with extral, and other noise markers with noise. Patterns in the example column are delimited by a comma.

Category	Cases	Fraction	Example patterns
User mistake, not repaired			
missing digits part (or truncated signal)	26	18.3%	F L
extra out-of-vocabulary words	9	6.3%	hallå F L D D D D D, ska man trycka nånstans eller F~
wrong digits	6	4.2%	F L D D D D D
wrong pronunciation of name or different name form	6	4.2%	F *L D D D D D, Alexander L D D D D D (F=Alec)
digits spoken in other language	3	2.1%	F L zero four eight six five
missing last name	3	2.1%	F D D D D D
speaker gives up	3	2.1%	noise *F extral
digits spoken as numbers	2	1.4%	F L noise DD DDD
mispronounced digit(s)	1	0.7%	F L *D D D D D
User mistake, with repair			
extra in-vocabulary words or fragments thereof	21	14.8%	*L L D D D D D, F L D *D D D D D, F L D D D D D D, F D D D F noise L D D D D D
System mistake (speech detection error or time-out)			
truncated signal or missing words	59	41.5%	~F L D D D D D, noise F L D D D D~, noise F~, F L D D
Other			
other	3	2.1%	

Table 6.12: 95% pre-trial confidence intervals for an observed false reject error rate on PER test sets given a "true" population error rate $p = 3\%$ or $p = 1\%$ and $N' = N^*/k = M\lfloor\bar{n}\rfloor^*/(1 + (\lfloor\bar{n}\rfloor^* - 1)\rho)$ independent tests for four choices of ρ. Intervals with lower limit 0.0 are one-sided confidence intervals, while others are two-sided confidence intervals.

					95%[a] confidence interval (%)			
test set	M[b]	N[c]	$\lfloor\bar{n}\rfloor^*$[d]	N^*[e]	$\rho = 0$	$\rho = 0.1$	$\rho = 0.2$	$\rho = 1$
$p = 3\%$								
T2b_G8	54	4643	50	2700	2.3–3.7	1.5–4.6	1.2–5.2	0.0–7.4[f]
T2b_Q:c	27	977	36	972	1.9–4.1	0.9–5.6	0.0–5.8	0.0–7.4[g]
$p = 1\%$								
T2b_G8	54	4643	50	2700	0.6–1.4	0.2–2.0	0.0–2.0	0.0–3.7[h]
T2b_Q:c	27	977	36	972	0.4–1.7	0.0–2.3	0.0–2.5	0.0–3.7[i]

[a] Due to the discreteness of the binomial distribution, actual confidence levels for intervals in the table vary between 95.4% and 98.3% (cf. Section 2.5.2).

[b] number of target speakers

[c] total number of tests in the set

[d] average number of tests per target (floored) after truncating right tail in Figure 6.6a

[e] adjusted total number of tests in the set ($N^* = M\lfloor\bar{n}\rfloor^*$)

[f] confidence level 97.7%

[g] confidence level 95.4%, i.e. same limit as with T2b_G8 but with lower confidence

[h] confidence level 98.3%

[i] confidence level 97.0%, i.e. same limit as with T2b_G8 but with lower confidence

6.3.4.6 Statistical significance

Table 6.12 shows 95% pre-trial confidence intervals[10] for observed overall false reject error rates for pooled target speakers on the single-condition gate/hall test set and each of the condition-parallel test sets for assumed "true" population error rates 1% and 3%, respectively. Confidence intervals are based on the assumptions made in Section 2.5.2 and four cases of choosing a value for the intra-speaker correlation coefficient ρ in Eq. (2.20) (p. 31). Here we use

$$N' = \frac{N^*}{k} = \frac{M\lfloor\bar{n}\rfloor^*}{1 + (\lfloor\bar{n}\rfloor^* - 1)\rho} \tag{6.1}$$

where M is the number of targets in the test set and $\bar{n}$ is the average number of tests per target. $\lfloor\bar{n}\rfloor^*$ is the average number of tests per target rounded downwards ($\lfloor\cdot\rfloor$) and adjusted for the fact that a few targets have very many tests in T2b_G8 as shown by Figure 6.6a. We (somewhat subjectively) chose $\lfloor\bar{n}\rfloor^* = 50$ for this test set and used the unadjusted $\lfloor\bar{n}\rfloor$ for the other test sets. Note that with (6.1), N' tends to M/ρ as $\lfloor\bar{n}\rfloor^* \to \infty$.

[10] Confidence limits in the table were computed with the `qbinom` function in the R software (`http://www.r-project.org/`).

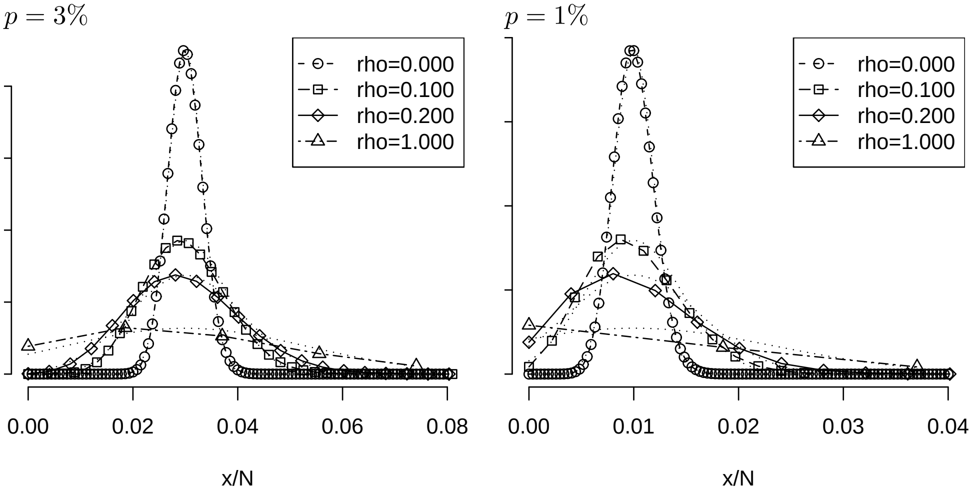


Figure 6.7: Binomial distributions used to compute confidence intervals for experiments on the T2b_G8 test set assuming "true" population false reject rate 3% or 1% and four choices of ρ. The normal approximation to each binomial is shown as a dotted line.

The first and the last case of choosing ρ in Table 6.12 are the two extremes in the assumption of partial dependence between observations discussed in Section 2.5.2.1, where we argued that the best choice of ρ should be somewhere in between the two extremes. Table 6.12 therefore also includes the cases $\rho = 0.1$ and $\rho = 0.2$. Unfortunately, we do not know at this point[11] which of these cases best describes reality. The table merely gives a perspective on the meaning of independence assumptions in terms of their effect on the length of confidence intervals. Figure 6.7 plots the binomial distributions from which confidence intervals in Table 6.12 for the T2b_G8 test set are computed, along with their normal approximations. Note that there are only a few points within the interesting range of observed error rate on the (discrete) binomial distributions with variances scaled according to $\rho > 0$, and resulting confidence limits are quantized by these points.

Table 6.13 shows example pre-trial confidence intervals for observed false reject rates for individual target speakers. It shows 95% confidence intervals for assumed "true" error rates 1% and 3%, respectively, and for a range of n (number of tests per target) spanning approximately (except for $n = \infty$) those values occurring for target speakers in the T2b_G8 test set as shown in Figure 6.6a. To calculate k of Eq. (2.17) we adopt (2.20) even though the latter was motivated by the beta-binomial distribution in the context of estimating an overall false reject rate from

[11] In Section 10.5.1 (p. 204) values for ρ are computed for post-trial experiments on the PER corpus, and Section 11.2.2 (p. 226) discusses the choice of ρ for pre-trial confidence intervals.

Table 6.13: 95% pre-trial confidence intervals for an observed false reject error rate for a single target speaker given a "true" error rate $p = 3\%$ or $p = 1\%$ for combinations of N and ρ_1. Intervals with lower limit 0.0 are one-sided confidence intervals, while others are two-sided confidence intervals.

	95%[a] confidence interval (%)			
$N = n$	$\rho_1 = 0$	$\rho_1 = 0.005$	$\rho_1 = 0.010$	$\rho_1 = 0.020$
$p = 3\%$				
∞		1.0–5.5 (96.5)	0.0–6.0 (96.9)	0.0–8.0 (98.3)
300	1.3–5.0 (96.0)	0.0–5.8 (97.1)	0.0–6.7 (97.5)	0.0–7.1 (96.3)
200	1.0–5.5 (96.5)	0.0–6.0 (96.9)	0.0–6.1 (95.2)	0.0–7.5 (96.9)
100	0.0–6.0 (96.9)	0.0–6.1 (95.2)	0.0–8.0 (98.3)	0.0–9.1 (98.3)
50	0.0–8.0 (98.3)	0.0–7.5 (96.9)	0.0–9.1 (98.3)	0.0–8.0 (96.2)
20	0.0–10.0 (97.9)	0.0–11.1 (98.4)	0.0–12.5 (98.9)	0.0–14.3 (99.2)
$p = 1\%$				
∞		0.0–2.5 (98.4)	0.0–3.0 (98.2)	0.0–4.0 (98.6)
300	0.0–2.0 (96.7)	0.0–2.5 (96.7)	0.0–2.7 (96.0)	0.0–4.8 (99.1)
200	0.0–2.5 (98.4)	0.0–3.0 (98.2)	0.0–3.0 (97.1)	0.0–5.0 (99.2)
100	0.0–3.0 (98.2)	0.0–3.0 (97.1)	0.0–4.0 (98.6)	0.0–3.0 (95.7)
50	0.0–4.0 (98.6)	0.0–5.0 (99.2)	0.0–3.0 (95.7)	0.0–4.0 (97.4)
20	0.0–5.0 (98.3)	0.0–5.6 (98.6)	0.0–6.3 (98.9)	0.0–7.2 (99.2)

[a] Due to the discreteness of the binomial distribution, actual confidence levels for intervals in the table vary as shown by parentheses in each cell (cf. Section 2.5.2).

multiple speakers with multiple tests. In (2.20) the coefficient ρ balances the correlation between speakers on the one hand and between trials from the same speaker on the other. It also widens confidence intervals to compensate for the distribution of individual false reject rates among targets in the test set. In the case of estimating a false reject rate for a single speaker, our ρ should correspond to correlation between single-speaker trials only, and possibly widen intervals because of inter-trial variation in the "true" underlying false reject rate, which we expect to be less than the corresponding inter-speaker variation. Thus we expect appropriate values for ρ in the single-speaker case to be smaller than in the multi-speaker case. To emphasize this difference we denote ρ in the single-speaker case as ρ_1. Thus, to calculate an equivalent number of attempts from the binomial distribution we use

$$N' = \frac{n}{1 + (n-1)\rho_1}. \tag{6.2}$$

Table 6.13 includes four choices of ρ_1, where $\rho_1 = 0$ corresponds to the case where all tests from a given target are independent and that the assumptions behind the error generation model motivating the use of the binomial distribution are assumed true (these assumptions were discussed in Section 2.5.2.1). Figure 6.8 shows plots of the binomial distribution (from which confidence intervals in Table 6.13

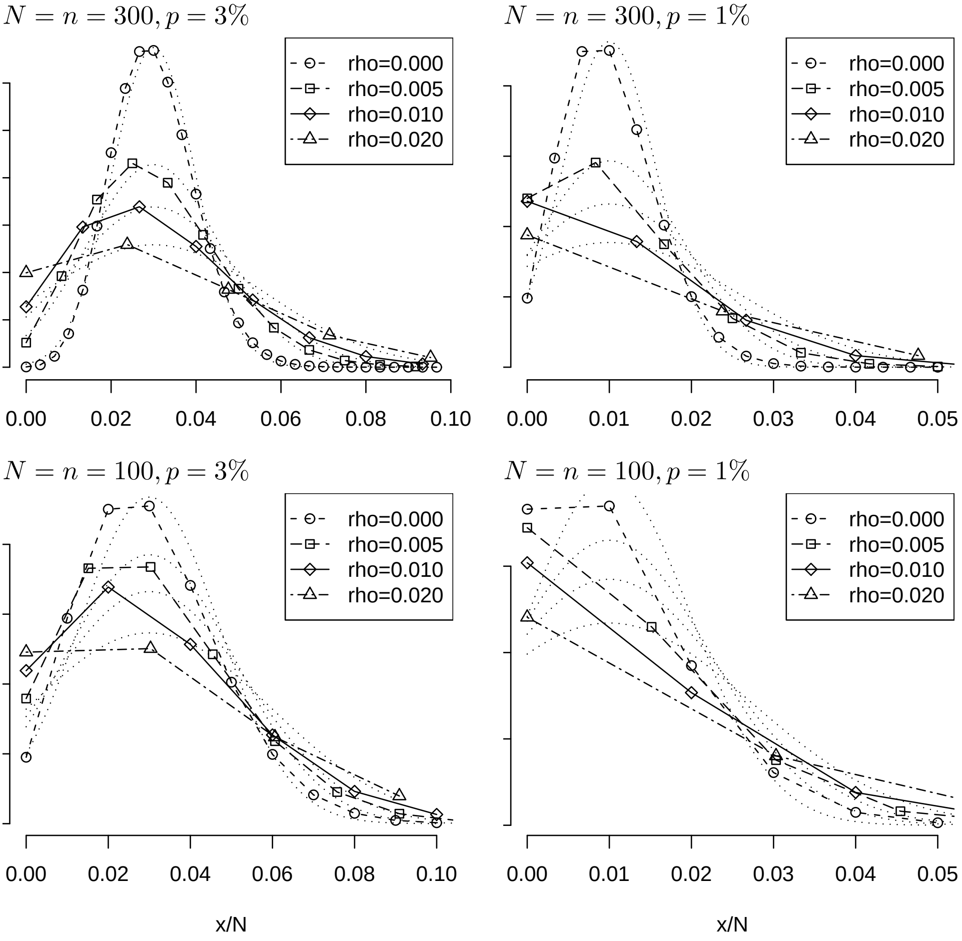


Figure 6.8: Binomial distributions used to compute confidence intervals for experiments on individual target speakers assuming "true" population false reject rate 3% or 1% and four choices of ρ_1. The normal approximation to each binomial is shown as a dotted line.

were computed) and their normal approximations for $n = 300$ and $n = 100$. Also for this single-speaker case, we don't know what values for ρ_1 are appropriate. Values for the table were selected through our prior belief and after studying distribution plots like those in Figure 6.8. Note that under our model (6.2), and assuming $\rho_1 > 0$, the width of confidence intervals is bounded from below by the value of ρ_1 no matter how many trials are observed for a target speaker, since N' tends to $1/\rho_1$ as $n \to \infty$. These bounds are shown in the table for $n = \infty$.

6.4 Cross-fertilization

Subjects listed in Table 6.14 participated both in the Gandalf collection and in the PER collection. The seven subjects listed above the dashed line have multiple (true-speaker) calls recorded in both corpora. There is a possibility to use recordings from those subjects for studying intra-speaker variation over very long time: more than 8 years. Both corpora contain aurally prompted 4-digit utterances, which could be used for such comparisons. A comparison is complicated by the fact that favorite handsets used by subjects in Gandalf were mostly *not* re-used in the landline/office condition of the PER collection, so some channel compensation method would need to be employed. The exceptions are Gandalf subjects `M014`[12], `M021` and `F089` who did re-use their Gandalf favorite handset during the PER collection.

6.5 The Polycost corpus and baseline experiments

Polycost (Hennebert et al., 2000; Melin, 1999) is a public[17] telephone quality speaker recognition corpus with 134 speakers and trans-European English and mother tongue utterances in 17 languages[18]. It was created within the framework of the European project COST250 during 1996-1999. The author's main scientific contribution to this corpus was in the definition of standard data sets for experiments. The four so called *baseline experiments* (BE) were initially defined in 1996 (Melin and Lindberg, 1996) and revised in 1999 (Melin and Lindberg, 1999a)[19] based on experience from using the initial formulation (Nordström et al., 1998).

Four baseline experiments were defined based on recordings from 110 subjects used both as target and impostor speakers:

- **BE1** text-dependent speaker verification using a fixed passphrase ("Joe took father's green shoe bench out") shared by all targets and known to impostors.
- **BE2** vocabulary-dependent speaker verification using sequences of 10 digits spoken in English. A fixed digit sequence, shared by targets and impostors, was used as test utterance. This particular digit sequence was not included in enrollment data.
- **BE3** text-independent speaker verification in target (and impostor's) mother tongue. Enrollment on unconstrained spontaneous speech and test on constrained utterances where subjects were asked to say their name, gender, town and country.
- **BE4** closed-set speaker identification task with the same enrollment and test utterances used in BE3. All 110 target speakers are used.

[12] same handset but different room and telephone line

[17] available through ELRA `http://www.elra.info/`

[18] nine languages if counting only those with five or more speakers each in the corpus

[19] also available online `http://www.speech.kth.se/cost250/`

Table 6.14: Subjects who participated in both Gandalf and PER. The seven subjects listed above the dashed line are those most likely to be useful for studying voice variation over long time since they have many telephone calls recorded both in Gandalf and PER.

Gandalf				PER (landline/office condition)			
id	time span[a]	calls[b]	function[c]	id	time span[a]	calls[d]	
						T	I
M010	9503–9608	26	dev[T,I]	M1032	0305–0403	149	18
M012	9503–9609	26	dev[T,I]	M1015	0305–0306	37	18
M014	9503–9609	26	dev[T,I]	M1005	0305–0309	46	2
F016	9503–9610	26	dev[T,I]	F1025	0305–0308	45	7
M021	9503–9608	26	dev[T,I]	M1003[e]	0305–0307	56	18
M086	9503–9609	26	eval[T,I]	M1010	0305–0310	42	18
F089	9503–9509	15	eval[T,I]	F1051[e]	0305–0312	28	7
M015	9503–9610	26	dev[T,I]	M1166	0403	0	1
F019	9503–9608	26	dev[T,I]	F1009	0310–0312	0	7
F031	9503–9612	26	dev[T,I]	F1031	0306	0	2
M033	9503–9607	25	dev[T,I]	M1002	0309, 0402	0	6
F070	9503–9610	25	eval[T,I]	F2034[f]	0403	0	0
M081	9504–9609	26	eval[T,I]	M2077[f]	0403	0	0
M085	9503–9608	26	eval[T,I]	M1044	0306	0	2
M087	9503–9608	26	eval[T,I]	M1045	0402	0	3
M092	9503–9607	26	eval[T,I]	M1016	0305	0	19
M177	9604	1	eval[I]	M2016[f]	0403	0	0
M190	9605, 9606	1	eval[I]	M1014	0306	1	2
F103	9601, 9604	1	bgr	F1000	0306	0	8
M110	9601	1	bgr	M1023	0305, 0312	1	4
M111	9602, 9608	1	bgr	M1013	0306	0	2
M112	9601	1	bgr	M1028[g]	-	0	0
M128	9612	1	bgr	M1034	0306	0	0
F192	9606	1	bgr	F1007	0305	0	3

[a] interval covered by recordings, denoted as *yymm* for start and end of interval, where *yy* is year (19*yy* or 20*yy*) and *mm* is month

[b] number of recorded test sessions

[c] how calls are used in the data sets defined in Section 6.2.4; *dev* for development set, *eval* for evaluation set, and *bgr* for background speaker set; *T* indicates subject is used in true-speaker tests and *I* subject is used in impostor tests

[d] number of sessions used in true-speaker (T) or impostor (I) part of test set S2b_LO

[e] Gandalf favorite handset is identical to handset used in PER landline/office true-speaker calls

[f] background speaker with enrollment sessions in all PER conditions

[g] gate/hall impostor subject only

Across all (revised) BEs, two files were used for enrollment, one from the first recorded session and one from the second session. Only same-sex and same-language impostor attempts were used, where "same-language" in the case of the utterances spoken in English by all subjects was defined as impostor and target having the same mother tongue. BEs 1–3 all consist of 664 true-speaker tests and 824 impostor tests, while only the true-speaker tests apply to BE4.

A set of 12 male and 10 female subjects uniformly selected from the major languages represented by target speaker were set aside for use as background speakers.

The differences in BE definitions between version 1 and 2 are in a reduction in the amount of enrollment data per target, in the restriction of impostor attempts to same-sex and same-language tests, and in the use of a target-independent threshold (rather than speaker-dependent thresholds) in *a posteriori* error rate measures, such as the EER. These changes were introduced to make the BEs more realistic.

Results from experiments on Polycost have been reported at COST250 meetings (Olsen and Lindberg, 1999; Melin, 1999) and at major international conferences. An overview of published results with the baseline experiments on speaker verification (BE1–3) is shown in Table 6.15. It is clear from this table that the revised BE definitions result in considerably higher error rates. Comparisons between results on Polycost and other corpora (Melin and Lindberg (1999b) using version 2.0 and Melin et al. (1998) using version 1.0; both compare Polycost results to results on Gandalf and the Dutch SESP corpus using the same ASV system), suggest the difficulty of at least the new BE2 is comparable to similar tasks on other corpora.

A large number of publications report results on speaker identification using variations of BE4 (Ambikairajah and Hassel, 1996; Altincay and Demirekler, 1999; Fatma and Cetin, 1999; Magrin-Chagnolleau and Durou, 1999, 2000; Magrin-Chagnolleau et al., 2002; Altincay and Demirekler, 2002; Katz et al., 2006a,b). It appears the definition of BE4 was ill-designed in the (too large) number of target speakers involved, since most users chose to use subsets of target speakers of different size, resulting in incomparable results.

To make the list of Polycost users known to the author complete, we also cite works where Polycost was used for speaker verification or identification experiments, but with their own experiment design (no reference to baseline experiments): Durou and Jauquet (1998); Ganchev et al. (2002); Mengusoglu (2003); Ganchev et al. (2004a,b); Siafarkas et al. (2004); Anguita et al. (2005); Ejarque and Hernando (2005). Finally, Ben-Yacoub et al. (1999) used Polycost for training background models.

In conclusion, we find the creation of the Polycost corpus as a public database a success in that it has been used at many sites by many researchers, and thus is likely to have contributed to the advance of the knowledge-base in the speaker recognition community. The small number of sites having implemented the baseline experiments to their full extent is a disappointment, however.

Table 6.15: Summary of Polycost results with baseline experiments.

BE	publication	EER (%)			comment
		global[a]	individual[b]		
		SS[c]	SS[c]	GBSI[d]	
Baseline Experiments version 1.0					
1	Nordström et al. (1998)	0.6	0.1	0.0	Nuance
	Gagnon et al. (2001)	[e]1.5			
	Nordström et al. (1998)	3.2	1.0	0.7	CAVE system
	Hernando and Nadeu (1998)			2.5	
2	Nordström et al. (1998)	0.4	0.1	0.1	GIVES (LPCC)
	Nordström et al. (1998); Melin et al. (1998)	1.5	0.3	0.2	GIVES (MFCC)[f]
	Nordström et al. (1998)	2.2	0.1	0.1	Nuance
3	Nordström et al. (1998)	11.0	6.3	4.2	Nuance
Baseline Experiments version 2.0					
1	Katz et al. (2006a)	[g]2.2			
	Nordström et al. (1998)	2.4			Nuance
	Melin et al. (1999)	12.8			COST250 refsys
2	Melin and Lindberg (1999b)	4.3			GIVES (MFCC)[f]
	Nordström et al. (1998)	4.6			Nuance
	Nordström et al. (1998)	5.1			GIVES (LPCC)
	Melin et al. (1999)	11.0			COST250 refsys
3	Nordström et al. (1998)	15.5			Nuance
	Melin et al. (1999)	15.7			COST250 refsys

[a] target-independent *a posteriori* threshold

[b] target-dependent *a posteriori* thresholds

[c] same-sex impostor attempts only

[d] cross-sex impostor attempts also included (exist in BE version 1.0 only); gender-balanced, sex-independent weighting of errors (Bimbot and Chollet, 1997; Bimbot et al., 2000)

[e] on subset of 91 out of 110 targets due to imposed restrictions on enrollment data

[f] same as the HMM system described in Section 3.3 but with background models trained on Polycost background speakers

[g] used four repetitions for enrollment instead of two as specified for BE1 v2.0

6.6 The VeriVox corpus

The VeriVox corpus (Karlsson et al., 2000, 1998) was recorded within a European project with the same name. The purpose of the corpus was to compare enrollment using neutral speech versus structured enrollment, where the latter refers to enrollment using a structured mix of six automatically elicited, voluntary speaking styles: Neutral, Weak, Strong, Slow, Fast and Denasalized. During a single session, each subject was performing a number of tasks implemented in a game-like computer environment. Between each task the subject was asked to read a list of digit sequences in a specified manner, to collect voluntary speaking styles. Involuntary speaking styles were collect *during* the tasks, for example while speaking in the presence of noise, speaking from memory at an increased rate due to time pressure and speaking while solving a task under high cognitive load (Karlsson et al., 2000). Speech with involuntary speaking styles was then used as the primary test material for speaker verification experiments. The contribution of the author in this project was the design and definition of speaker verification experiments.

Chapter 7

Robust error rate estimation

7.1 Introduction

This chapter focuses on the mathematical side of the second issue related to error estimation and introduced in Section 2.5: given a list of test trials, how do we estimate (technical) error rates efficiently? By efficiency we here mean a good trade-off between estimate quality and the number of trials needed for the estimation. Loosely, an estimate is good (of high quality) if it is "close" to the "real" quantity that we are trying to estimate.

The overall false reject rate (FRR) and the overall false accept rate (FAR) are conventionally estimated with a Maximum Likelihood (ML) method, given an *a priori* decision threshold, as defined by Eq. (2.11)[1]. For the purposes of this chapter, we will refer to error rates computed this way as *non-parametric FRR* and *FAR*, or FRRd and FARd with a suffix d (for data) to indicate that they are computed directly from observed data.

The EER and points on a DET curve are usually also computed with ML estimates directly from score data from test trials, though some interpolation may be used to compute the EER if there is no pair of FRR and FAR where the two are exactly equal. We will refer to an EER computed this way (from score data directly) as the *non-parametric EER*, or EERd.

An alternative error rate estimation approach is to first estimate some statistical models to represent score distributions for the true-speaker and impostor classes and then compute error rates from those models (Elenius and Blomberg, 2002). Assuming each of the underlying distributions can be appropriately described by a parametric distribution and the family of such parametric distributions is known, the advantage with this method is that the estimate of distribution parameters, and eventually the verification error rates, can be based on all score points, and not be biased by the score points around the estimate in question. A draw-back is of course that it is usually difficult to show with certainty that a selected family of

[1] p. 26

Table 7.1: Overview of error rates used in this chapter and how they are estimated. Items within parentheses list estimator types.

	data representation	
observation type	non-parametric	parametric
score	EERd (ML)	EERp (ML) FRRp (ML, MAP) FARp (ML, MAP)
decision	FRRd (ML, MAP) FARd (ML, MAP)	

parametric distributions is appropriate. We refer to this method as the *parametric score distribution method* (or simply *parametric method*), an EER computed this way as the *parametric EER*, or EERp, and correspondingly write FRRp and FARp for the parametric false reject and false accept rates.

In this chapter we first look at the normal distribution as a possible parametric model of true-speaker and impostor score distributions from both theoretical and empirical points of view. We then derive Bayesian MAP estimators for error rates using both the non-parametric method and the parametric method under the assumption that a MAP estimate is more efficient than an ML estimate because it uses prior information about distributions of scores or error rates. Table 7.1 illustrates the different error rate measures and how they are computed in terms of observation type (score or decision), data representation (non-parametric or parametric), and the optimization objective used in deriving estimators (ML or MAP). We finally evaluate the Bayesian estimators on PER data (cf. Section 6.3) and compare them to the conventional ML estimators. For the evaluation we focus on the estimation of error rates for individual target speakers. In particular, we look at how error rates are distributed over speakers in the test group and how individual error rates change with time from enrollment.

For the empirical evaluation, three ASV systems are used. They are the same systems used in Chapter 10: two instances of our research system described in Chapter 3 (a baseline system and a retrained system, cf. Section 10.4) and a commercial system (cf. Section 3.1, p. 37). Properties of the research system are also used in our theoretical discussion of the appropriateness of the normal distribution for approximating score distributions.

7.2 Score distribution

To develop error rate estimation techniques based on the approximation of empirical score values by a statistical model, we first need to find an appropriate model, i.e. a family of parametric distributions. The normal distribution have been proposed in previous work (e.g. Lund and Lee, 1996; Surendran, 2001). Furthermore, DET plots are constructed to show straight lines when impostor and true-speaker score distributions are normal (Martin et al., 1997), and they often show fairly straight lines as long as plots are based on a sufficient number of test trials (e.g. Przybocki and Martin, 2004). The normal distribution also lends itself to easy mathematical handling. Hence, if the normal distribution is appropriate, it is a practical choice for our score distribution model.

7.2.1 Normal approximation

The central limit theorem states the sum of a large number of independent, identically distributed random variables is normally distributed. More general forms of the theorem also exist where variables are not required to be identically distributed, but other weaker conditions are required (Feller, 1968), for example that the variables are uniformly bounded. The score z for a single test utterance in our ASV system given by (3.23)[2] is based on the sum of a large number of frame level scores, in turn based on sums over log-likelihood ratio contributions from pairs of Gaussian mixture terms (cf. Eq. (3.6) for the HMM subsystem and Eqs. (3.20–3.22) for the GMM subsystem). Those frame scores are not identically distributed, but they should be uniformly bounded provided speaker model variance parameters are properly estimated (no very small variances). There is certainly a measure of statistical dependency between observation vectors, due to repetitions of phonemes within a test utterance, co-articulation effects, consecutive observation vectors originating from the same phone realization, etc., but in practice dependency effects can often be compensated for by counting a large number of dependent variables as equivalent to a smaller number of independent variables. For example, consider a typical PER test utterance “Håkan Melin 2 1 9 5 8” spoken as [hoːka mɛˈliːn ˈtvoː ˈɛt ˈniːˌɛ fɛm ˈɔtˌa] without pauses between words[3]. The utterance contains 13 distinct phonemes and 23 phoneme realizations. It was spoken by the author in 1.75 seconds and was thus represented by 175 observation vectors. The utterance score is based on a sum of 175 frame level scores, but the equivalent number of statistically independent variables behind the 175 frame level scores is maybe as low as 13, but more likely somewhere around three times that number. This may well not be a large enough number with respect to the bound on our summation terms to claim that the central limit theorem indeed applies to the utterance score values, but we assume for now the normal distribution is a fair approximation. We will present some empirical evidence to support this assumption below.

So far we discussed only the distribution of scores for a given pair of claimant (possibly an impostor) and a claimed identity. The true-speaker and impostor score distributions from which we estimate overall ASV system error rates are unions of utterance level scores from groups of client and impostor speakers, respectively. Such distributions are not necessarily normal, but should be well approximated by GMMs. For our discussion on the use of non-parametric vs. parametric error rate

[2] p. 47

[3] spaces in the transcription are included only to mark word boundaries

measures on groups of target speakers, we will still compare such score distributions to their normal approximations below.

7.2.1.1 Individual targets

Turning to the data, we start by inspecting true-speaker and impostor score distributions for individual target speakers using the retrained system tested on the S2b_G8 test set with target models trained on the E2a_G8 enrollment set (cf. the solid curve in Figure 10.1a). Figure 7.1 shows normal quantile plots (or normal QQ-plots) for the samples of true-speaker score values z in Eq. (3.23) for nine individual male target speakers. They are the nine targets with the largest number of true-speaker tests in S2b_G8, and the number of tests among these nine targets range from 305 tests for `M1032` to 118 tests for `M1047`. Each graph plots sample quantiles (ordinate axis) for a given target against quantiles of the normal distribution estimated from the sample data from that target (abscissa axis). With a perfect fit between normal distribution and data, all markers would appear along the thin line, though in practice, minor deviations at the tails of the distribution can be expected due to data scarcity. The plots show that score data follow normal distributions fairly well for each of the nine targets, with a tendency towards longer tails for target `M1020` indicated by the slightly curved shape of the plot. Corresponding plots for the remaining targets also show little deviation from the thin line, supporting our assumption that true-speaker score distributions are well approximated by normal distributions. At least it seems clear that there is no systematic deviation from straight lines in the plots shared among targets to suggest that another family of uni-modal parametric distributions would be more appropriate than the normal distribution. Inspection of normal quantile plots for impostor score data leads us to the same conclusion, as well as plots for the baseline system and the landline/office condition.

7.2.1.2 Gender groups

Next we look at the distribution of score data over groups of target speakers in the gate/hall condition and test set S2b_G8. Figure 7.2 shows normal quantile plots for impostor and true-speaker scores in the groups of female and male target speakers respectively, while Figure 7.3 is a more direct visualization of the same distributions using histograms together with normal distributions computed from the same data. Each histogram have been normalized to have area 1 (using trapezoidal numerical integration) to approximate a probability density function. All distributions except that for female impostor scores follow their corresponding normal distribution well. Female impostor data have a shorter left tail than expected from the normal distribution, but this is also the distribution computed from the smallest number of score points (208 score values vs. for example 913 for the male impostor data). The number of tests in the S2b_G8 test set is unevenly spread over targets. We therefore also created the corresponding normal quantile plots from score data where

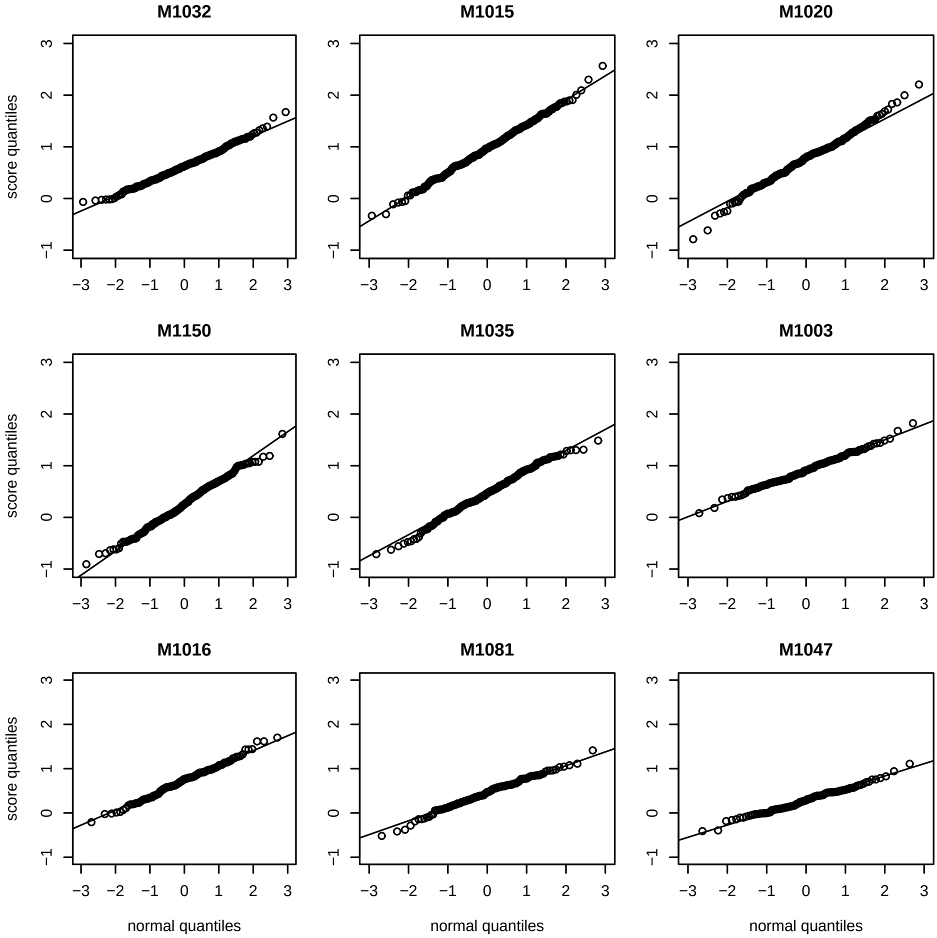


Figure 7.1: Normal quantile plots for true-speaker score data (ordinate axes) from the retrained system against their normal distributions (abscissa axes) for nine individual target speakers in the PER gate/hall condition.

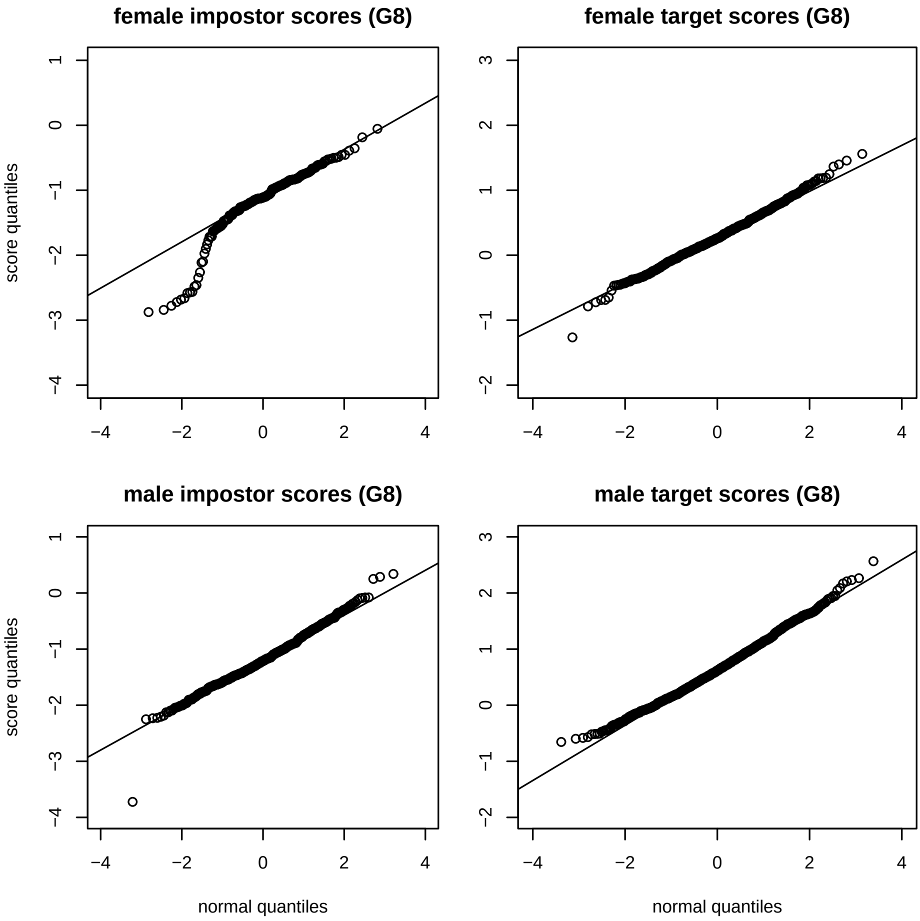


Figure 7.2: normal quantile plots for impostor or true-speaker score data (ordinate axes) from the retrained system against their normal distributions (abscissa axes) for female and male target speakers in the PER gate/hall condition.

the number of tests was limited to 20 impostor tests and 40 true-speaker tests per target speaker[4]. A comparison with Figure 7.2 showed very little difference.

[4] 208 and 595 points for the female impostor and target distributions respectively, and 760 and 1401 points respectively for the corresponding male distributions

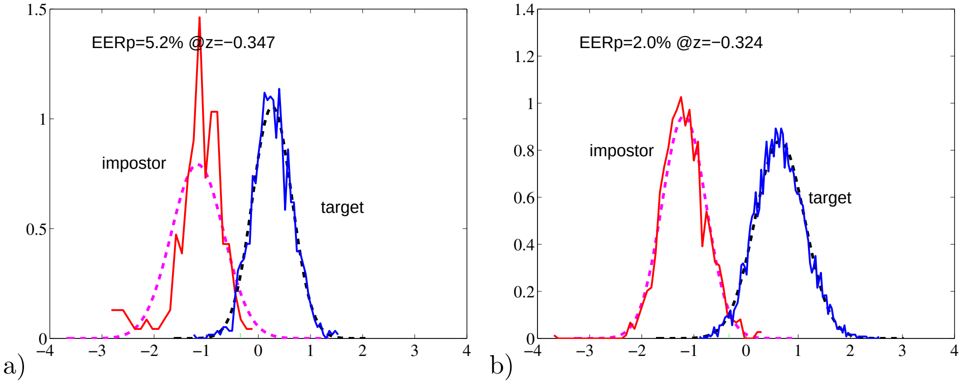


Figure 7.3: Score histograms for true-speaker and impostor tests, and normal approximations of the two distributions, for a) female and b) male data in the PER gate/hall condition. Histograms for female data have 25 and 50 bins for impostor and true-speaker scores respectively, and 50 and 100 bins for male data. Scores are produced by the retrained system using the E2a_G8 enrollment set and the S2b_G8 test set.

7.2.1.3 Test group

We then pool scores from male and female targets to look at the distribution of all scores from the S2b_G8 test set in Figure 7.4 (upper panes; normal quantile plots) and Figure 7.5a (histograms). Figure 7.5b and the lower panes of Figure 7.4 illustrate the the corresponding distributions from the landline/office condition (test set S2b_LO). The fit is fair also for these distributions. The curved shape of the quantile plot for the gate/hall true-speaker scores indicates this distribution is slightly skewed.

A third method to assess the fit of sample data to a normal distribution is to use a formal hypothesis testing method for normality, like the Shapiro-Wilk's normality test or the Kolmogorov-Smirnov test. Neither of the two tests supports the hypothesis of normality for our score data. Note that these tests are powerful in rejecting distributions where tails deviate from the normal distribution, and we have seen from the quantile plots that tails do deviate.

To summarize our findings this far from studying score distributions from the retrained system, it seems that impostor and true-speaker scores for individual target speakers follow normal distributions well. Also pooled score values for male and female target groups, and even all targets taken together, follow normal distributions reasonably well. We cannot state the distributions of pooled scores from our test group *are* normally distributed, but we think they are close enough to be approximated by normal distributions. We therefore move on to using the normal approximation as a basis for computing error rates.

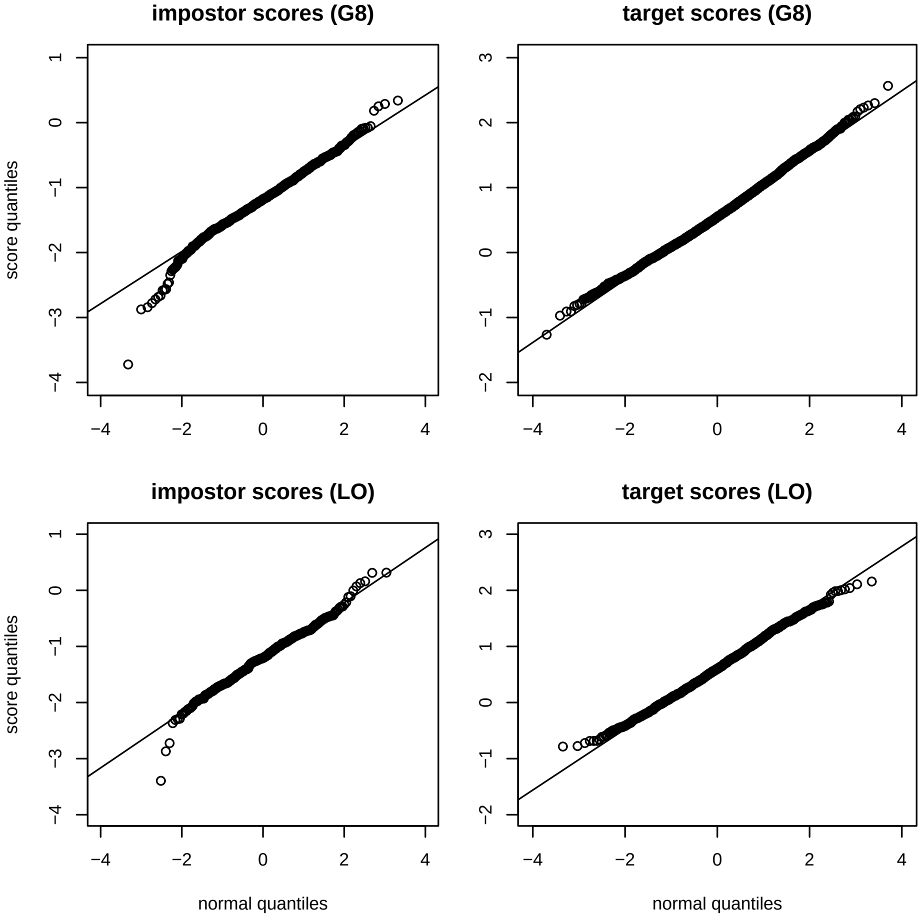


Figure 7.4: normal quantile plots for impostor and true-speaker score data (ordinate axes) against their normal distributions (abscissa axes). Upper panes show the PER gate/hall condition, while the lower panes show the telephone/landline condition.

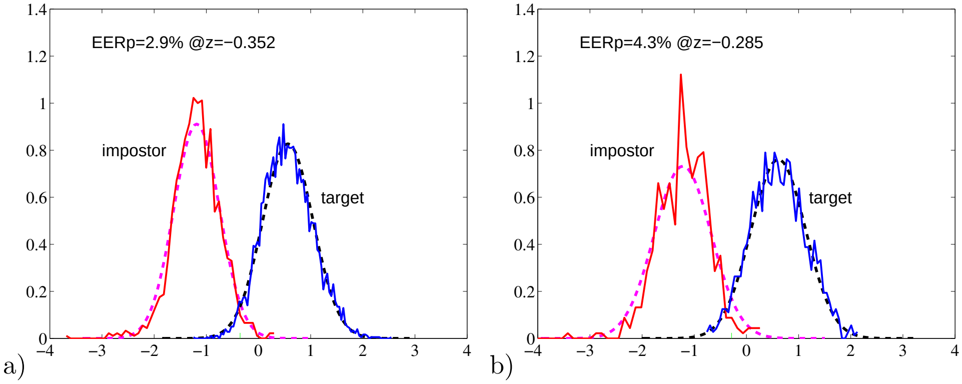


Figure 7.5: Score histograms for true-speaker and impostor tests, and normal approximations of the two distributions, for the a) gate/hall and b) landline/office conditions. Histograms have 50 bins each except for the true-speaker class in the gate/hall condition which has 100 bins. Scores are produced by the retrained system using the E2a_*c* enrollment set and the S2b_*c* test set.

7.3 Error rate

Like suggested by Elenius and Blomberg (2002), statistical models can be applied either directly to decision errors or to the score values that decisions are based on (assuming a score-based ASV system). Modeling score values has the advantage that models are independent of a particular decision threshold, but requires that one can observe score values produced by the ASV system. Modeling decision errors directly is an alternative when the decision threshold has already been determined, for example when analyzing an ASV system put into operation in a production environment.

After recalling the conventional ML estimates of false accept and false reject rates for completeness, this section derives MAP estimators for the same quantities using first the non-parametric method then the parametric score distribution method. The non-parametric methods operate on the *decision output* of an ASV system, while parametric methods compute error rates from a parametric model of score distributions estimated from *score output* of the ASV system (cf. Table 7.1).

A Bayesian (MAP) approach involves assuming or estimating a prior distribution of model parameters. If estimated from data, the prior distribution itself is useful in that it describes how the parameters, for example a false reject error rate, vary within a population of target speakers. This was used for example by Elenius and Blomberg (2002) to compute a decision threshold based on the criterion that a given maximum fraction of targets were allowed a false reject error rate greater than a given value.

7.3.1 Non-parametric methods

Like in Section 2.5.2, assume false reject errors by an ASV system for a given target speaker are generated at random and without memory at a constant rate, i.e. assume each true-speaker test is a Bernoulli trial with a constant error probability p. We already defined X as the random variable for the number of errors observed in N trials, and stated that X is binomially distributed with probability mass function given by (2.14).

Correspondingly, assume false accept errors by an ASV system for a given pair of impostor and target speaker are generated at random and without memory at a constant rate. The same distribution then applies to the impostor case. For simplicity, the derivation of error rate estimators below will be presented for the true-speaker case, i.e. for the false reject rate, where not otherwise stated. Given the corresponding assumption for the impostor case, the same derivations can be applied to the estimators of false accept rate.

7.3.1.1 Maximum likelihood

The ML estimate of the false reject error probability p follows directly from the assumption of binomial distribution

$$\hat{p} = \arg\max_{p} P_X(x|p) = x/N. \tag{7.1}$$

7.3.1.2 Maximum a posteriori

To formulate a Bayesian estimator for p given an observation of x errors in N trials, we treat p as the outcome of a random variable ρ. That is, for each target speaker, a value for p is drawn at random from ρ. The posterior distribution of ρ is given by Bayes rule as

$$P_\rho(p|x) = \frac{P_X(x|p)P(p)}{P(x)}. \tag{7.2}$$

We then need a prior distribution $P(p)$ describing our *a priori* knowledge, or belief, about ρ. While any distribution, theoretically or empirically motivated, is possible for use as a prior distribution in the Bayesian framework, a *conjugate prior* to the distribution of observations $P_X(x|p)$ allows for closed-form posterior distributions $P_\rho(p|x)$ (e.g. Lee, 1989). The conjugate prior of the binomial distribution is a beta distribution. Hence, if a beta distribution would be appropriate for our distribution of ρ, it would also be a good choice for the prior distribution. Then we would assume $\rho \in \text{Beta}(\alpha, \beta)$ with probability density function

$$P(p) = \frac{(1-p)^{\beta-1}p^{\alpha-1}}{B(\alpha,\beta)} \tag{7.3}$$

where $B(\alpha, \beta)$ is the beta function and $\alpha, \beta > 0$ our hyper parameters.

Fitting prior distributions To test if the beta distribution is appropriate for our prior, we try to fit beta distributions to observed individual false reject rates in our data. Given a list of observations of ρ, hyper parameters α and β can be computed from the first and second moments of observations. For a distribution Beta(α, β) the mean and variance are

$$\begin{cases} \mu_\rho = \dfrac{\alpha}{(\alpha+\beta)} \\ \sigma_\rho^2 = \dfrac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}. \end{cases} \tag{7.4}$$

Thus, estimates $\hat{\mu}_\rho$ and $\hat{\sigma}_\rho^2$ of the mean and variance of ρ provide estimates for our two hyper parameters

$$\begin{cases} \hat{\alpha} = \left(\dfrac{\hat{\mu}_\rho - \hat{\mu}_\rho^2}{\hat{\sigma}_\rho^2} - 1 \right) \hat{\mu}_\rho \\ \hat{\beta} = \left(\dfrac{1}{\hat{\mu}_\rho} - 1 \right) \hat{\alpha}. \end{cases} \tag{7.5}$$

The upper pane of Figure 7.6 shows histograms of individual false reject rates for target speakers and the retrained and baseline research systems, and the commercial system without adaptation, together with beta distributions with parameters α and β computed according to (7.5). False reject rates in the histograms are computed with the non-parametric method (FRRd) directly from score data points and a target-independent EERd-threshold. The figure displays a good fit for the beta distribution in all three cases, suggesting that a beta distribution is indeed appropriate as the prior distribution for estimating individual false reject rates. The lower pane of Figure 7.6 shows that false accept rates are also properly described by beta distributions. Note, however, that histograms show false accept rate per target speaker averaged over all available impostor speakers, and not for unique pairs of impostor and target speaker as was the initial assumption for the derivation of the MAP estimator.

MAP estimator We can then proceed to formulate the maximum a posteriori (MAP) estimator for p. Because the beta distribution is a conjugate prior of the binomial data distribution, the posterior distribution is also a beta distribution. In our case, after observing x errors in N attempts, we have $P_\rho(p|x) \in$ Beta$(\alpha+x, \beta+N-x)$ (e.g. Lee, 1989) with probability mass function

$$P_\rho(p|x) = \frac{(1-p)^{\beta+N-x-1} p^{\alpha+x-1}}{B(\alpha+x, \beta+N-x)}. \tag{7.6}$$

The MAP estimate of p is

$$\tilde{p} = \arg\max_p P_\rho(p|x) \tag{7.7}$$

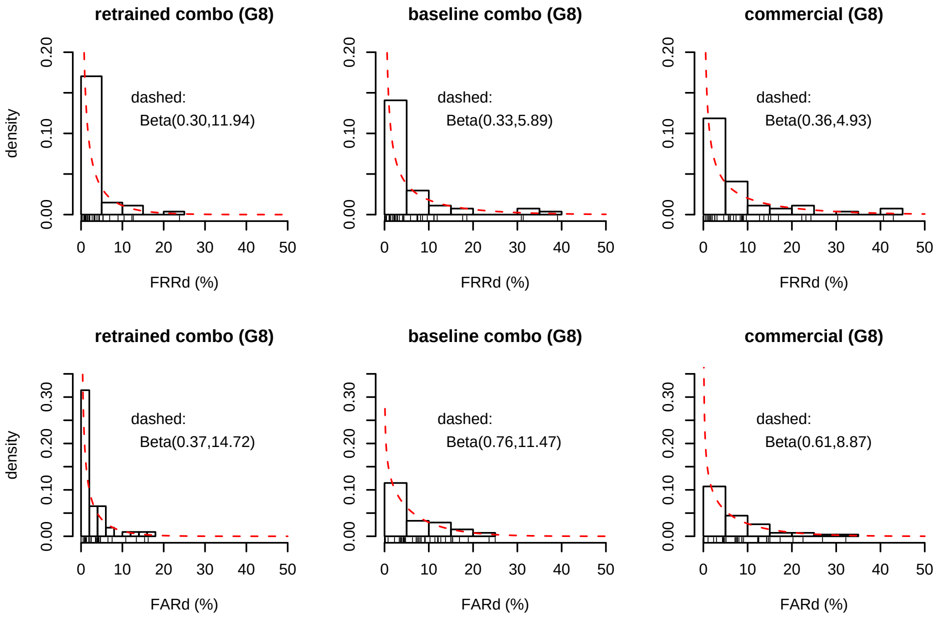


Figure 7.6: Histograms of FRRd (upper panes) and FARd (lower panes) for 54 individual target speakers and three ASV systems in the PER gate/hall condition using enrollment set E2a_G8 and test set S2b_G8. Decision threshold for each plot is the global, gender-independent EERd threshold. Each graph also plots a fitted beta distribution (dashed curve) with α and β computed from the observations of false reject/accept rates. The short lines below the histogram bars indicate individual FRRd or FARd values.

that is

$$\tilde{p} = \begin{cases} \frac{\alpha+x-1}{\alpha+\beta+N-2} & \text{if } \alpha + x > 0, \\ 0 & \text{otherwise} \end{cases} \tag{7.8}$$

which tends to x/N as $N \to \infty$, i.e. the MAP estimate tends to the ML estimate.

As already stated in Section 2.5.2.1, the assumption about false reject errors being generated at random and without memory at a constant rate for a given target is obviously not strictly valid. For example, mismatching input channels and background noise, learning effects and temporary voice changes from head colds are likely to vary the error rate over time as they occur. Recent experience of erroneous decisions may influence a claimant to alter his speaking, causing attempts to be not strictly without “memory”. These are flaws in the assumptions underlying the derivation of the MAP estimator that should be kept in mind when interpreting results from it.

Similar assumptions were made about false accept errors for a given pair of impostor and target speaker. Also these assumptions are obviously not strictly valid. An impostor who is given multiple tries may for example vary his strategy in trying to imitate the target. Furthermore, the prior distributions for false accept rates depicted in Figure 7.6 are not computed for individual pairs of impostor and target. Instead, the FAR values presented in the histograms are averages over all available impostors for each individual target speaker. A MAP estimator used to compute a corresponding average FAR against a given target speaker, that uses such a prior distribution, is derived on the assumption that the false accept rate for a given target speaker (not specifying a particular impostor) is generated at random and without memory at a constant rate. This assumption is even less valid, since it is well known that some impostors may be more successful than others against a given target speaker (Doddington et al., 1998).

7.3.2 Parametric methods

By parametric score distribution methods, we refer to methods that estimate error rate through a two-step procedure: first estimate the parameters of a statistical model of true-speaker and impostor score distributions, and second compute error rates from the statistical models.

In this section we will assume normal score distributions. It was shown in the beginning of this chapter that this distribution is a reasonable approximation of true-speaker and impostor score distributions, at least for individual target speakers and the ASV system described in Chapter 3.

Given the assumption about normal score distributions and estimates of mean and variance of a true-speaker (μ_T, σ_T^2) and impostor (μ_I, σ_I^2) score distributions, resulting error rates can easily be computed. False reject and false accept error rates for a given decision threshold θ are

$$\begin{cases} \mathrm{FRRp} = \Phi(\theta|\mu_T, \sigma_T^2) \\ \mathrm{FARp} = 1 - \Phi(\theta|\mu_I, \sigma_I^2) \end{cases} \tag{7.9}$$

where $\Phi()$ is the normal cumulative distribution function. EERp can be calculated by first calculating the EERp-threshold θ_{EERp}

$$\begin{aligned} \theta_{\mathrm{EERp}} &= \arg\min_{\theta} |\mathrm{FARp}(\theta) - \mathrm{FRRp}(\theta)| \\ &= \arg\min_{\theta} |1 - \Phi(\theta|\mu_I, \sigma_I^2) - \Phi(\theta|\mu_T, \sigma_T^2)| \end{aligned} \tag{7.10}$$

and then EERp itself

$$\mathrm{EERp} = \Phi(\theta_{\mathrm{EERp}}|\mu_T, \sigma_T^2). \tag{7.11}$$

7.3.2.1 Maximum likelihood

Denote as μ and σ^2 the mean and variance of a normal distribution approximating the true-speaker score distribution for a given target speaker. The probability of observing a sequence of n independent true-speaker score values z (denoted as a vector $\mathbf{z}$) from a given target speaker is then

$$P_Z(\mathbf{z}|\mu,\sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{(z_i-\mu)^2}{2\sigma^2}} = \frac{1}{(2\pi\sigma^2)^{n/2}} \mathrm{e}^{-\frac{S_z+n(\mu-\mu_z)^2}{2\sigma^2}} \tag{7.12}$$

where μ_z is the sample average over observed score values in $\mathbf{z}$, and

$$S_z = \sum_{i=1}^{n}(z_i-\mu_z)^2. \tag{7.13}$$

The ML estimates of μ and σ^2 are then

$$(\hat{\mu},\hat{\sigma}^2) = \arg\max_{\mu,\sigma^2} P_Z(\mathbf{z}|\mu,\sigma^2) \tag{7.14}$$

that is,

$$\begin{cases} \hat{\mu} = \mu_z \\ \hat{\sigma}^2 = S_z/n. \end{cases} \tag{7.15}$$

The ML estimates of the mean and variance of the impostor score distribution are computed analogously from a sequence of independent impostor score values.

7.3.2.2 Maximum a posteriori

Again denote as μ and σ^2 the mean and variance of a normal distribution approximating the true-speaker score distribution for a given target speaker. Treat μ and σ^2 as the outcome of random variables U and V. The posterior joint distribution of U and V after observing a sequence of n independent true-speaker score values z (denoted as a vector $\mathbf{z}$) from a given target speaker is

$$P_{U,V}(\mu,\sigma^2|\mathbf{z}) = \frac{P_Z(\mathbf{z}|\mu,\sigma^2)P(\mu,\sigma^2)}{P(\mathbf{z})} \tag{7.16}$$

where the distribution of observations $\mathbf{z}$ given known μ and σ^2 was given in (7.12).

The conjugate prior for the normal distribution with unknown mean and variance is a normal/chi-squared distribution (e.g. Lee, 1989, pp. 73 and 237)

$$\begin{aligned} P(\mu,\sigma^2) &= P(\sigma^2)P(\mu|\sigma^2) \\ &= \frac{S_0^{\nu_0/2}}{2^{\nu_0/2}\Gamma(\nu_0/2)} \frac{1}{\sigma^{2(\nu_0/2+1)}} \mathrm{e}^{-\frac{S_0}{2\sigma^2}} \times \frac{1}{\sqrt{2\pi\sigma^2/n_0}} \mathrm{e}^{-\frac{(\mu-\mu_0)^2}{2\sigma^2/n_0}} \\ &\propto \frac{1}{\sigma^{2((\nu_0+1)/2+1)}} \mathrm{e}^{-\frac{S_0+n_0(\mu-\mu_0)^2}{2\sigma^2}} \end{aligned} \tag{7.17}$$

with hyper-parameters (S_0,ν_0) for the inverse chi-squared part $P(\sigma^2)$, and (μ_0,n_0) for the normal part $P(\mu|\sigma^2)$.

Fitting prior distributions Starting with the inverse chi-squared distribution $P(\sigma^2)$ we have

$$\begin{cases} \mu_V = E[\sigma^2] = \dfrac{S_0}{\nu_0 - 2} \\ \sigma_V^2 = E[(\sigma^2 - \mu_V)^2] = \dfrac{2S_0}{(\nu_0 - 2)^2(\nu_0 - 4)}. \end{cases} \tag{7.18}$$

Estimates $\hat{\mu}_V$ and $\hat{\sigma}_V^2$ of the mean and variance of V therefore provide estimates for our first two hyper parameters as

$$\begin{cases} \hat{S}_0 = 2\hat{\mu}_V \left(\dfrac{\hat{\mu}_V^2}{\hat{\sigma}_V^2} + 1 \right) \\ \hat{\nu}_0 = 2 \left(\dfrac{\hat{\mu}_V^2}{\hat{\sigma}_V^2} + 2 \right). \end{cases} \tag{7.19}$$

Since in (7.17), μ and σ^2 are not independent (there is no conjugate prior with independent μ and σ^2), to estimate hyper-parameters for the normal part of (7.17) we first compute the marginal distribution of μ (analogous to Lee (1989, p. 70))

$$\begin{aligned} P(\mu) &= \int_0^\infty P(\mu, \sigma^2)\, d\sigma^2 \\ &\propto \int_0^\infty \sigma^{-2((\nu_0+1)/2+1)}\, \mathrm{e}^{-\frac{S_0 + n_0(\mu-\mu_0)^2}{2\sigma^2}}\, d\sigma^2 \\ &\propto \left(S_0 + n_0(\mu - \mu_0)^2\right)^{-(\nu_0+1)/2} \end{aligned} \tag{7.20}$$

By substituting into (7.20)

$$t = \frac{\mu - \mu_0}{\sqrt{S_0/(\nu_0 n_0)}} \tag{7.21}$$

we see that

$$\begin{aligned} P(\mu) &\propto \left(S_0 + n_0(\mu - \mu_0)^2\right)^{-(\nu_0+1)/2} \\ &\propto \left(1 + \frac{t^2}{\nu_0}\right)^{-(\nu_0+1)/2} \end{aligned} \tag{7.22}$$

which means that our substitute variable t has a Student's t distribution with ν_0 degrees of freedom with mean 0 and variance $\nu_0/(\nu_0 - 2)$. It then follows directly from (7.21) that

$$\begin{cases} \mu_U = E[\mu] = \mu_0 \\ \sigma_U^2 = E[(\mu - \mu_U)^2] = \dfrac{S_0}{\nu_0 n_0} \cdot E[t^2] = \dfrac{S_0}{n_0(\nu_0 - 2)}. \end{cases} \tag{7.23}$$

Estimates $\hat{\mu}_U$ and $\hat{\sigma}_U^2$ of the mean and variance of U therefore provide estimates for our last two hyper parameters as

$$\begin{cases} \hat{\mu}_0 = \hat{\mu}_U \\ \hat{n}_0 = \dfrac{S_0}{\hat{\sigma}_U^2(\nu_0 - 2)}. \end{cases} \tag{7.24}$$

Figure 7.7 shows histograms of observed score variance and mean values for true-speaker tests for individual target speakers and the retrained and baseline research systems, and the commercial system without adaptation in the gate/hall condition. Histograms of score variance values are plotted together with inverse chi-squared distributions with hyper parameters S_0 and ν_0 computed according to (7.19), while histograms of score mean values are plotted together with translated and scaled Student's t-distributions (7.21) with hyper parameters μ_0 and n_0 computed according to (7.24). The figure displays a good fit for the estimated inverse chi-squared distribution in all three cases, suggesting that this distribution family is appropriate as the prior distribution for estimating the variance of true-speaker scores for individual targets. The corresponding fit for the estimated Student's t distributions are reasonable, but histograms are not quite symmetric like the parametric distributions. Figure 7.8 shows that variance and mean of impostor scores for individual target speakers are also properly described by the chosen parametric distributions.

On the whole, it looks like the normal/chi-squared distribution is an appropriate prior distribution for the unknown mean and variance of score values for individual target speakers, both for true-speaker and impostor scores.

MAP estimator We can then proceed to formulate the joint maximum a posteriori (MAP) estimators for μ and σ^2. Because the normal/chi-squared distribution is a conjugate prior for the normal distribution with unknown mean and variance, the posterior distribution is also a normal/chi-squared distribution. In our case, after observing n independent true-speaker scores $\mathbf{z}$ for a given target speaker, we have (Lee, 1989, p. 74)

$$\begin{aligned} P(\mu,\sigma^2|\mathbf{z}) &\propto P(\mathbf{z}|\mu,\sigma^2)P(\mu,\sigma^2) \\ &\propto \frac{1}{\sigma^{2(n/2)}}\,\mathrm{e}^{-\frac{S_z+n(\mu-\mu_z)^2}{2\sigma^2}} \times \frac{1}{\sigma^{2((\nu_0+1)/2+1)}}\,\mathrm{e}^{-\frac{S_0+n_0(\mu-\mu_0)^2}{2\sigma^2}} \\ &= \frac{1}{\sigma^{2((\nu_0+n+1)/2+1)}}\,\mathrm{e}^{-\frac{S_0+S_z+n_0(\mu-\mu_0)^2+n(\mu-\mu_z)^2}{2\sigma^2}} \\ &= \frac{1}{\sigma^{2((\nu_1+1)/2+1)}}\,\mathrm{e}^{-\frac{S_1+n_1(\mu-\mu_1)^2}{2\sigma^2}} \end{aligned} \tag{7.25}$$

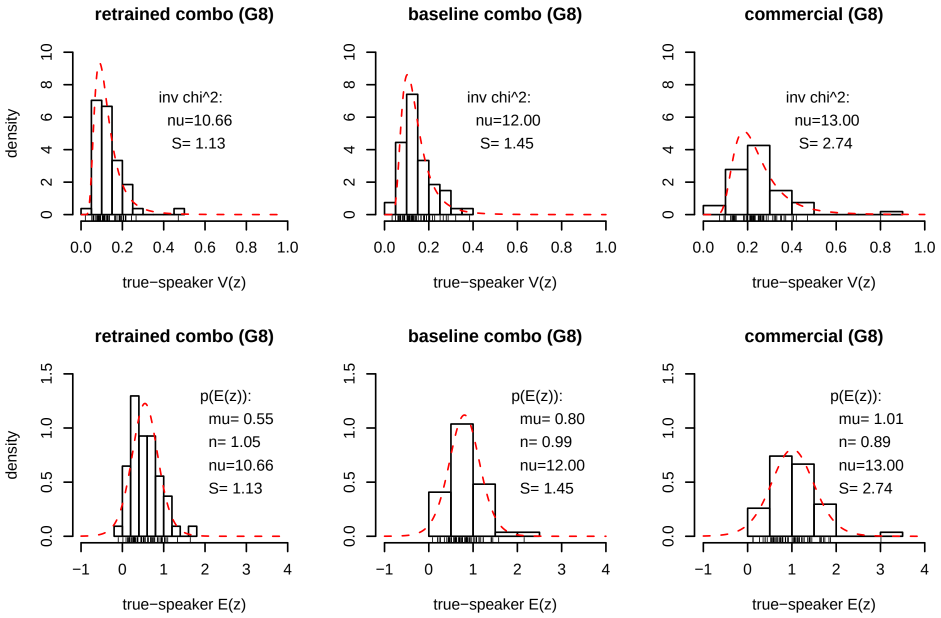


Figure 7.7: Histograms of true-speaker score variance (upper panes) and mean (lower panes) for 54 individual target speakers and three ASV systems in the gate/hall condition using enrollment sets E2a_G8 and test sets S2b_G8. Each graph also plots a fitted inverse chi-squared distribution (upper panes) or a translated and scaled Student's t distribution (lower panes) with hyper parameters computed from the observations of score mean and variance values. The short lines below the histogram bars indicate individual score variance or mean values.

where

$$\begin{cases} S_1 = S_0 + S_z + \dfrac{n_0 n}{n_0 + n}(\mu_0 - \mu_z)^2 \\ \nu_1 = \nu_0 + n \\ \mu_1 = \dfrac{n_0\mu_0 + n\mu_z}{n_0 + n} \\ n_1 = n_0 + n \end{cases} \tag{7.26}$$

That is, the joint posterior distribution has the same shape as the prior but with parameters (S_1, ν_1) for the inverse chi-squared part, and (μ_1, n_1) for the normal part.

The joint MAP estimate of μ and σ^2 is

$$(\tilde{\mu}, \tilde{\sigma}^2) = \arg\max_{\mu,\sigma^2} P_{U,V}(\mu, \sigma^2|\mathbf{z}) \tag{7.27}$$

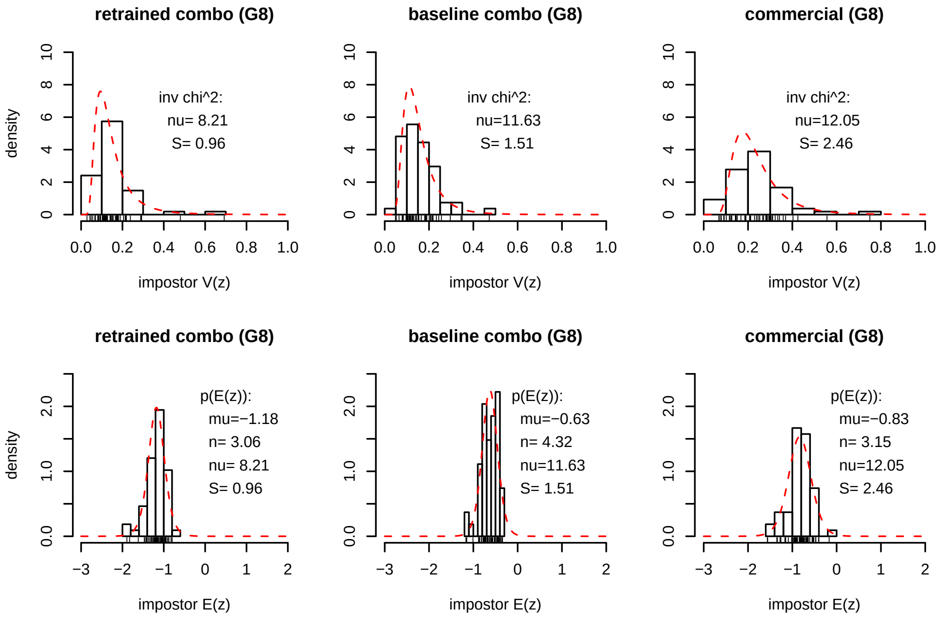


Figure 7.8: Histograms of impostor score variance (upper panes) and mean (lower panes) for 54 individual target speakers and three ASV systems in the gate/hall condition using enrollment sets E2a_G8 and test sets S2b_G8. Each graph also plots a fitted inverse chi-squared distribution (upper panes) or a translated and scaled Student's t distribution (lower panes) with hyper parameters computed from the observations of score mean and variance values. The short lines below the histogram bars indicate individual score variance or mean values.

that is[5],

$$\begin{cases} \tilde{\mu} = \mu_1 \\ \tilde{\sigma}^2 = \dfrac{S_1}{\nu_1 + 3} \end{cases} \tag{7.28}$$

which tends to the pair of ML estimates $\hat{\mu} = \mu_z$ and $\hat{\sigma}^2 = S_z/n$ as $n \to \infty$.

Given MAP-estimated parameters of the true-speaker score distribution for a target, the corresponding false reject rate for a threshold θ can now be calculated from Equation (7.9).

The assumptions underlying this MAP estimator of score distribution parameters are similar to those made for the MAP estimator for error rate: independent

[5] the maximization problem (7.27) can be solved with standard calculus by solving for equality to zero the partial derivatives of the logarithm of (7.25) with respect to μ and σ^2.

trials and a fixed, time-invariant, parameter distribution for a given target speaker. In addition we here also made the assumption that score distributions are normal.

7.4 Experiments

In this section we compare the *non-parametric* and *parametric* error rate estimation methods (cf. Table 7.1) experimentally using PER data. We first look at estimation of error rates for individual targets and then for groups of targets. The two cases differ in the number of available test trial observations and in that assumptions underlying the estimation methods are violated to a higher degree for groups of target speakers than for individual target speakers. For individual targets we look both at batch estimation of (individual) EER, at incremental estimation of FRR and FAR, and at the problem of detecting error-prone target speakers ("goats"). For target groups we look at global EER and DET curves.

7.4.1 Error rate by target

7.4.1.1 Batch estimation

In this section we consider the problem of computing the EER for individual target speakers given a sequence of observed score values. Two approaches to computing the EER were described above: the conventional non-parametric method (EERd) and the proposed parametric score distribution method (EERp). Results from the two methods are compared here in terms of histograms of error rate in a group of target speakers.

A target-specific EER, based on a target-dependent *a posteriori* decision threshold, is of little interest as a measure of actual performance in a real application. Because the threshold is adjusted after seeing (many) true-speaker *and* impostor test trials, the indicated error rate is usually much smaller than that achieved in practice (Nordström et al., 1998), because in reality the decision threshold must be determined before the decision can be taken. In practice, either a target-independent threshold is used, or a target-dependent *a priori* threshold must determined from data available before the time of the decision. However, the target-specific EER may be of interest as a measure of speaker discriminability. It is used for this purpose in Chapter 8 of this thesis, for example.

Figure 7.9 shows normalized histograms for both EERp and EERd over individual targets for three systems in the gate/hall condition: the retrained and baseline research systems and the commercial system without speaker adaptation.

A compelling feature shared by the three EERp histograms in Figure 7.9 is "smoothness"; EERp values for the three systems seem to be drawn from smooth distributions. Distributions resemble for example an exponential distribution, or in the case of the baseline system a gamma distribution (of which the exponential distribution is a special case). EERd histograms, in contrast, are more "rough".

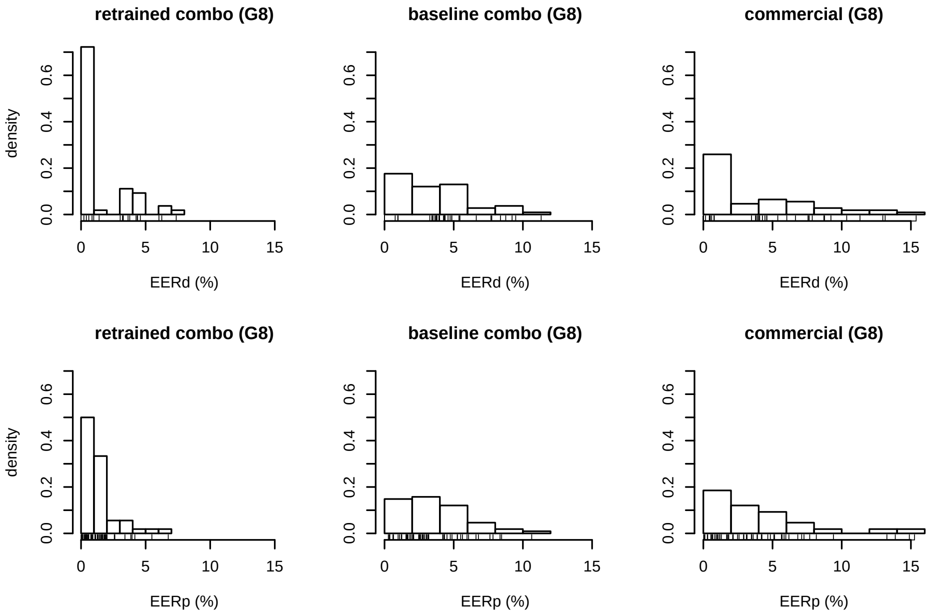


Figure 7.9: Comparison between histograms of EERd (upper panes) vs. EERp (lower panes) for individual target speakers and three ASV systems in the gate/hall condition using enrollment sets E2a_G8 and test sets S2b_G8. The short lines below the histogram bars indicate individual values.

It feels more intuitive that the true distribution of EER for individual targets is smooth.

For all three systems, histograms show more targets in their left-most bin for EERd than for EERp.

Figure 7.10 shows histograms of EERp over individual targets for the three ASV system in the landline/office condition. Again all three histograms are smooth, and they all seem to fit an exponential distribution.

7.4.1.2 Incremental estimation

Comparing the two methods (Equations (7.8) vs. (7.28)) for estimating the false reject or false accept rate for a given target speaker, we expect the one operating on score values to be more efficient with respect to how many trials are needed for an expected maximum estimation error. This is because there is more information in the observation of a score value than in the observation of a binary decision.

To evaluate the two methods for estimating individual false reject rates we used a leave-one-out cross validation method. Out of the 54 target speakers in the S2b_G8

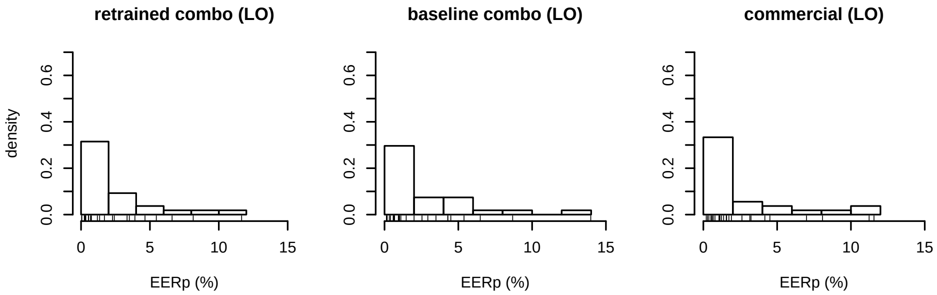


Figure 7.10: Histograms of EERp for individual target speakers and three ASV systems in the landline/office condition using enrollment sets E2a_LO and test sets S2b_LO. The short lines below the histogram bars indicate individual EERp values.

test set, the 17 targets (14 male, 3 female) with more than 100 true-speaker trials were used for testing. For each of the 17, true-speaker score distribution parameters and false reject error rates were computed from the target's own true-speaker scores, and prior distribution parameters were computed from all other 53 targets in the S2b_G8 test set. A target-independent (*a posteriori*) EERd-threshold computed from data from all 54 targets was used to map scores to decisions.

True-speaker score observations were grouped into two-week blocks such that block 1 contains scores from test sessions recorded during 0-13 days after enrollment, block 2 contains scores from 14-27 days after enrollment, etc. The shortest time between enrollment and the last recorded test session among the 17 target speakers is 34 weeks, corresponding to 17 blocks of score data. These blocks of score data were then used to compute estimates of true-speaker score distribution parameters and false reject error rates as a function of time after enrollment. Since target speakers in general did not produce an equal amount of test sessions per week, score observations are unevenly distributed among blocks. Blocks with less than three score observations were not used for these tests.

Figure 7.11 shows how MAP estimates of false reject error rate (Eq. (7.8) for the non-parametric estimation method and (7.28, 7.9) for the parametric score distribution method) evolve with an increasing number of observations and time after enrollment for the 17 individual target speakers and the retrained research system in the gate/hall condition. The first estimate (at 0 weeks) is based on the prior distributions only, and no observations from target speakers themselves. The second estimate (at 2 weeks) is based on the prior distribution and the first block of score data. After each new observed block of scores, the most recent posterior distribution is used as the prior distribution for the next block, etc. Upper panes show the estimates themselves, while lower panes show the root-mean-square (RMS) error for each estimate assuming the *a posteriori* ML FRRd estimate for each target

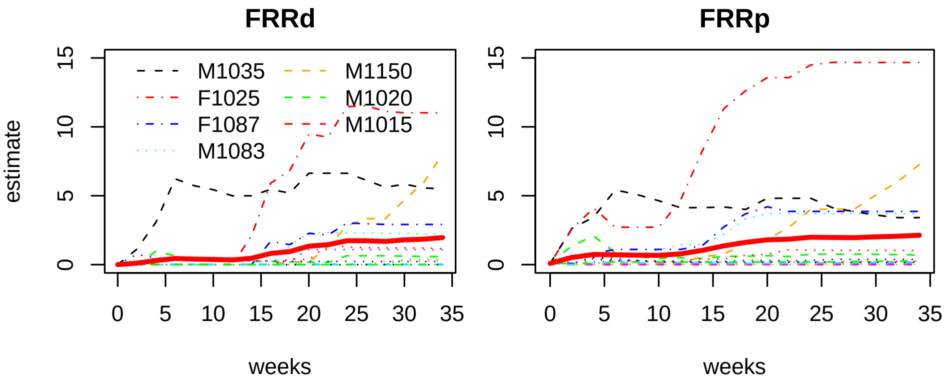


Figure 7.11: Non-parametric (FRRd) and parametric (FRRp) accumulated MAP estimates of false reject rate with time between enrollment and test for 17 target speakers in the PER gate/hall condition and the retrained research system. Thick lines show group average. Legend is the same for both graphs.

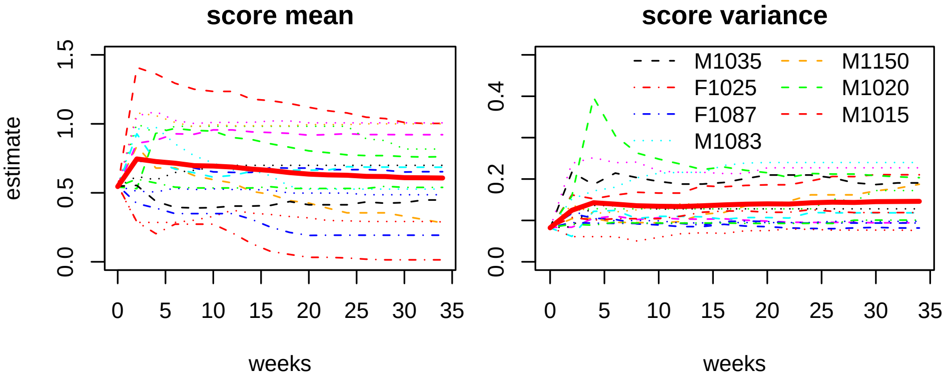


Figure 7.12: Accumulated MAP estimates of score mean and variance providing the FRRp estimates in Figure 7.11. Thick lines show group average. Legend is the same for both graphs.

as the “true” reference value, i.e. the FRRd value computed using all the available true-speaker tests in the S2b_G8 test set for each target speaker (including tests in sessions possibly recorded more than 34 weeks after enrollment). The figure also shows the average estimate and the total RMS error over all 17 targets as thick lines.

FRRp-estimates for a target speaker are based entirely on the decision threshold and estimates of the mean and variance of an assumed normal true-speaker score distribution (7.9). Figure 7.12 shows how the latter estimates evolve. From score mean estimates it is clear that the prior information is “weak” since estimates change rather much after the first observed block of score data compared to the *a priori* estimates, and then change more smoothly. This is consistent with the parameter values seen for prior distributions in Figure 7.7 with $n_0 = 1.05$ which can be interpreted in the context of Equation (7.26) such that the prior data is worth a single observation only. The prior for variance estimation seems to be a little less weak since it takes two or three blocks of observations for estimates to level out.

For several targets in Figure 7.12, the score mean estimate decreases with time. This is also true for the average over all 17 targets, though this trend is much due to the curve with highest score mean (corresponding to target `M1015`) which has a very clear negative trend. The negative trend in score mean estimates translates into a positive trend in FRRp-estimates which does not appear as clearly. Because of the shape of the normal distribution, small fluctuations in the score mean estimate generally result in large changes in FRRp for speakers in the low score mean region relative to the decision threshold. The negative trend is consistent with previous research (cf. Rosenberg, 1976) concluding that dissimilarity between a target model and the target’s test utterances increase with time from enrollment.

From Figure 7.11 it is clear that false reject rates are low for most target speakers. Only five of the 17 targets demonstrate false reject rate estimates above the mean, which levels out at approximately 2%. It is re-assuring that these are the same five target speakers with the FRRd and FRRp estimates. Further comparing the two estimation methods, it seems that the parametric method (FRRp) facilitates finding of the targets with higher false reject rate earlier than the non-parametric method (FRRd). For example, at 10 weeks only one of the targets have FRRd>0.5% (`M1035`) while five targets have FRRp>0.5% (`M1035`, `F1025`, `F1087`, `M1083`, `M1020`). Of the latter five targets, all except `M1020` end up with both FRRd and FRRp estimates greater than 2% after 34 weeks. Seen as detectors of targets with high false reject rate (or “goat detectors”), both methods miss `M1150` who shows increased FRRd and FRRp estimates only after around 20 weeks.

Estimates presented this far were computed with the MAP methods derived in this section and prior distributions were adapted after each block of observations such that estimates after all data have been seen is effectively based on all the data. Figure 7.13 compares these MAP estimates to three other estimates. The first is an ML estimate based on accumulated data, i.e. estimates after a given block of score data is based on all observed data up to and including that block. This is the ML equivalent of the above MAP estimate. The second estimate is a MAP estimate based on the (original) prior distribution and the most recently observed block of score data, and the third estimate is the corresponding ML estimate based on the most recent block. Naturally, most recent block-estimates fluctuate much more than accumulated estimates, more so for ML compared to MAP estimates and for FRRd compared to FRRp estimates.

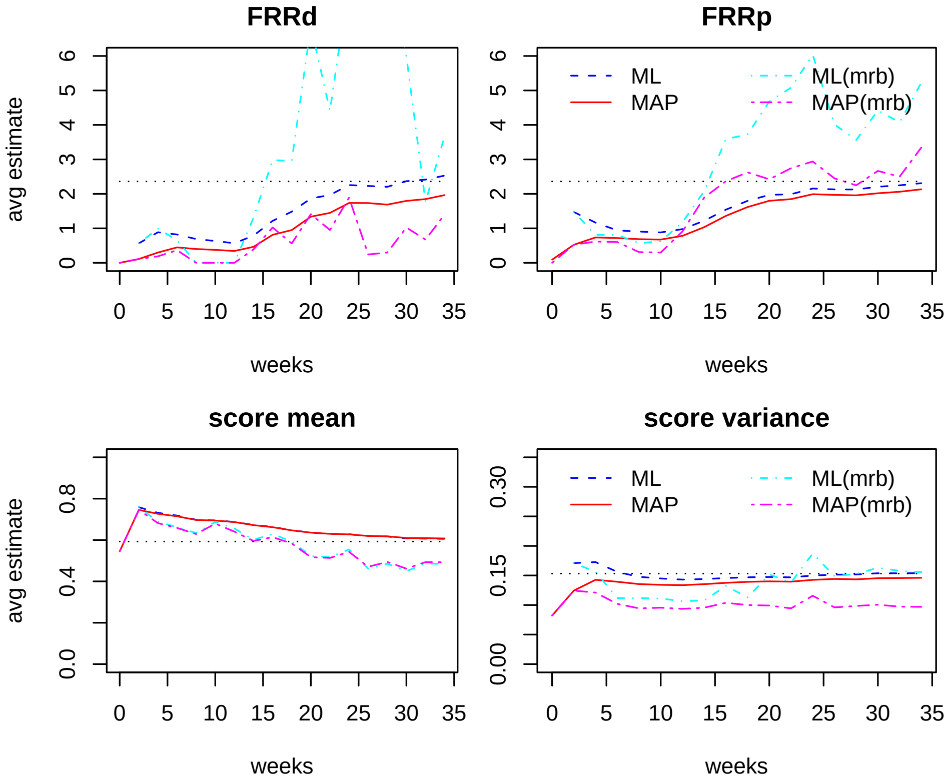


Figure 7.13: False reject rate with time between enrollment and test estimated by four variants each of non-parametric (FRRd) and parametric (FRRp) methods for 17 target speakers in the PER gate/hall condition and the retrained research system (upper panes). Variants are ML and MAP methods applied to accumulated observations and on the most recent two-week block of observations (mrb). Lower panes show estimates of score mean and variance providing the FRRp estimates. Legend is the same for all graphs.

Table 7.2: Comparison of non-parametric (EERd) and parametric (EERp) equal error rates (in %) and the corresponding *a posteriori* score thresholds ($\theta_{\text{EERd}}, \theta_{\text{EERp}}$) on several PER test sets and conditions (c).

test set				non-parametric		parametric		statistics		
name	c	N_{ts}	N_{imp}	θ_{EERd}	EERd	θ_{EERp}	EERp	$\Delta\theta$[a]	$\Delta\epsilon$[b]	rel[c]
S2b_c	G8	4643	1121	-0.348	2.41	-0.352	2.87	-0.004	0.46	19%
	LO	1228	422	-0.341	3.09	-0.285	4.33	0.056	1.24	40%
S2b_Q:c	G8[d]	977	393	-0.276	2.55	-0.316	2.76	-0.040	0.21	8%
	LO	977	393	-0.347	3.52	-0.304	4.65	0.043	1.13	32%
	MO	977	393	-0.404	4.82	-0.413	5.80	-0.009	0.98	20%
	MH	977	393	-0.392	5.28	-0.408	7.17	-0.016	1.89	36%

[a] threshold difference $\theta_{\text{EERp}} - \theta_{\text{EERd}}$

[b] equal error rate difference EERp − EERd

[c] equal error rate difference relative to EERd

[d] full name plus five-digit test utterances, as opposed to in other results reported for the S2b_Q:G8 test set where only name plus four digits are used (for example Figures 7.15 and 10.3)

7.4.2 Error rate by test group

The EER calculated from the normal distributions in the gate/hall case on the entire test set S2b_G8 is EERp = 2.9%, compared to EERd = 2.4% when calculated directly from the data. Table 7.2 shows the corresponding EER estimates for the other conditions. Parametric EERs are consistently higher than the respective non-parametric EERs. The average difference EERp − EERd is around 1.0, or 25% relative to EERd. These averages hold for all table entries as a group, as well as for the group of four S2b_Q:c entries. The average threshold difference $\theta_{\text{EERp}} - \theta_{\text{EERd}}$, on the other hand, is close to zero. This suggests that the two methods result in the same EER-threshold estimate, while the parametric method consistently estimates a higher EER than the conventional non-parametric method.

To allow a comparison between score data and their normal approximation in terms of DET curves, synthetic score data can be generated from computed normal distributions. Figure 7.14 compares real and synthetic score data for the retrained system in the gate/hall and landline/office conditions, while Figure 7.15 compares the four conditions using such synthetic data from the condition-parallel S2b_Q:c test sets. Corresponding non-parametric DET curves for the condition-parallel test sets are shown in Figure 10.3. While non-parametric DET curves in Figure 10.3 unexpectedly show little or no difference between the mobile/office and mobile/hall conditions, our parametric (synthetic) DET curves show such a difference more clearly. A more pronounced difference between gate/hall and landline/office is also shown by the parametric curves.

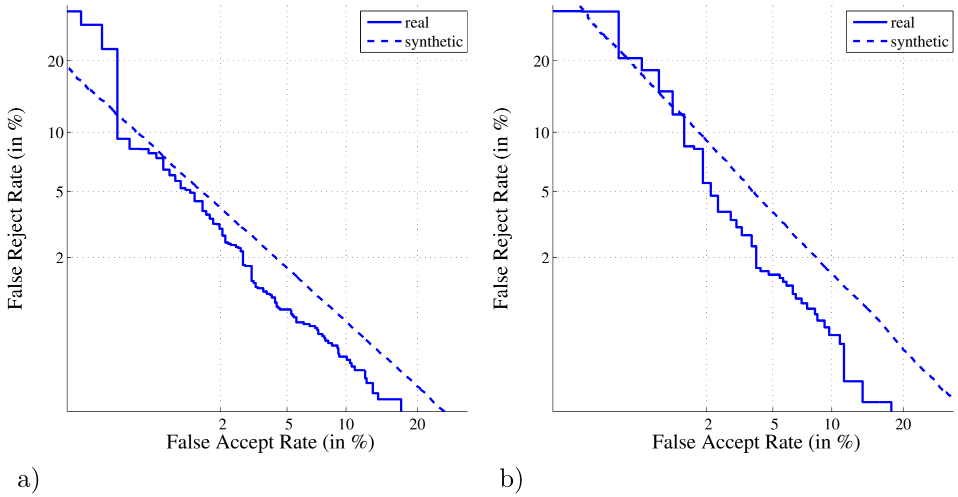


Figure 7.14: DET curves for the retrained research system in the a) gate/hall and b) landline/office conditions (enrollment E2a_*c*, test set S2b_*c*). The solid line represents actual score data from the experiment while the dashed line represents synthetic score data generated from the normal approximations of the true-speaker and impostor score distributions.

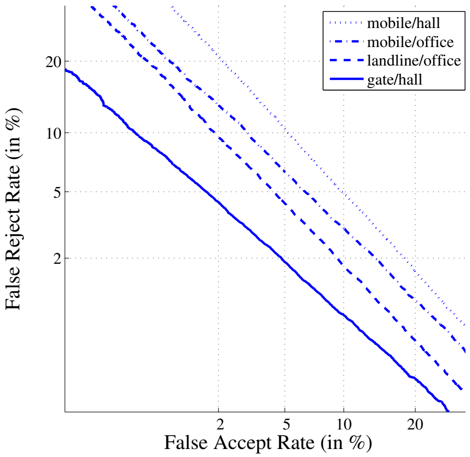


Figure 7.15: A comparison between conditions using synthetic score data generated from normal approximations of the original score distributions from the condition-parallel test sets (S2b_Q:*c*) and the retrained system (cf. Figure 10.3, p. 199). A name plus four digits is used in all conditions. EERp values are 7.2% (MH), 5.8% (MO), 4.6% (LO) and 3.1% (G8).

7.5 Discussion

How much of fluctuations in estimated false reject rates and score distribution parameters in the figures are due to changes in the underlying “true” parameters and how much is due to estimation error?

To investigate if false reject rates are constant or change with time from enrollment we compared false reject rate estimates on chronologically ordered data to estimates on the same data observed in a random order. Figure 7.16 shows average FRRd and FRRp estimates together with the average score mean and variance estimates behind FRRp. The first 100 observations from each of the 17 target speakers with more than 100 available true-speaker tests have been used, irrespective of elapsed time from a target’s enrollment session to the recording of the test utterance, and estimates are computed after every tenth observation using both ML and MAP on accumulated data. The figures show clearly that the average false reject rate increases and the average score mean decreases with time from enrollment. Similarly increasing EERs with time from enrollment were found by Caminero et al. (2002) for “middle-term” (approximately one year from enrollment) vs. “short-term” (up to about half a year). For “long-term” (up to two years) they found that EERs decreased slightly from the “middle-term”.

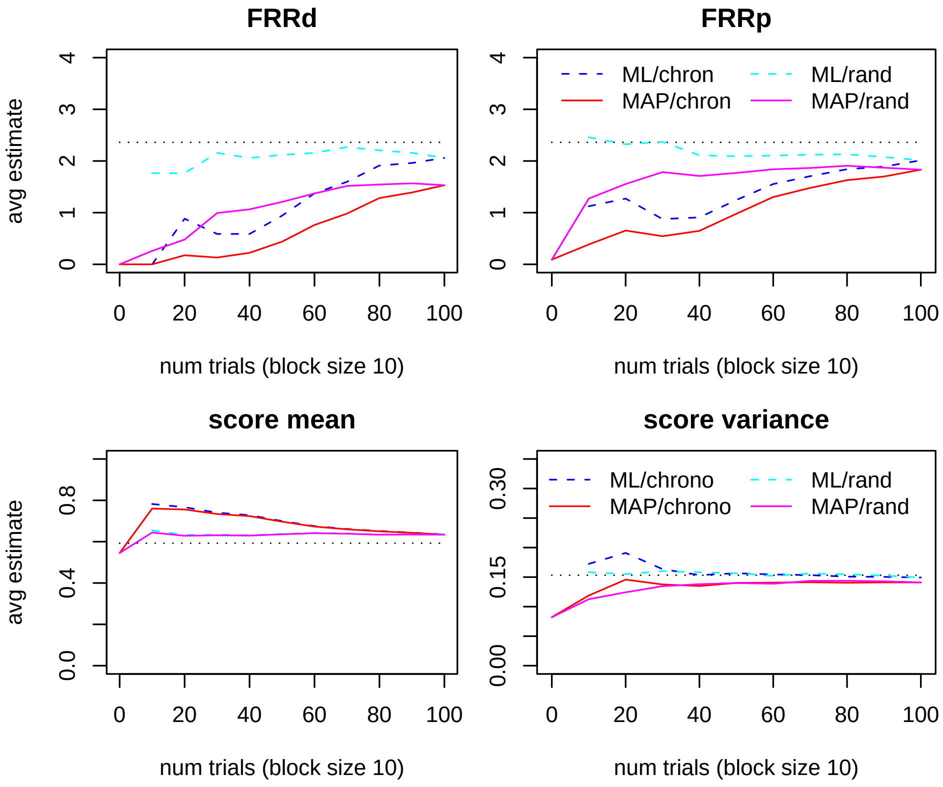


Figure 7.16: False reject rate for blocks of 10 observations seen in chronological vs. random order (upper panges). Estimates by ML and MAP variants of non-parametric (FRRd) and parametric (FRRp) methods with accumulated observations for 17 target speakers in the PER gate/hall condition and the retrained research system. Lower panes show estimates of score mean and variance providing the FRRp estimates. Legend is the same for all graphs.

Part IV

Experiments

Chapter 8

Prompting methods

8.1 Introduction

Speaker verification systems can be classified as being text-dependent, text-independent or prompted. Systems of the prompted class work similarly to a text-dependent system but have the feature that the system prompts the claimant what to say each time the system is used (Higgins et al., 1991).

There are two main reasons for wanting a speaker verification system to prompt the claimant with a new passphrase for each new test occasion: (1) clients do not have to remember a fixed password or passphrase and (2) the system can not easily be defeated by an impostor re-playing recordings of a legitimate user's speech.

In a telephony application the obvious way of prompting passphrases is by presenting them through the telephone with a prompting voice. We refer to this method as *aural prompting* (Doddington, 1985). The prompting voice may be a synthetic voice or recordings of a human voice.

As an alternative, passphrases can be presented to claimants visually, for claimants to read and speak. We refer to this method as *visual prompting* (Doddington, 1985). There are at least three possible approaches for presenting passphrases visually: through the use of on-line displays, password lists or password generators.

With on-site applications, the use of a display is the most straight-forward approach for implementing visual prompting. The application simply displays the passphrase and asks the claimant to speak it. This approach was used for example in the PER application. With telephony applications, on the other hand, the display approach is probably not the best one. While modern mobile telephones have graphical displays, they may not be practical to use for visual prompting of passphrases mainly because, unless a hands-free utility is used, the handset is held by the ear while claimants are talking, and claimants cannot view the display simultaneously. Furthermore, for ASV applications accessed remotely via the telephone, use of the display requires some facility for the application to send passphrases to the display remotely and in real-time. For ASV applications implemented locally in the handset itself, display control should not be a problem. Most existing handsets connected to the landline network do not have a display, and thus visual prompting through a display is not an option with them unless additional hardware is provided.

The next two approaches both involve providing clients off-line with one time passphrases that they are asked to speak. Passphrases can be provided either through a list or by means of a passphrase generator. Both lists and generators have already been used in electronic banking services for many years. For example, Nordea[1] uses lists distributed by mail on a plastic card covered with opaque coating from which users scrape codes one by one, while SEB[1] uses Digipass-type generators that display a code after activation by a PIN-code. Generators can be used in response-only mode or in challenge-response mode. In the latter case, the application provides a challenge code for the user to type into the generator, and the generator then produces the code for the response. In current banking applications, the one time passwords themselves serve for authentication security since only a claimant who has the list or the generator (and knows the PIN to the generator) can produce the right codes (without guessing). If used in conjunction with ASV by asking claimants to speak the one time passwords, we add a biometric layer of security by verifying the claimant's voice. Hence, this way of visually prompting passwords to claimants has a good security potential, with the drawback in user convenience since clients must have the list or generator available when using the application.

This chapter addresses the problem of how to prompt the user with a passphrase by presenting two comparative experiments. In the first experiment (A), visual prompting of four-digit sequences is compared to aural prompting of the same sequences. The second experiment (B) compares the use of four-digit and five-digit sequences as the aurally prompted passphrase. Each experiment is analyzed by looking at the number and type of speaking-errors subjects make while saying the different passphrases, and by comparing the performance of an automatic speaker verification system on passphrases acquired under the various conditions. The verification part is a re-run of experiments presented in (Lindberg and Melin, 1997) with additional ASV systems (those presented in Chapter 3) and with different performance measures: DET curves and EERs or FRRs with speaker-independent thresholds.

8.2 Data

The experiments were conducted on the Gandalf corpus (cf. Section 6.2), i.e., data were not collected during actual usage of a speaker verification system. In the corpus recording, aural prompting was implemented by playing the prompt followed by a 100 ms beep sound. The recording started after the beep and continued during a fixed time interval whose length was determined individually for each type of

[1] Nordea and SEB are large banks operating in Sweden

recorded item. Prompts were synthesized with a rule-based formant synthesis TTS system (Carlson et al., 1982, 1991) to ensure exact reproducibility of the prompting voice. Visual prompting was implemented by printing digit strings on a manuscript that subjects were reading from. The individual digits were separated by a space to indicate they should be read as digits and not as numbers. Comparing to the three approaches for visual prompting presented above, the method used when recording Gandalf data best resembles the password list approach. During a recording session, the four visually prompted items were always recorded before the aurally prompted items.

8.3 Experiment

Two separate experiments were conducted. The first (Experiment A) aimed at comparing speaker verification on digit strings collected through visual prompting vs. aural prompting under the assumption that the prompted digit strings in both cases were “cleanly” produced by subjects and captured by the recording equipment. That is, we wanted to compare how subjects speak in response to the two prompting strategies with respect to speaker verification error rate. In a complementing study we then looked at how often recordings of subjects’ responses contained some kind of error and again compared the two prompting strategies. Errors could be for example digit substitutions, wrong word order or disfluencies, which all appear in the speech signal, but also truncated recordings. We will refer to all such errors as *speaking or recording errors*, or *SREs*. Note that for most types of errors, it is not clear if they are caused by the implementation of the prompting or recording method (for example imperfect speech synthesis or too short recording windows), by inabilities or mistakes in subjects (for example hearing loss or lack of concentration), by disturbances in the subject’s environment, or combinations of the above.

The second (Experiment B) focused on the effect of digit string length with aural prompting under the hypothesis that longer digit strings would generate more SREs in subject responses, while when correctly spoken they would also allow for lower speaker verification error rate due to the availability of more test data. In this experiment we chose to measure the effect of SREs directly through speaker verification error rate, though we have also made a study on a particular kind of SRE, namely word order (or transposition) errors.

In Experiment A, verification tests were made on pairs of visually and aurally prompted versions of the same digit string. A pair was always recorded in the same telephone call and only pairs where both recordings contained precisely the requested four-digit sequence were used (recordings with SREs were sorted out through manual listening). Among the 1850 true-speaker test calls from client subjects in Gandalf there are 455 such pairs that can be used for true-speaker tests. Among those, 405 were chosen, so that to each target there are at least four true-speaker test pairs. This selection gives 69 targets with on average six true-speaker tests per target. For impostor tests, one pair from each of the client subjects plus one pair from each of 37 other subjects were used. Even though test data were selected carefully through the just described procedure, there is still a possible source of bias between the two prompt types in that visual prompts appeared before aural prompts in a all sessions.

The main goal of Experiment B was to compare four-digit vs. five-digit aurally prompted sequences. Since the test material for that comparison must be chosen differently from Experiment A (five-digit speech-prompted sequences are only recorded in the 17th and later test calls in Gandalf, cf. Table 6.6, p. 97), the comparability between results from A and B ran a risk of getting lost. Therefore, visually prompted four-digit sequences were also included in experiment B. Hence, verification tests were made on triples of items recorded during the same telephone call, where each triple contains one visually prompted four-digit sequence plus one four-digit and one five-digit aurally prompted sequence. Two groups of true-speaker test sets were designed from such triples. The first group contained triples with no SREs. These sets were combined with impostor test sets into test sets referred to as B/clean. The second group contained all other triples, namely those in which at least one of the constituents of a triple contained at least one SRE. In other words, triples of utterances from a single call in which for at least one combination of prompting method and passphrase length a speaking or recording error was found. These sets were combined with (the same) impostor test sets into test sets referred to as B/dirty.

The 61 client subjects with 8 or more such triples available were selected as targets in Experiment B. The average number of triples per subject is 17.5 including SREs and 15.5 excluding them. Data for impostor tests were chosen analogously to experiment A; one triple per speaker was chosen with no items in the triple having an SRE. The impostor parts of test sets B/clean and B/dirty are identical, thus any differences measured between the two groups of test sets are due entirely to the true-speaker tests.

The number of speakers and tests used in each experiment is summarized in Table 8.1, where numbers for male and female targets are presented separately. Test sets are exactly those used in (Lindberg and Melin, 1997), except that cross-sex impostor attempts have been omitted.

Target models were built from 25 *visually prompted* five-digit sequences recorded in one session (Gandalf digits enrollment set 1s1h*1, cf. Table 6.8, p. 101). In these 25 sequences each digit occurs at least twelve times and in all left and right contexts.

For studying SREs in visually prompted versus aurally prompted items, all available calls from the 61 subjects used as targets in experiment B were used. The 61 subjects all have recorded at least 20 test calls, thus it should be possible to observe potential learning effects. The total number of calls for this SRE study is 1511, with four visually prompted and two aurally prompted four-digit items in each call. 535 of those calls also contain two aurally prompted five-digit items.

Table 8.1: Properties of test sets used in the verification test part of experiments A and B. Numbers indicate dimensions of each of the prompt-specific test sets, i.e. in Experiment A there are two test sets with the dimensions given in this table, while in Experiment B there are three test sets for B/clean and three for B/dirty. All impostor tests are same-sex tests.

Experiment	A		B/clean		B/dirty	
	male	female	male	female	male	female
targets	39	30	37	24	25	17
true-speaker tests						
per target on average	6.5	5.7	15.5	15.6	3.0	2.8
total number of tests	253	170	573	374	75	48
impostor tests						
additional subjects	25	12	33	10		
per target	63	41	69	33		
total number of tests	2457	1230	2553	792		

8.4 Speaker verification systems

Results are presented in this chapter for four ASV systems. In addition to the HMM and GMM-based systems and their combination described in Chapter 3 of the present thesis, results from the system used in the original experiments reported in (Lindberg and Melin, 1997) have been included. The latter system is referred to as the CAVE-system from the project it was developed within (Bimbot et al., 2000). It is similar to the HMM-based system presented in Section 3.3, but differs in the following major aspects:

1. target and background HMMs have two Gaussians per state (vs. 8)
2. the static part of feature vectors are LPCC-type (vs. MFCC)
3. target models are trained from scratch (vs. using the background models for initial values)
4. target model variances are trained (vs. fixed to the same values as in a background model)
5. the background model is gender-independent (vs. gender-dependent)
6. word-level alignment at test time is produced through Viterbi searches within the target and background models using a common silence/garbage model (vs. external alignment shared between target and background models).

Furthermore, in these experiments, word-level segmentation at training was manual with the CAVE-system (vs. automatic given word-level transcriptions of training utterances) and the CAVE background model was trained on the 15 male and

15 female background speakers in Gandalf (vs. 960 speakers from the SpeechDat corpus). Note that while the CAVE-system is exactly the same as used in (Lindberg and Melin, 1997), performance measures used in this chapter are not the same as in the reference, and hence the numbers are not the same.

8.5 Results

8.5.1 Speaking and recording errors

Recorded items used in the verification part of Experiment A are those where the text content of the recording is exactly that of the prompted text. This section presents some observations on the remaining items, i.e. those with at least one SRE, divided into two groups: those where the passphrase is complete and those where it is not. A passphrase is here considered complete if the requested digits are included in the recording and occur in the correct order.

Table 8.2 shows the frequencies of occurrence of SREs in our data in percent of the number of recorded four-digit items of each prompt type. Statistics are first given for all types of SREs pooled together, then separately for complete and incomplete passphrases, and finally for each type of SRE we have identified.

The division into three groups of test calls (1–4, 5–16, 17–26) in Table 8.2 is somewhat arbitrary, but allows the observation of potential short and long term changes in error rate while subjects get more used to the prompting procedures. The last group (17–26) was chosen to match calls used in Experiment B where five-digit aurally prompted sequences are available.

As can be seen in the table, the recording procedure with aural prompts caused trouble initially. Subjects frequently started speaking but were somehow disturbed by the beep, and re-started saying the whole sequence. Most of those errors could perhaps be eliminated by removing the beep from the prompting procedure.

The lower part of Table 8.2 shows observations from items where the passphrase is not complete. “Digits spoken as numbers” refers to cases like “1 2” spoken as *twelve*, which naturally occur only with visual prompting.

A large portion of the word substitution errors turned out to be confusions between digits 1 ([ɛt$^{\mathrm{h}}$]) and 6 ([sɛks]). Since those errors are likely to have come from misinterpretations of the prompting voice, they are separated from other word substitution errors in the table. Digits 1 and 6 are confused especially in the context *after the digit 6*, e.g. 6-1 ([sɛksɛt$^{\mathrm{h}}$]) was often perceived as 6-6 ([sɛksɛks]). Note that the synthesized speech was played through a telephone line and hence the high-frequency components of /s/ were attenuated. From the 17th test call the sequences with the pair 6-1 were no longer included in the pool of possible prompts and therefore the error rate for word substitution due to the synthesizer decreased considerably.

A detailed study of all types of SREs for experiment B is not given here. Instead, verification results are given below for the cases where SREs are included and excluded respectively. It can be noted, however, that the proportion of five-digit

Table 8.2: Observations on four and five-digit items with some speaking or recording error (SRE). Numbers are given as the percentage of the number of recorded items of a prompt type. Rows with indented left column show a factorization into different kinds of errors. Bold-face numbers indicate errors that are considered systematically related to the prompt type, while other errors are more related to the particular implementation used when recording the Gandalf corpus.

number of digits/prompt type	4/visual			4/aural			5/aural[a]	5/aural[b]
range of test calls:	1–4	5–16	17–26	1–4	5–16	17–26	17–26	17–26
speaking or recording error (SRE)	1.5	0.86	0.74	15	5.9	2.4	10	8.4
passphrases complete	0.20	0.38	0.28	10	1.4	1.2	2.8	2.0
recording method				8.81	1.02	0.47	0.37	0.56
other	**0.20**	**0.38**	**0.28**	**1.19**	**0.41**	**0.74**	**2.4**	**1.5**
passphrase incomplete	1.33	0.48	0.46	5.1	4.5	1.2	7.5	6.4
digits spoken as number	**0.10**	**0.07**	**0.09**					
word subst. due to subject	**0.00**	**0.03**	**0.14**	**0.82**	**0.68**	**0.19**	**1.3**	**0.19**
word subst. due to synthesizer				3.07	2.73	0.46	0.93	1.9
wrong word order				**0.41**	**0.75**	**0.56**	**5.0**	**3.2**
recording method	1.13	0.20	0.09	0.00	0.07	0.00	0.00	0.37
omitted word	**0.00**	**0.03**	**0.05**	**0.82**	**0.27**	**0.00**	**0.19**	**0.75**
other	**0.10**	**0.14**	**0.09**					
sum of bold-face factors	**0.20**	**0.27**	**0.37**	**2.05**	**1.70**	**0.75**	**6.5**	**4.2**

[a] prompt played once

[b] prompt played twice

Table 8.3: Distribution of word order error rates over subjects in four and five-digit aurally prompted sequences. A table cell shows how large fraction of subjects made word order errors at a rate within the given range with a given prompt type. Results are based on 86 subjects in the four-digit case and 63 subjects in the five-digit case.

	Test set (number of digits/prompt type)		
Range of error rate (%)	4/aural	5/aural (single prompt)	5/aural (double prompt)
–5	97%	73%	76%
6–15	3%	6%	14%
16–25		14%	10%
26–35		5%	
36–		2%	

items where the passphrase is incomplete is as high as 7.5% in response to a single prompt and 6.4% with the prompt played twice, to be compared to the 1.2% for four-digit aurally prompted items (single prompt) in calls 17–26 in Table 8.2.

8.5.2 Word order speaking-errors

This section presents some observations on a particular kind of SRE, namely word order (or transposition) errors made by subjects in response to aural prompts. Results reported here are based on all available Gandalf true-speaker test calls.

8.5.2.1 Four-digit sequences

The data set is 1850 calls from 86 subjects with two four-digit aurally prompted items in each call. The number of items with a word order error is 28, or 0.8%. Their distribution over subjects is shown in Table 8.3 together with the corresponding distributions for five-digit strings. Figure 8.1 shows the same distribution graphically with higher histogram resolution than the table.

Table 8.4 shows how often digits in certain positions of the phrase are swapped. The 'other'-entry for four-digit sequences in the latter table represents two cases with a swap between positions #1-#3. It is clear that a swap between the two middle digits is the most common word order error.

8.5.2.2 Five-digit sequences

The data set is here 539 calls from 63 subjects with two five-digit aurally prompted items in each call. Those two items have a difference: the first has a single prompt while the second has a double, i.e. the prompt was played twice to the subject. The

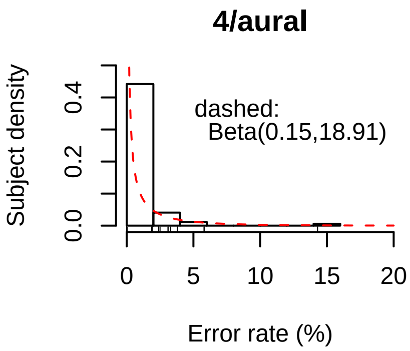


Figure 8.1: Distribution of word order error rate with aurally prompted four-digit strings (4/aural) over 86 subjects. Dashed curve shows the best fit beta distribution.

Table 8.4: Distribution of word order errors on positions of the four and five-digit aurally prompted sequences. The *Positions*-columns show how large fraction of the errors occurred as swapping of digits in the two positions given in the *Positions*-columns. An example of an error which is sorted into the "#2-#3" row for four-digit sequences is '1 $\underline{5}$ $\underline{3}$ 0' → '1 $\underline{3}$ $\underline{5}$ 0'. The total number of word order errors are 28 for four-digit strings and 50 for five-digit strings.

	Test set (number of digits/prompt type)			
	4/aural		5/aural	
	Positions	Proportion	Positions	Proportion
initial	#1-#2	7%	#1-#2	6%
medial	#2-#3	82%	#2-#3	32%
			#3-#4	46%
final	#3-#4	4%	#4-#5	6%
other		7%		10%

fraction of items with a word order error is 5.9% for a single prompt and 3.3%[2] for a double prompt. The distribution of errors over subjects is shown in Table 8.3 for single and double prompts respectively. Note that the statistics on individual speakers is weak since the number of recorded items of an aurally prompted five-digit sequence is small, 20 for 42 of the subjects and less for the remaining 21 subjects.

Table 8.4 shows how often digits in certain positions of the phrase are swapped. The 'other'-entry for five-digit sequences in the table contains five errors which all

[2] These fractions are larger than those reported for word order errors in Table 8.2 because some utterances with a word order error also contained other types of errors and were counted under other categories in the table.

Table 8.5: Speaker verification EERs for the two test sets in experiment A with four ASV systems.

System	Test set (number of digits / prompt type)	
	4/visual	4/aural
CAVE	7.2%	8.9%
HMM	5.4%	7.2%
GMM	9.1%	10.7%
Combined	5.2%	7.2%

can be described as swapping groups of digits, where in the observed cases one group is two digits and the other group is one digit. An example of such an error is '6 $\underline{3}$ $\underline{4\,1}$ 8' → '6 $\underline{4\,1}$ $\underline{3}$ 8'. In four of the five cases the middle groups are swapped (as in the example), while in the fifth case the two first groups were swapped ('$\underline{4\,0}$ $\underline{9}$ 8 7' → '$\underline{9}$ $\underline{4\,0}$ 8 7'). The formulation *swapping two groups of digits not including the first and the last digit in the sequence* hence covers as much as 86% of word order errors for the five-digit sequences. The same formulation degenerates to the #2-#3-case for four-digit sequences, which corresponds to 81% of those sequences.

Similar observations on transposition errors have been made in research on short-term memory retention. For example Bjork and Healy (1974) found that items, in their case consonant letters, in non-initial and non-final position were more likely to be recalled in the wrong order than initial and final items. Furthermore, they found that order information was lost more quickly than item information. Their experiments were conducted with visual prompts. Subjects read a letter sequence aloud, followed by a number of digits, and was then asked to recall the letter sequence. While there are a number of differences between the task their subjects were tested for and the task of repeating an aurally prompted digit sequence, results are indeed similar, suggesting that general properties of human short-term memory may explain our observations on SREs.

8.5.3 Speaker verification performance

Experiment A. Table 8.5 presents speaker verification EERs (using speaker-independent thresholds as opposed to in (Lindberg and Melin, 1997)) for each of the two test sets in Experiment A for the four ASV systems, and Figure 8.2 shows the corresponding DET curves for the HMM subsystem (Figure D.1 in Appendix D includes DET curves for all four systems). To show what happens with the balance between false accept and false reject rates when the prompting method changes, the FAR/FRR value pair for a specific threshold has been marked in DET plots, namely that for the threshold corresponding to the *a posteriori* EER threshold on the case with visually prompted digits. Denote this threshold $\theta^{\mathrm{v}}_{\mathrm{EERd}}$. The threshold is determined individually for each ASV system.

For all four systems, the EER is lower for visually prompted utterances than for

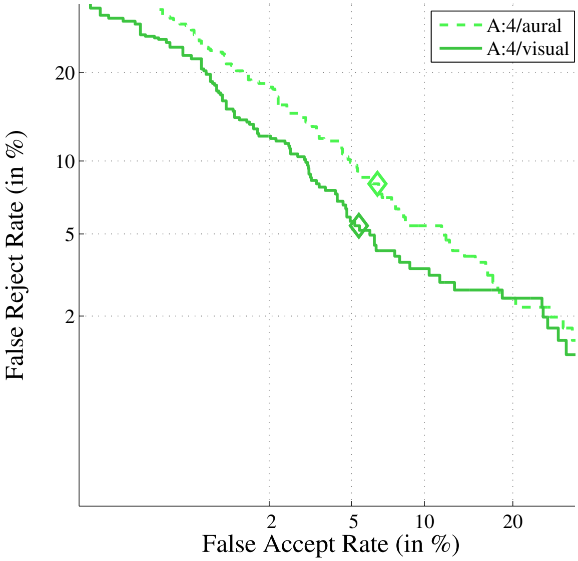


Figure 8.2: DET curves from Experiment A for the HMM subsystem and aurally vs. visually prompted 4-digit strings. Diamonds indicate the FRR/FAR pair for a threshold $\theta^{\mathrm{v}}_{\mathrm{EERd}}$ determined as the EER threshold on the visually prompted data.

aurally prompted ones, and DET plots show this trend holds along the entire range of operating points. Looking at all the systems together (Figure D.1) it seems that the FRR is more affected than the FAR by the change in prompting strategy, since the operating point marked by diamonds is shifted consistently along the FRR axis, while the shift along the FAR axis is smaller and with varying sign. This implies that the effect of changed prompting strategy is subordinate to speaker identity in the speaker verification process, a result that is comforting but not surprising.

To further investigate on the observed difference in overall FRR between the two prompting methods, we look at distributions of error rate over targets (cf. Lindberg and Melin, 1997). First, we look at (non-parametric) FRR (FRRd) at the same threshold $\theta^{\mathrm{v}}_{\mathrm{EERd}}$ shown in the DET plots. Second, as an indication of speaker-discriminability we look at individual EER with *a posteriori* target-specific, prompt method-dependent thresholds. In this case, EERs are estimated with the parametric ML method (EERp) presented in Section 7.3.2 (p. 137).

Figure 8.3 shows the distribution of individual FRRd and EERp for the two contrasting test sets and the HMM subsystem, while Figure 8.4 shows the distribution of *changes* in individual error rates when going from visual prompts to aural prompts. Figures D.2 through D.5 in Appendix D include the corresponding distributions for all four systems. Figures show that for a majority of targets, FRR does not change at all with prompting method. For targets where there is a difference, a

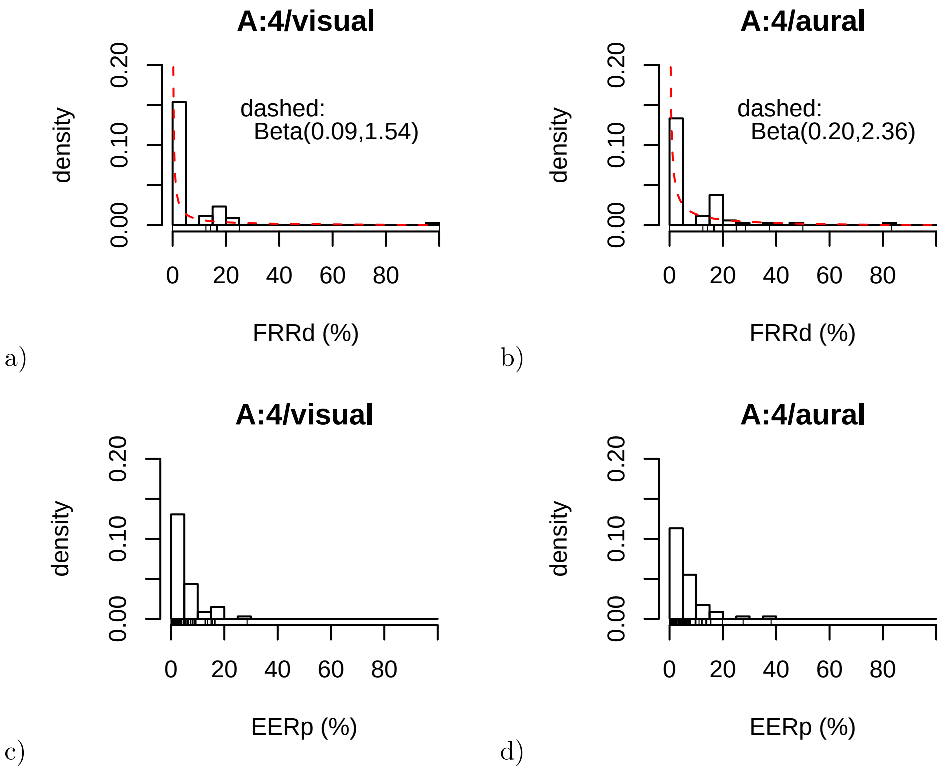


Figure 8.3: Distribution of non-parametric FRR (FRRd) at a fixed target-independent threshold $\theta^{\mathrm{v}}_{\mathrm{EERd}}$ determined as the EER threshold on the visually prompted data (a,b), and EERp (target-dependent *a posteriori* thresholds) (c,d) over targets in Experiment A (visually vs. aurally prompted 4-digit strings) with the HMM subsystem. Dashed lines in a) and b) show fitted beta distributions (cf. Section 7.3.1.2, p. 135).

majority see an error rate increase when going from visual to aural prompts. The situation is similar for EERp, suggesting that speaker discriminability decreases with aural prompts relative to visual prompts.

The lack of bars for small FRRd differences in Figure 8.4a is due to differences being quantized into steps the size of $1/n_i$, where n_i is the number of tests available for target i. The average number of tests per target in Experiment A is 6.5 for male and 5.7 for female targets (cf. Table 8.1), hence the clusters of bars around ±17%. An alternative form of presentation would be a bar chart that shows differences in the *number* of errors instead of differences in error *rate*.

In Section 10.5.1.1 (p. 210), we show how a non-parametric test called *McNe-*

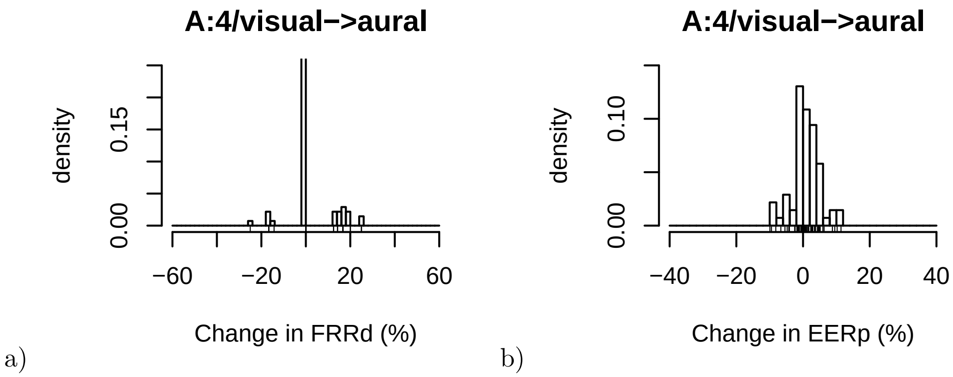


Figure 8.4: Distribution over targets of differences in a) FRRd at a fixed target-independent threshold $\theta^{\mathrm{v}}_{\mathrm{EERd}}$ determined as the EER threshold on the visually prompted data and b) EERp (target-dependent *a posteriori* thresholds), when comparing the two sets in Experiment A with the HMM subsystem. A positive difference x indicates that the individual FRRd (EERp) for aurally prompted items was x units higher than for visually prompted items.

mar's test for the significance of changes (e.g. Siegel, 1956) can be used to test for differences between overall FRR. The same method can be applied here. In fact, the test, as applied in Section 10.5.1.1, is basically a formalization of the comparison we made above on Figure 8.4a. Under the null hypothesis that there is no difference between the two prompting methods, the test checks if the number of targets that appear to the left in the histogram is sufficiently different from the number of targets that appear to the right, not to be a result of chance. The McNemar test indicates a difference (at 5% level of significance) in FRR between the visual and aural prompts in Experiment A for the GMM subsystem and the combined system, but not for the CAVE and HMM systems. For EERp, a significant difference could not be detected for any of the ASV systems (contrary to our graphical interpretation of figures). Table 8.6 shows detailed results of the McNemar tests.

Table 8.6: Results of McNemar's test of differences at 5% level of significance.

case A	case B	system	p[a]	diff[b]
Effect 1: *FRRd at fixed target-independent, system-specific threshold* θ^v_{EERd}				
visual	aural	CAVE system	0.14	-
		HMM subsystem	0.044	-
		GMM subsystem	0.007	x
		combined HMM and GMM system	0.004	x
Effect 2: *EERp with target-dependent* a posteriori *threshold*				
visual	aural	CAVE system	0.63	-
		HMM subsystem	0.15	-
		GMM subsystem	0.34	-
		combined HMM and GMM system	0.23	-

[a] probability that test statistic x has observed value T_{χ^2} or greater ($P_{\chi^2}(x >= T_{\chi^2})$)

[b] 'x' indicates a statistically significant difference detected by a two-sided test at $\alpha = 0.05$

Experiment B. Table 8.7 shows FRR for Experiment B for the four ASV systems and the B/clean and B/dirty groups of test sets. The decision threshold for each combination of system and test set is determined as the *a posteriori* EER threshold for the B/clean version of the test set. FRRs are presented rather than EERs because impostor parts of the test sets are identical, thus differences between "clean" and "dirty" test sets lie only in the true-speaker parts of test sets. The table also shows results for the B/clean and B/dirty test sets pooled (B/all). These test sets contain a naturally occurring blend of correctly spoken utterances and utterances with SREs for the respective prompting methods and passphrase lengths. Thus, by comparing the prompting methods and passphrase lengths through the B/all test sets, we take into account how often SREs occur, how many verification errors they cause and verification performance in the SRE-free utterances.

Figure 8.5 shows DET curves for Experiment B with the HMM subsystem (DET curves for all four systems are included in Appendix D). The figure contains four panes. The first three compares each of the three test sets for the B/clean and B/dirty groups, and the pooled B/all. The final pane compares the three groups for a single test set, namely that with visually prompted 5-digit strings. While FRRs in Table 8.7 are computed from an EER threshold on the 4/visual test set, the diamonds included in DET plots mark the operating point determined by the EER on the 4/visual test set, like in Figure 8.2.

Results for aurally prompted digit strings show that 5 digits perform slightly better across all tested ASV systems, given that utterances are correctly spoken and recorded (B/clean test sets). On the B/dirty test sets there is no clear difference (note that there are only 123 true-speaker tests per test set, so the statistical uncertainty in DET curves is much larger than in the B/clean case). With SREs taken into account at their natural frequency of occurrence (B/all), there is again no consistent advantage in error rate for any of the two passphrase lengths across our ASV systems. Thus, the advantage for 5-digit strings on clean data seems to have been consumed by the effect of SREs.

Comparing visually and aurally prompted 4-digit strings in Experiment B, there appears to be no advantage for visual prompts on clean data like there was in Experiment A. On the B/dirty test sets, visual prompts show smaller error rates (still with a larger statistical uncertainty) than aural prompts, in particular for the two HMM-based systems. In the pooled B/all case, no clear advantage is shown. Note that in Experiment B pairs of visually and aurally prompted 4-digit strings do not necessarily contain the exact same digit sequence like in Experiment A, and that visually prompted strings are drawn from a pool of only four different sequences, while aurally prompted sequences are drawn from a pool of 20 sequences (see Section 6.2.2.5, p. 96).

Finally, note that the large differences in error rate for four-digit visually and aurally prompted sequences in Experiments A and B (Tables 8.5 vs. 8.7) come from the fact that A and B have very different test sets. B includes only calls from the same handset, the so called favorite handset (cf. Section 6.2), while A includes calls from many different handsets. The error rates in experiment A are therefore

Table 8.7: Speaker verification FRR for the two groups ('clean' and 'dirty') of three test sets in Experiment B with four ASV systems, and the two groups pooled ('all'). Decision thresholds are determined as the *a posteriori* EER threshold on the 'clean' version of a test set for each system and test set. Note that the impostor part of test sets is the same in all groups.

System	Group	Test set (number of digits / prompt type)		
		4/visual	4/aural	5/aural
CAVE	B/clean	4.9%	5.2%	4.4%
	B/dirty	2.4%	8.1%	6.5%
	B/all	4.6%	5.6%	4.7%
HMM	B/clean	3.3%	3.3%	3.4%
	B/dirty	4.1%	9.8%	11.4%
	B/all	3.4%	4.2%	4.2%
GMM	B/clean	6.2%	6.3%	5.0%
	B/dirty	8.1%	10.6%	17.9%
	B/all	6.4%	6.7%	6.5%
Combined	B/clean	3.2%	3.3%	3.0%
	B/dirty	3.3%	6.5%	12.2%
	B/all	3.2%	3.6%	4.0%

generally higher.

8.6 Discussion

In experiment A, the overall EER for visually prompted sequences was lower than for aurally prompted sequences. One should keep in mind, though, that target models were trained on visually prompted speech. The result can be interpreted such that there is *a difference* in how subjects speak a phrase when it is given to them through visual rather than aural prompts. It is not clear that visual prompts generally provide for better results than the aural prompts. If the target models in the case of aural prompts were also trained on aurally prompted speech, the result would probably be different.

The difference in DET curves between visual and aural 4-digit prompts is larger in Experiment A than in the 'clean' part of Experiment B (Figure 8.2 vs. Figure 8.5a). In Experiment A, which was designed for exactly this comparison, utterances within each pair contain the exact same digit sequence, and therefore results from A should be more reliable in this respect. Furthermore, results in Experiment A are based on more target speakers than Experiment B suggesting also that the comparison in Experiment A is more reliable (even though there are more tests per target in Experiment B; cf. the discussion in Section 10.5.1).

In addition to these differences, we can also identify a number of systematical differences in the data sets used in the two experiments, such as the number of

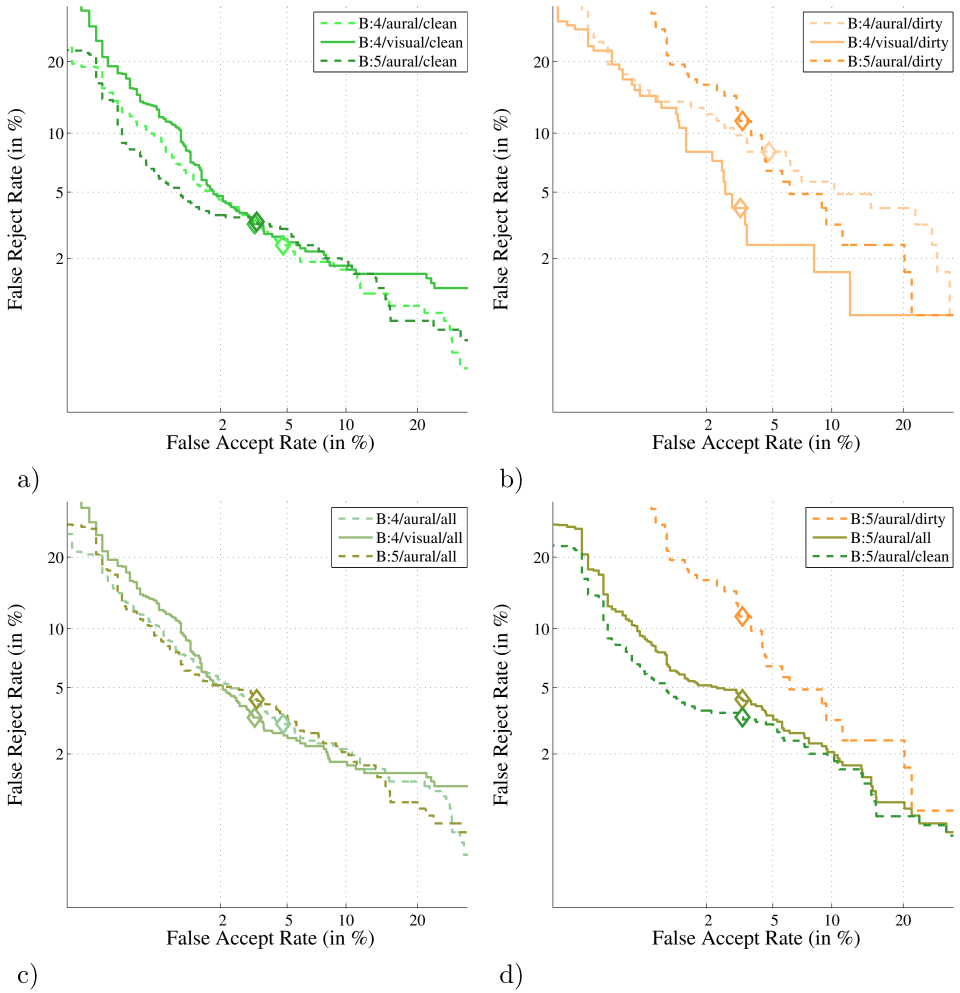


Figure 8.5: DET curves from Experiment B with the HMM subsystem (corresponding plots for the other systems are included in Appendix D): a) B/clean test sets, b) B/dirty test sets, c) B/clean and B/clean merged into B/all, d) DET curves from a through c for aurally prompted 5-digit strings drawn together.

used handsets, slightly different sets of target speakers, the number of different digit sequences used, etc. A difference we find interesting, and that may have contributed to the decreased difference between visual and aural prompts between Experiments A and B, is that Experiment A uses test data recorded during the first four months of the data collection, while B uses test data recorded from month seven and later. Hence, there may be a learning or normalization effect such that subjects have for example "learned" to speak equally natural in response to both visual and aural prompts.

In the comparison between four and five aurally prompted digits, results considered for all four ASV systems as a group showed that the advantage of more test data in the longer utterances was consumed by the higher frequency of SREs. This finding is different from that in (Lindberg and Melin, 1997), which was based on the CAVE system only and where we found that the longer digit sequences performed better also when including SREs. In this revised study (with other performance measures), the CAVE system still performs better with the longer digit sequences, but it was the only system that did so.

The portion of SREs is a measure of how often an ASV system would have to give the claimant a new try just because the passphrase was wrong. The only systematic sources of SREs related to visual prompts seems to be reading disfluencies (included in the "other" group in Table 8.2) and digits pronounced as numbers. Observed error rates for both are very small relative to the EER of the used ASV systems. For aural prompts, observed SRE rates were higher and of the same order of magnitude as the EERs. In this case, SREs seem to have come from two sources: either the subject did not hear the prompt correctly, or the short-term memory failed him and he repeated the wrong sequence, either with the wrong word order or with word substitutions. In particular the word order error rate in response to aurally prompted five-digit sequences was large, 5% with a single prompt and 3% when the prompt was played twice.

Since longer test utterances still have greater potential for lower error rates; how do we get rid of SREs or how do we deal with them? We envisage four approaches. The first approach is to use visual prompts instead of aural prompts. We saw fewer SREs with visual prompts than aural prompts with four-digit sequence, and we believe the difference would be even larger with five-digit sequences. While not tested explicitly, this belief is supported by experience from the PER experiments presented in Chapter 10 where five-digit strings were visually prompted to claimants. If we need to use aural prompts, the three main approaches are to make the ASV system reject SREs, to make it deal with them, and/or to lower the frequency of generated SREs.

SREs can be rejected by working with speech recognition or text verification techniques (e.g. Li et al., 2000) to check that a recorded response contains exactly the prompted digit sequence and nothing else. One such approach was successfully implemented in the PER system, where an utterance was rejected if the prompted digit string did not show up in a (short) N-best list produced by the speech recognizer.

A key to making (word-based) ASV systems more robust to SREs may be in how word alignments are produced and used. For example, one of the differences between the two HMM-based systems in the present study is in alignments. The CAVE system produces word alignment separately within the target and background models using the target and background HMMs themselves, while the HMM system developed in this thesis uses a single external alignment that is produced by a speech recognizer and that is shared between target and background models. This difference could be one reason to why the CAVE system showed better robustness to SREs in our experiment. Another suggestion for increased SRE robustness is to smoothly accept word order errors by claimants in the verification process.

The frequency of generated SREs with aural prompts can be reduced by increased intelligibility in the prompting voice (improved synthetic voice or recordings of a human voice), by avoiding digit combinations with higher probability of generating confusions (for example the combination "6 1" was often perceived as "6 6" in our data), and possibly to play a prompt to claimants twice. To repeat the prompt is a rather extreme measure, but it may be useful as part of a back-off strategy when a dialog system detects SREs during several consecutive attempts in a verification dialog. Along with improving the aural prompting mechanism, the "recording error" part of SREs can be reduced by improving the recording mechanism. In the present study we used a fixed-time recording window. While this technique is easy to implement and unaffected by background noise, it is of course inflexible. The time window needs to be sufficiently large to capture the majority of responses and thus results in unnecessarily slow response times from the ASV system for most claimant responses. Controlling the recording window adaptively through the use of a good speech detector or feedback from an incremental speech recognizer may be a better option. In the telephone part of our subsequent PER experiment we used strings of four rather than five digits to be on the safe side. We also used a diphone-type synthetic voice (instead of the formant-based synthesis used in the Gandalf collection).

An alternative to increasing the length of aurally prompted digit strings is to use short digit strings with a sequential decision strategy as discussed in Section 10.5.2 (p. 211).

Word order error rate in responses to aurally prompted digit strings was found to spread unevenly over subjects as shown by Table 8.3 and Figure 8.1. This opens a possibility for exploiting individual word order error rate as a feature in speaker recognition. For example, an ASV system could try to estimate the word order error rate in a target using the MAP approach presented in Section 7.3.1 (p. 134) (Figure 8.1 shows a nice fit for a beta distribution that could be used as a conjugate prior for a binomial distribution of the number of observed word order errors). Given an estimated word order error rate in a target and in a general background population, a likelihood ratio for an observed word order error, or the lack of one, can then be calculated. Such a likelihood ratio could then perhaps be used in combination with other ASV techniques such as those presented in Chapter 3 of this thesis, or fit into a multi-classifier recognition framework.

8.7 Conclusion

From the study of speaking and recording errors (SREs) in response to visual and aural prompts respectively, it is clear that aural prompts leave the claimant with a more difficult task and more SREs are therefore produced. While SRE rate for aurally prompted four-digit strings was not critical, increasing the digit string length to five resulted in an SRE rate in the same order of magnitude as EER for a speaker verification task on SRE-free data for the tested ASV systems. In particular, word order errors were found to be a significant source of error in aurally prompted five-digit strings. We conclude that aural prompts, if used, should be limited to four digits.

As a means for prompting passphrases visually, password generators, like those currently used in many electronic banking services, may offer a very high security potential if claimants were asked to speak a generated code. To the current security in legitimate users already *having* the unique generator and *knowing* its activation PIN, speaker verification would add a biometric layer of security.

Speaker verification experiments on visually and aurally prompted passphrases indicate that there may be a (small) difference between speech produced in response to the respective prompt types, which affect the performance of both HMM and GMM-based speaker verification system. Since all our experiments were made with enrollment speech elicited through visual prompts, we were not able to establish that one prompt type results in speech more suitable for ASV than the other, only that given visual prompts at enrollment, there is a small advantage for visual prompts over aural prompts at test time in terms of ASV error rate. It seems likely that from an ASV error rate point of view it is better to use matching prompt types at enrollment and test, since all similarity based ASV systems are more or less sensitive to mismatched conditions. In choosing between aural and visual prompts, however, we believe other considerations than ASV error rate differences (on SRE-free data) will be more important in practice, such as application context, user preference and re-prompt rate due to SREs.

Chapter 9

Variance estimation

9.1 Introduction

In practical applications ASV systems are generally used in contexts where very few target enrollment data are available. One problem with using small training data sets is the risk of over-training, that is, parameters of the target model are over-fitted to the particular training data. Especially variance parameters are susceptible to over-fitting: a variance estimated from only a few data points can be very small and might not be representative of the underlying distribution of the data source.

The maximum likelihood (ML) principle is often used in training parameters of continuous density hidden Markov models (HMM). The most general implementation of that principle (the EM-algorithm) consists in optimizing all parameters of the HMM, including means and variances of state pdfs. With sparse training data from an enrollee, target model variances tend to be over-trained (Bimbot et al., 2000).

In this chapter[1] we look at three heuristic approaches to robust estimation of target model variances in the context of word-level, left-right HMM-based speaker verification. The three approaches are referred to as *variance flooring*, *variance scaling* and *variance tying* and they can all be viewed as modifications of the ML/EM-algorithm.

By variance flooring, we modify the EM-algorithm to impose a lower bound on variance parameters, a variance floor. With this method any given variance value will have its corresponding floor value as a lower bound during iterations of EM. A new problem is then how to compute this floor value. Within the CAVE-project a method to compute variance floors is suggested (Bimbot et al., 2000). With this method, all variance vectors of all HMMs in a speaker model share one flooring vector. This vector is estimated as the variance over some calibration data set

[1] Experiments and most of the results in this chapter have previously been published in (Melin et al., 1998; Melin and Lindberg, 1999b).

multiplied by a constant variance-flooring factor. The calibration data set can be for instance the same data used to train background models.

Variance flooring can be implemented with several levels of "resolution" in up to three "dimensions". The first dimension is the vector index, where resolution can range from a scalar floor, where all components of a variance vector share a floor value, to a floor vector where each component has its own floor value. The second dimension is time (represented by a state sequence in a left-right HMM) where a unique floor can be shared by variance vectors within all states in all models, ranging to each state having its own floor. The third dimension is feature space, where different parts of the feature space may have their own floor. An example of the latter is when each Gaussian term within a composite pdf has its own floor value.

An alternative modification to the EM algorithm is to keep variances fixed while updating means and transition probabilities (Matsui and Furui, 1993). In the context of speaker verification, where a background model is often used for likelihood normalization, the variances of the target model can be copied from the background model. A background model is often trained on a lot of data from many speakers and all parameters of the model can be reliably estimated with the original EM-algorithm. If background model variances are used systematically in target models, target model variances become target-independent. This modification can be generalized by setting target model variances proportional to variances in the background model. We refer to this approach as variance scaling.

Through variance tying the number of variance parameters to estimate can be reduced. Like variances can be floored at different resolutions, as introduced above, they can also be tied at the same "resolutions" along the same "dimensions", for example all mixture components sharing a variance vector within an HMM state or within an entire HMM. The purpose of reducing the number of variance parameters is to allow the remaining parameters to be robustly estimated.

In this chapter we compare several variations of the three principle modifications to the EM-algorithm mentioned above. The comparison is made on three separate telephone quality corpora in three different languages. The recognition tasks are slightly different, but are all some form of text-dependent task using digits.

From the variety of possible variance flooring methods we try three variants with gradually increasing resolution: model-dependent, state-dependent and mixture component-dependent vector floors. The various floor vectors are computed as an empirical constant times a basis vector, like in (Bimbot et al., 2000). The basis vector is derived from speech data or directly from a multi-speaker model. We look empirically at verification error rate as a function of the scale factor to see if there is a minimum for some value. We then do similar experiments with variance scaling. Finally, we reduce the number of variance parameters by tying variance vectors across mixture components within each state. The target models have eight mixture components per state, and by tying variances within states we reduce the number of variance parameters by a factor eight. We compare tied-variance models with the original ones for the various variance estimation methods to see if the smaller number of variances can be more robustly trained.

Since the variance flooring and scaling techniques involve the setting of an empirical constant, its usefulness depends on to which extent the choice of an optimal scaling factor will generalize from development data to new evaluation data. We present a series of experiments to investigate on such generalization properties.

9.2 Data

Three corpora have been used in the tests: Gandalf (cf. Section 6.2), SESP (Boves et al., 1994; Bimbot et al., 2000) and Polycost (Hennebert et al., 2000). Furthermore, the development and evaluation parts of Gandalf were used separately in this chapter as if they were two different corpora. All corpora contain digital telephony data recorded through the ISDN. Table 9.1 summarizes the main features of the corpora. The notation used for enrollment sets is that introduced for Gandalf in Section 6.2.4, i.e. NsMh$*t$, where N is the number of sessions, M the number of handsets, and t is the approximate (effective) amount of speech in minutes. The norm for the amount of speech is Gandalf where 25 five-digit sequences are estimated to one minute of speech (one digit is 0.5 seconds).

Segmentation of speech data into words is made on a per-utterance basis with a speech recognizer operating in forced alignment mode given the manuscripted text.

9.3 Speaker verification systems

Experiments in this chapter are made with variations of the word-level HMM system described in Section 3.3 (p. 40), where variations include a number of different ways to estimate target model variances. In addition to these variations, systems used in this chapter all use a fixed background model for score normalization during verification tests selected for each target during enrollment as in (Melin and Lindberg, 1999b) (in Section 3.3 the background model was instead re-selected at each new verification test). In this chapter, the HMM system described in Section 3.3, where target model variances are copied from the male or female background model (variance scaling with scale factor 1.0), and with the fixed selections of background model, is referred to as the *baseline system*.

9.3.1 Parameter estimation

In the baseline system, a background model is selected individually for each target speaker and each word during enrollment as one of two competing gender-dependent multi-speaker models, with no *a priori* information on the gender of the enrollee. When training the target model, the best matching multi-speaker model is copied as a seed for the target model.

Target model means and mixture weights are always estimated from enrollment data with the ordinary EM equations while transition probabilities are kept con-

Table 9.1: Summary of main features of the three corpora and their protocols used in variance estimation experiments. Number of speakers are given as #male/#female.

	Test corpus	Gandalf		Polycost[a]	SESP
	Set	dev-set	eval-set		
Task	language	Swedish		English	Dutch
	native speakers	100%		15%	100%
	enrollment	1s1h*1.0		2s1h*0.2	4s2h*0.9[b]
	password	2 x 4 digits		10 digits	14 digits
Test data	clients	22 / 18	24 / 18	61 / 49	21 / 20
	impostors	23 / 18	58 / 32	61 / 49	21 / 20
	total number of true-speaker tests	927	886	664	1658
	impostor tests (same-sex)	790	1926	824	763
Background data	corpus	SpeechDat		Polycost	Polyphone
	speakers	399 / 561		11 / 11	24 / 24
	total time (approx.)	5 h		0.5 h	0.3 h
	examples per digit and speaker	4		19	5

[a] Polycost test was baseline experiment 2, version 2.0, as defined in (Nordström et al., 1998).

[b] This enrollment set is referred to as G in previous literature (Bimbot et al., 2000). The number of handsets is an estimate.

stant. Target model variances are estimated with one of two alternative methods. To define those methods we denote as $\boldsymbol{\sigma}^2_{wjk}$ the variance vector of target model w, state j and mixture component k; as $\mathbf{s}^2_{wjk}$ the corresponding variance vector of the seed model or some calibration data; and as α or γ a scalar, system-global scale factor. In the first method, referred to as *scaled variances*, target model variances are inferred directly from the seed variances (9.1), and no training on enrollment data is involved. The second method is *variance flooring* where variances are trained from enrollment data with a constraint on the minimum variance as given by (9.2), which is applied after every iteration of the (modified) EM algorithm. Note that with $\gamma = 0$ this method converges to the original EM algorithm.

$$\boldsymbol{\sigma}^2_{wjk} = \alpha \cdot \mathbf{s}^2_{wjk} \tag{9.1}$$

$$\boldsymbol{\sigma}^2_{wjk} \geq \gamma \cdot \mathbf{s}^2_{wjk} \tag{9.2}$$

9.3.2 Tied variances

To reduce the number of parameters to train, the variances of a set of state distributions can be tied to a single vector. We use a letter-pair $\vartheta_v = a/b$ to indicate the "level" of tying, where a indicates tying in the target model and b in the background model. Letters a and b can take symbols in an ordered alphabet $\Lambda = \{\mathrm{X}, \mathrm{S}, \mathrm{M}\}$, where X indicates one variance vector per mixture component (no tying), S one vector per state, and M one vector per model. With $\vartheta_v = \mathrm{S/S}$, equations (9.1) and (9.2) are still valid if we remove index k. If $\vartheta_v = \mathrm{S/X}$, on the other hand, we need to compute a state variance from mixture component variances in the background model. We try a simple heuristic approach by estimating a state variance through a linear combination of mixture component variances. Eqs. (9.3) and (9.4) are then our modifications of (9.1) and (9.2) for the case $\vartheta_v = \mathrm{S/X}$, where c_k is the mixture weight for component k.

$$\boldsymbol{\sigma}^2_{wj} = \alpha \cdot \sum_k c_k \mathbf{s}^2_{wjk} \tag{9.3}$$

$$\boldsymbol{\sigma}^2_{wj} \geq \gamma \cdot \sum_k c_k \mathbf{s}^2_{wjk} \tag{9.4}$$

With variances tied across entire target HMMs ($\vartheta_v = \mathrm{M/X}$), the basis vector $\mathbf{s}^2_w$ for variance flooring in model w is computed directly from the background speech data used to train the corresponding background HMM. Equations (9.1) and (9.2) are then used with $\mathbf{s}^2_w$ in place of $\mathbf{s}^2_{wjk}$.

9.3.3 Tied variance floors

To summarize information that was already given above, basis vectors for variance flooring are derived in one of the three following ways:

- model-dependent floor: the basis-vector for word model w is the variance of all feature vectors within background data segments identified as the corresponding word, from speakers of the same gender as the automatically detected gender of the enrollee
- state-dependent floor: the basis-vector for a state j in word model w is computed as a linear combination of variance parameters of the Gaussian mixture in state j in a background model
- mixture component-dependent floor: the basis-vector for a mixture component k in state j of model w is the variance of mixture-component k in the corresponding state of a background model.

The "resolution" of a variance floor introduced above can be described in the same framework as in the previous section if the variance floor vector is viewed as a tied vector. We can then define another letter-pair ϑ_f to denote the tying level of the variance floor. The variable ϑ_f takes values from the same alphabet Λ.

With respect to the resolution of variance tying and variance flooring, results will be presented in this chapter for two series of experiments. In the first, variances are not tied ($\vartheta_v = \text{X/X}$) while variance floors are tied at the model or state level. In the second series, variance floors are tied at the same level as variances themselves ($\vartheta_f = \vartheta_v$).

9.3.4 Feature extraction

To check if results for the MFCC feature vectors used in the baseline system are valid also with another type of feature vector, MFCCs are replaced with LPCCs in one experiment. Parameters from a 16-pole linear prediction filter are computed with the autocorrelation method and are transformed to 12-element cepstrum. The energy term is the raw log-energy within each frame of samples, normalized within each utterance to have constant maximum amplitude for every utterance.

9.4 Results

Results in terms of equal-error-rate (EER) based on same-sex impostor attempts and a speaker-independent *a posteriori* threshold for test cases presented below will be presented in two ways. First, EER is given as a function of scale factor γ or α (Figures 9.1–9.3). In all figures results for the variance scaling case with $\alpha = 1$ are included as a baseline for comparison. Second, results for particular choices of scale factor are given in Table 9.2. In the Table, we then treat the Gandalf development set as our development corpus and determine an optimal *a priori* scaling factor $\hat{\gamma}$ (or $\hat{\alpha}$) for each system setup from this data. We treat the other three data sets as our evaluation corpora and compare the average error rates achieved with *a priori* scale factors on the three evaluation sets to the corresponding results with

Table 9.2: Average EER for a) variance flooring and b) variance scaling. Baseline is target-independent scaled variances with $\alpha = 1$. For the fourth column, the scale factor was chosen *a posteriori* for each individual data set. For the last column, a single scale factor was chosen based on the Gandalf development set and used as an *a priori* factor with the other data sets. All averages are taken over the three other data sets (Gandalf evaluation, SESP and Polycost).

a) flooring

tying level	flooring level	baseline	*a posteriori* γ	*a priori* $\hat{\gamma}$
X/X	M/X	4.22%	5.12%	5.14% ($\hat{\gamma} = 0.6$)
X/X	S/X	4.22%	4.13%	4.52% ($\hat{\gamma} = 0.8$)
X/X	X/X	4.22%	3.94%	3.95% ($\hat{\gamma} = 1.1$)
S/X	S/X	8.11%	3.97%	4.53% ($\hat{\gamma} = 0.9$)
S/S	S/S	4.01%	3.95%	4.00% ($\hat{\gamma} = 1.0$)

b) scaling

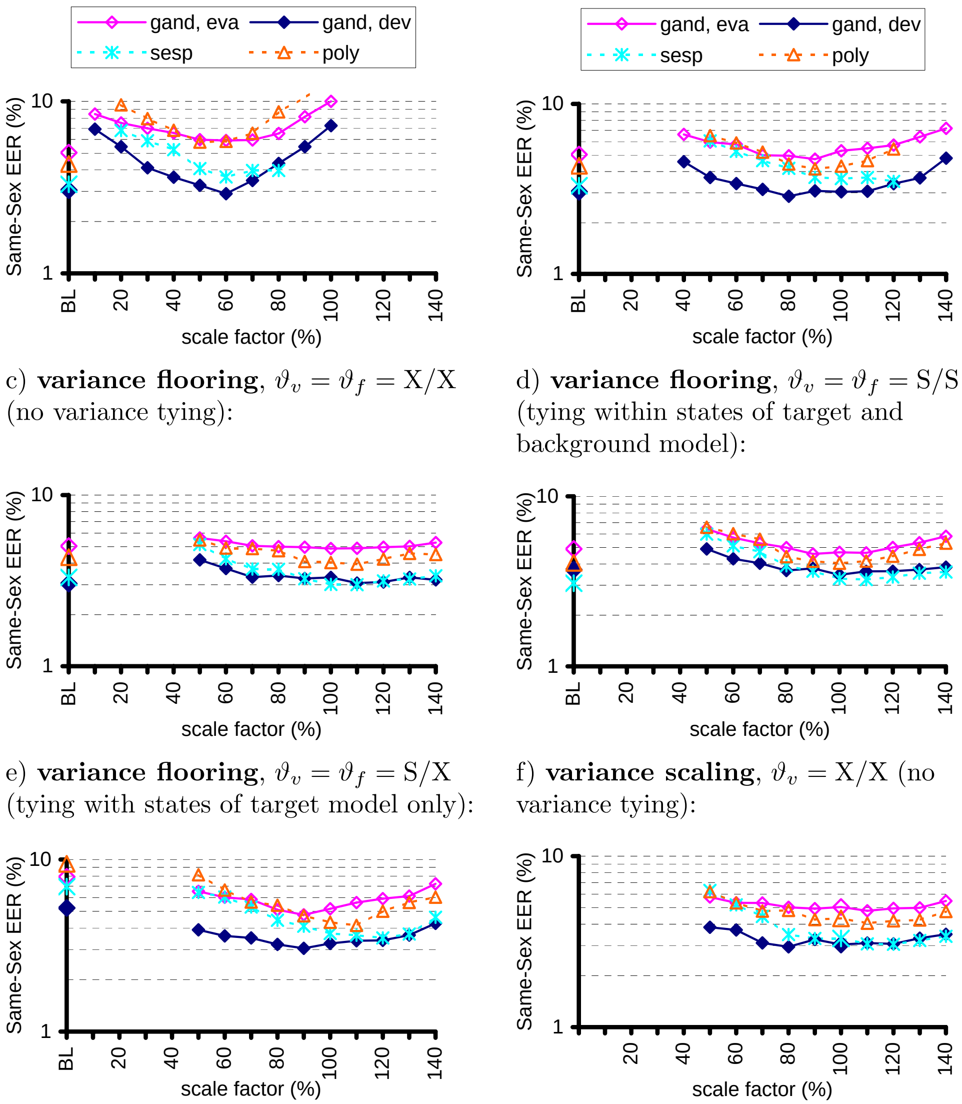
tying level	flooring level	baseline	*a posteriori* α	*a priori* $\hat{\alpha}$
X/X	-	4.22%	3.96%	4.44% ($\hat{\alpha} = 0.8$)

a posteriori optimal scale factors for each individual evaluation corpus, and with baseline.

Figures 9.1a–c show the error rate as a function of γ with no variance tying and for three cases of variance flooring in target models: model, state and mixture component-dependent variance floors (ϑ_f: M/X, S/X and X/X). They show that the higher resolution in flooring, the less critical is the choice of scaling factor, since the minima in those curves are much wider and the position of the minima are closer to each other than for low resolution flooring. Considering results in Table 9.2, there is a clear trend that higher resolution in variance flooring is better than lower, and only for the mixture component-dependent floors is the average error-rate with an *a priori* scale factor lower than with target-independent variances.

Figures 9.1c–e show the error rate as a function of γ for variance flooring and three cases of variance tying, ϑ_v: X/X, S/S and S/X (X/X represents no tying at all). In these cases, variance floor vectors are tied at the same level as variance vectors themselves ($\vartheta_v = \vartheta_f$), i.e. the variance floor resolution is as high as possible given the level of variance tying.

Since the variance flooring method is applied to avoid over-training of variances on sparse training data, it can be expected that for a given recognition task and corpus, the need for flooring would systematically decrease with increased size of the enrollment set. The more training data the less should variances need to be floored. Hence, we expected the optimal scale factor in variance flooring to decrease with larger enrollment sets. Figure 9.2 shows a comparison of EER as a function of scaling factor with enrollment sizes from 0.3 to 1 minutes (3 to 12 training examples per digit). Graphs are included for mixture component and state dependent



a) **variance flooring**, $\vartheta_v = \mathrm{X/X}$, $\vartheta_f = \mathrm{M/X}$ (no variance tying, model-dependent variance floor):

b) **variance flooring**, $\vartheta_v = \mathrm{X/X}$, $\vartheta_f = \mathrm{S/X}$ (no variance tying, state-dependent variance floor):

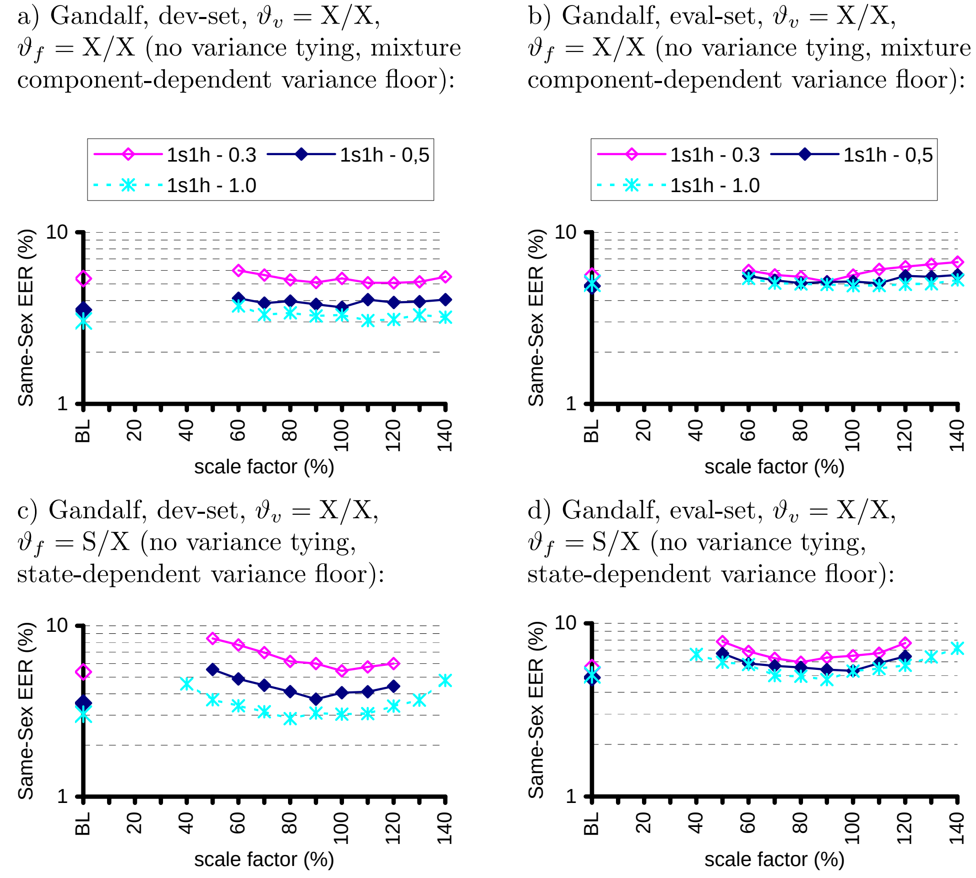
c) **variance flooring**, $\vartheta_v = \vartheta_f = \mathrm{X/X}$ (no variance tying):

d) **variance flooring**, $\vartheta_v = \vartheta_f = \mathrm{S/S}$ (tying within states of target and background model):

e) **variance flooring**, $\vartheta_v = \vartheta_f = \mathrm{S/X}$ (tying with states of target model only):

f) **variance scaling**, $\vartheta_v = \mathrm{X/X}$ (no variance tying):

Figure 9.1: Same-sex EER as a function of the variance flooring factor γ (a–e) or the variance scaling factor α (f) for the four data sets and for three levels of variance tying (ϑ_v): variance flooring a–c) X/X, d) S/S, e) S/X, and f) variance scaling X/X. Charts a–c differ in how the variance floor vector is tied in the target model: a) within model b) within state c) no tying of variance vector. In charts a–e the baseline case with target-independent scaled variances, $\alpha = 1$ and the corresponding variance tying configuration is included at the left ('BL'). DET curves and score distribution plots for (b,c,f) are included in Appendix E.



a) Gandalf, dev-set, $\vartheta_v = \mathrm{X/X}$, $\vartheta_f = \mathrm{X/X}$ (no variance tying, mixture component-dependent variance floor):

b) Gandalf, eval-set, $\vartheta_v = \mathrm{X/X}$, $\vartheta_f = \mathrm{X/X}$ (no variance tying, mixture component-dependent variance floor):

c) Gandalf, dev-set, $\vartheta_v = \mathrm{X/X}$, $\vartheta_f = \mathrm{S/X}$ (no variance tying, state-dependent variance floor):

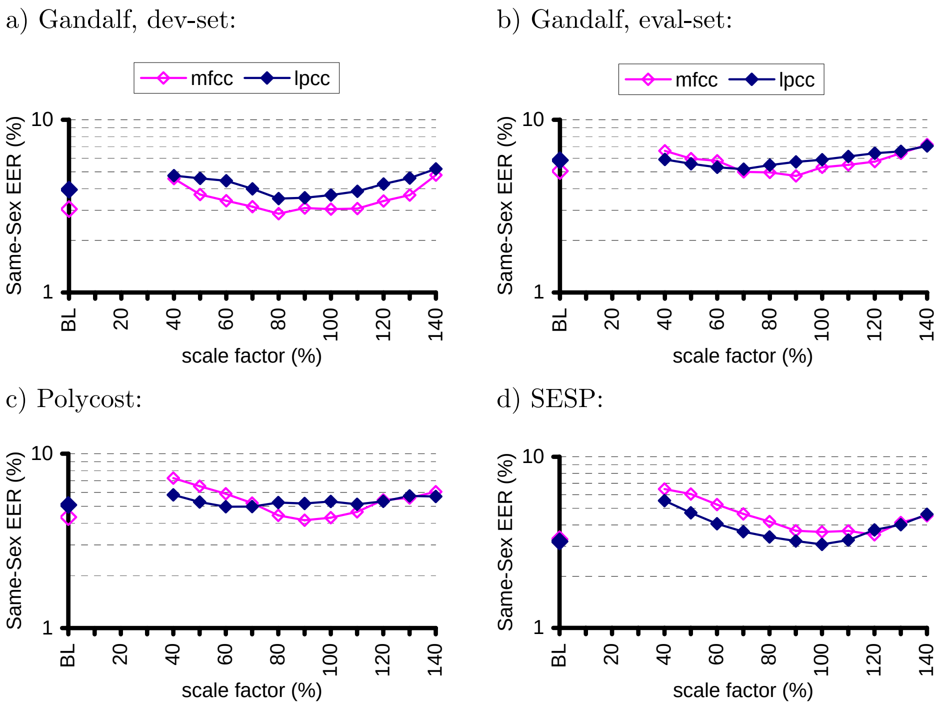
d) Gandalf, eval-set, $\vartheta_v = \mathrm{X/X}$, $\vartheta_f = \mathrm{S/X}$ (no variance tying, state-dependent variance floor):

Figure 9.2: Comparison between different enrollment set sizes on the Gandalf development and evaluation sets. Variance floors are either not tied (a,b) or tied at the state level in target models. 'BL'-points show results for variance scaling with $\alpha = 1.0$. Variances parameters themselves are not tied.

variance floors on the development and evaluation parts of Gandalf, respectively. The expected trend is found in Figure 9.2c but not in the other three graphs.

One could further expect that the improvement of variance flooring relative to target-independent variances would be higher with larger enrollment sets than with smaller ones. There is no evidence for this in the figure.

We also compared the MFCC-based features used so far, with LPCC-based features. Figure 9.3 shows error-rates for state-dependent floors on the four data sets. The locations of minima are roughly the same for the two feature types on the respective databases suggesting that a scaling factor optimized for one feature



a) Gandalf, dev-set:

b) Gandalf, eval-set:

c) Polycost:

d) SESP:

Figure 9.3: Comparison between EER with MFCC-based and LPCC-based features on the four corpora. Each curve contains results with fixed, target-independent variances (BL) and with variance flooring with state-dependent floors.

type is reusable for another. With LPCC features, like with MFCC features, the smallest achieved EERs are comparable to those produced by the baseline system, with target-independent variances.

The purpose of tying variances across a set of mixture components is to reduce the number of variance parameters so the remaining parameters can be robustly estimated from enrollment data. We therefore expected the relative improvement with variance flooring over baseline to be smaller with tied variances than with non-tied variances. This is also the case in Table 9.2 for ϑ_v = S/S (4.01% → 3.95%) relative to X/X (4.22% → 3.94%). However, in terms of absolute EERs, differences between S/S and X/X are small.

The motivation for ϑ_v = S/X is to allow for a high modeling accuracy in the background model for which there is usually much data available, while having a more coarse but robust model for the target speaker for which there are usually little available data. The poor performance of scaled variances in Figure 9.1e indicates that the computation of state variances from the background model variances in Eq. (9.3) is not good for predicting the state variances of the target model. An alternative approach would be to use a multi-speaker model with tied variances in parallel to the one with non-tied variances, and to take the seed variances ($\mathbf{s}^2_{wj}$) from the former while using the latter for score normalization.

There is a stronger correlation between error curves in Figures 9.1c–e for the two Gandalf sets than between those and the corresponding SESP and Polycost curves. This is reasonable since the choice of a good scaling factor for the various methods may depend on relationships between training data for background models and data for client enrollment and test; and since those are different for the corpora we used. It is therefore likely that better predictions could have been made from development sets especially designed for the SESP and Polycost sets respectively. Table 9.2 includes results for an *a posteriori*, corpus-dependent choice of scaling factor that give a hint on what results could be achieved with such development sets. The table shows that results with *a priori* and *a posteriori* choices of γ are very similar in the X/X and S/S cases.

Figure 9.1f shows error rate as a function of scaling factor α for variance scaling and $\vartheta_v = \text{X/X}$. Curves have a similarly flat shape as the corresponding curves in Figure 9.1c. In both figures, the average EER over all curves have a minimum at a scale factor around 1.1. An optimal value greater than 1.0 is unexpected since it would mean that all target model variances are larger than the corresponding variances in the background model. However, because of the flat shape of the curves, our estimate of the best scale factor clearly has a large variance.

9.5 Discussion

In search of some insight into what effect variance flooring and scaling have on the ASV system, we include in Appendix E graphical presentations of score distributions produced with a range of scale factors. Score distributions are included for all four data sets used in this chapter, with three variance estimation methods and a scale factor range 0.40–1.40 at 0.20 intervals. The methods are state-dependent variance floors ($\vartheta_f = \text{S/X}$), mixture component-dependent variance floors ($\vartheta_f = \text{X/X}$) and variance scaling. In all cases, variances are not tied ($\vartheta_v = \text{X/X}$).

Score distributions are illustrated by three types of graphical methods: DET plots, normal quantile plots and score histograms. Both the actual score distributions and normal distributions defined by estimates of the mean and variance of the respective score distribution are shown. The normal distribution serve both as a reference for the shape of score distributions, and as a (hypothesized) robust estimate of the underlying "true" score distribution (cf. Section 7.2, p. 126).

A trend observed from normal quantile plots and score histograms is that the left tail of true-speaker distributions is shortened with a raised variance floor or scaled variance. This is manifested in the normal quantile plots as the lower left end of the plot bending upwards, and in the score histograms as the tail being drawn closer towards the center of the distribution. Since the integral of the left tail in a true-speaker distribution up to a threshold value correspond to the false reject error rate of a system, the observed shortening of the tail should correspond to a relative decrease in false reject rate. This trend is accompanied by a lengthening of the right tail of the impostor score distribution, corresponding to a relative increase in false accept rate.

The trends referred to in the previous paragraph can be generalized to say that a class-dependent score distribution is changed with increased variance floor, or scaled variance, such that the left tail is shortened and the right tail is lengthened – distributions are skewed to the left. "Class" here refers to true-speaker or impostor test sets.

To understand the performance of the ASV system both the true-speaker and impostor score distributions must be taken into account, so we turn our attention to the DET plots in Appendix E.

As a basis for discussing the effect of variance flooring and scaling on DET curves, we will establish some basic relationships between the properties of score distributions and the resulting DET curves.

First, recall that the coordinate system used in DET plots is designed to draw pairs of Gaussian score distributions as straight lines (Martin et al., 1997).

Second, as shown by Auckenthaler et al. (2000), an *increased mean difference* between impostor and true-speaker distribution results in a *shift* of the DET curve *towards the origin*, while an *increase in true-speaker score variance* results in a *counter-clockwise rotation* of the DET curve.

Third, consider non-normal score distributions. As an example of DET curves for non-normal true-speaker score distributions, Figure 9.4 shows DET curves for four types of true-speaker score distributions (on the right), together with plots of the pdfs themselves (on the left). All four true-speaker score distributions have zero mean and a variance equal to 60. A single normal impostor distribution with identical variance and a shifted mean (-25) is shared by all four DET curves. The solid blue DET curve (b) shows a normal true-speaker score distribution as a straight line, while DET curves (a) and (c) show asymmetric distributions. (a) is a χ^2-distribution flipped horizontally about its own mean. It is *skewed to the right* with a long left tail and a short right tail. This distribution results in a *concave-shaped* (◟) DET curve. (c) is the (non-flipped) χ^2-distribution that is *skewed to the left* with a short left tail and a long right tail and appears as a *convex-shaped* (◝) DET curve. The fourth distribution (d) is a Student's t-distribution with five degrees of freedom. It is symmetric like the normal but has *longer tails and a more narrow "hill"*. It appears in the DET plot as a *concave-shaped* curve.

Now we are ready to discuss the effect of variance flooring and scaling on DET curves.

DET curves drawn from our normal approximations of score distributions in the Appendix are mostly parallel within each plot, indicating that the score distribution variance does not change much with the scale factor value in variance flooring or scaling. Their distance relative to the origin do change, however, indicating that the separation between impostor and true-speaker score mean changes. The latter

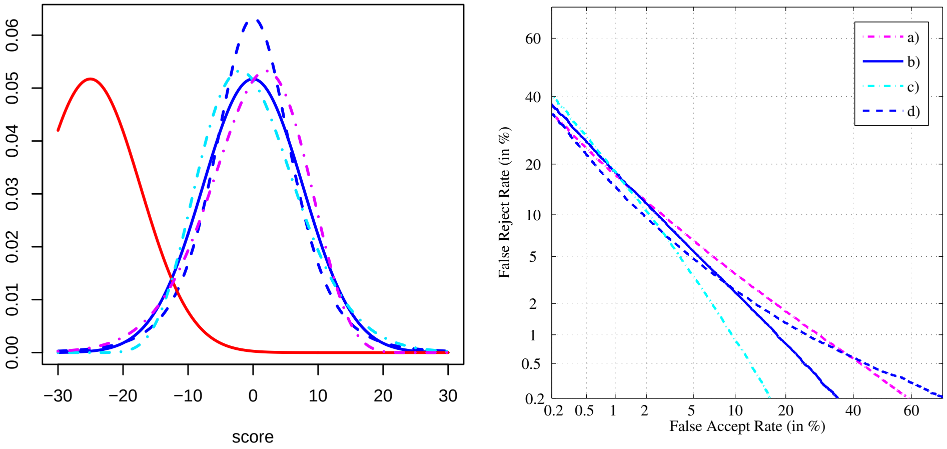


Figure 9.4: Four examples of simulated true-speaker score distributions and the corresponding DET curves. a) χ^2-distribution flipped horizontally about its own mean, b) normal distribution, c) χ^2-distribution, d) scaled Student's t-distribution with five degrees of freedom. All true-speaker distributions have the same mean and the same variance.

change is more pronounced with state-dependent variance floors than with mixture component-dependent floors and with variance scaling.

Looking at deviations from the Gaussian shape of score distribution through DET curve bending, no clear pattern emerges that is common to all data sets. On the Gandalf evaluation data and Polycost data, DET curves are concave (pp. 297 and 313). This together with score histograms and quantile plots indicate long left tails on the true-speaker score distributions. For the Gandalf evaluation data, this result is mainly due to a single target speaker, from which most of the score points in these tails originate from. The ASV system obviously has performed badly for this target, and none of the variance flooring or scaling techniques were able to remedy this.

On Gandalf development data (p. 289) and SESP data (p. 305), there is a tendency for an increased scale factor to rotate the DET curve clock-wise, resulting in larger differences between DET curves for different scale factors at the high-left and low-right regions of the DET plots than in the middle region around the EER. In most cases DET curves are straight, except for a few examples of curve bends that can probably be explained by random variations due to the limited number of score data points. The rotation is found in the direct DET plots, but usually not in DET plots based on a normal approximation of score distributions, so the rotation seems to be due to some other effect than changed variances. With the observed left-skewness of class-dependent score distributions observed on quantile plots and

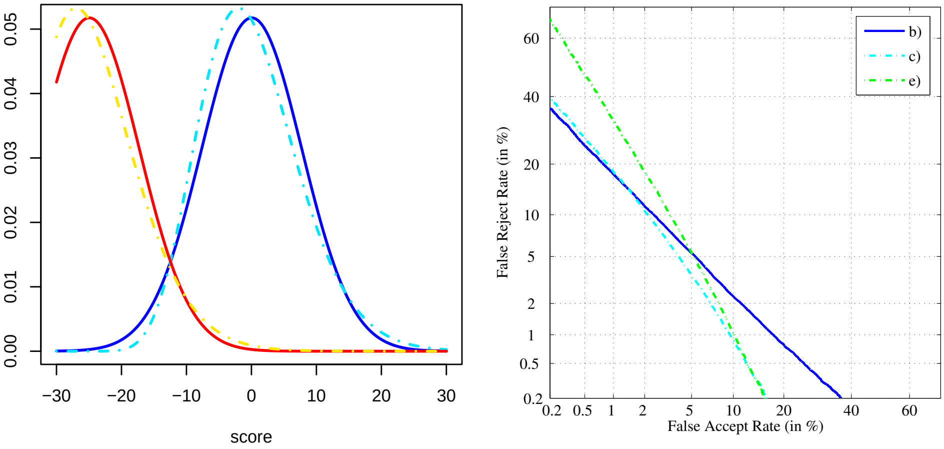


Figure 9.5: Examples of simulated true-speaker and impostor score distributions and corresponding DET curves. b) normal true-speaker and impostor distributions, c) χ^2 true-speaker distribution and normal impostor distribution, e) χ^2 true-speaker and impostor distributions. All distributions have the same variance and each class (true-speaker and impostor) of distribution have the same mean.

score histograms in mind, we therefore generated synthetic score data from skew true-speaker *and* impostor score distributions. The result is shown in Figure 9.5 as DET curve (e). In the same plot are included the curves for normal true-speaker and impostor distributions (b) and for normal impostor distribution with a skew true-speaker distribution (c) from Figure 9.4. We see that the result of skewing both distributions to the left, as in the example, is a clock-wise rotation of the DET curve together with a slight convex bending of the curve. Hence, the distribution skewness at increased scale factor values observed on the pdfs themselves is consistent with a clock-wise rotation of the DET curve seen in our DET plots.

While the pattern of rotation of DET curves was not seen for Gandalf evaluation data, as noted above, it can be noted that the change in DET curves in the upper left region of DET plots could be the result of an “underlying” rotation, but that the (lower) right part of the curves have been “pinned” by a bad performance of the ASV system for a few target speakers. Hence, observations on Gandalf evaluation data does not necessarily contradict the discussion in the previous paragraph. Results on Polycost data still fall outside of the DET curve rotation pattern.

To summarize observations on DET curves in Appendix E, the primary effect of variance flooring and scaling in terms of changes to score distributions is a translation of DET curves relative to the origin, such that there is often a minimum error rate for each operating point criterion for some scale factor. A secondary effect is a clock-wise rotation of curves with increased scale factors. We hypothesize the latter effect to be due to an increased skewness of true-speaker and impostor distributions to the left with shortened left pdf tails and lengthened right tails.

These results support the conclusions drawn from EER figures in this chapter and suggests that the effect of variance flooring and scaling is larger at operating points with high or low ratios between false reject and false accept rates than around the EER operating point.

9.6 Conclusions

In this chapter we compared three modifications to the EM algorithm for HMM training on sparse data in the context of text-dependent speaker verification.

The first approach used target-independent variances. Variances were copied from a gender-dependent, multi-speaker background model and were kept fixed while the EM-algorithm was applied to means and mixture weights. This approach was also generalized to variance scaling, where the target model variance was set equal to a constant scale factor times the corresponding background model variance. The scaling aspect brought no advantage in terms of optimizing EER — the error rate did not change much with small changes in the scale factor around 1.0.

In the second approach, variances were trained but they were floored after each iteration of EM. Three variants of the variance flooring method with different resolution were tried and it was found that the one with the highest resolution performed best. In this approach, the floor for the variance vector of a given Gaussian mixture component is proportional to the corresponding variance vector in the background model. The optimal scaling factor for this kind of variance flooring was found to be around 1, which means that all variances were about the same value as the corresponding target-independent variances, or larger. The average EER over three evaluation data sets with an *a priori* scale factor determined on a development data set was smaller than with target-independent variances. However, the difference was very small, and the approach with target-independent variances is much simpler.

The third approach was to reduce the number of variance parameters through variance tying. One important advantage brought by tied variances is reduced storage requirements. With variances tied across eight mixture components within each state, 30% of the size is saved. Another expected advantage is that fewer parameters can be more robustly estimated, but no positive effect was observed from this, and recognition accuracy with and without tying was comparable.

These results consolidate similar observations made in (Newman et al., 1996) and at NIST evaluations in text-independent ASV (NIST, 1998; Reynolds et al., 2000) that target models trained as adaptation of multi-speaker models with keeping covariance matrices constant brings a significant advantage, especially in the case of very scarce enrollment data.

One important approach not tested in this chapter is to train variances with a MAP method (Lee and Gauvain, 1993, 1996). MAP adaptation provides data-dependent weighting between *a priori* information (e.g. the background model variance) and observed data (sample variance in enrollment data), such that enrollment data get high weighting where available, while *a priori* information is used otherwise. This should allow for training variances robustly without variance flooring. Experiments with MAP adaptation of variances in the context of GMMs for ASV have shown some success compared to using target-independent variances. Early GMM experiments on Gandalf data showed an advantage for adapting variances (Neiberg, 2001), while later experiments (with an improved system) targeted at optimizing the system for use in the PER application showed little difference compared to using target-independent variances. (Reynolds et al., 2000) found it better to update only mean parameters and leave variances unadapted on Switchboard data.

Chapter 10

PER experiments

10.1 Introduction

This chapter reports on findings from an evaluation of the on-site and telephone versions of the PER system described in Chapter 5 (p. 79). The evaluation was conducted with speech data collected through actual use of the two system versions. The data collection and data themselves were described in Section 6.3 (p. 101).

All our development of the speaker verification component of the PER system before the collection of evaluation data was made using general purpose telephone corpora Gandalf (Melin, 1996), Polycost (Hennebert et al., 2000), SpeechDat (Elenius, 2000) and Switchboard, since the Department's research was directed on telephone applications of speaker verification (Lindberg and Melin, 1997; Melin, 1998; Melin et al., 1998; Nordström et al., 1998; Melin and Lindberg, 1999b; Bimbot et al., 1999, 2000; Neiberg, 2001). Hence, the system used to collect live evaluation data was not optimized for the particular application it was used in. However, in parallel to collecting evaluation data, separate, application-specific development data were collected allowing for off-line simulation experiments with an optimized system. In this chapter, results are presented both for the initial, general-purpose system and the optimized, application-specific system.

Besides the variants of our own research system, a commercial speaker verification system has also been tested on the collected corpus. Results from these tests serve as calibration of the data and the recognition tasks.

Results from practical use of ASV technology for person authentication in on-site applications have been reported in several publications. Test sites include Texas Instruments corporate headquarters in Dallas (Doddington, 1985), Siemens in Munich (Feix and DeGeorge, 1985) , LIMSI in Paris (Mariani, 1992), AT&T Bell Labs in cooperation with a large bank (Setlur and Jacobs, 1995), Fraunhofer Institute in Erlangen (Wagner and Dieckmann, 1995), University of Frankfurt (Schalk et al., 2001) and Panasonic Speech Technology Laboratory in Santa Barbara (Morin and Junqua, 2003). At AT&T Bell Labs the application was an automated teller machine (ATM), while at all other sites it was a voice-actuated lock that secured access to a physical room or building.

At Texas Instruments, a template based system was installed in the mid 1970s (Rosenberg, 1976; Doddington, 1985). It was aurally text prompted using strings of four words like “Proud Ben served hard”, and used a sequential decision strategy where claimants were asked to speak new word sequences until a certain level of confidence was achieved. False reject and false accept rates (casual impostors) of below 1% are reported with on average 1.6 utterances required by the sequential decision strategy. Users were required to step into a booth to use the system.

At LIMSI, a text-dependent, template based system was first publicly demonstrated in 1985. It was installed in a voice-actuated door lock application at the lab in 1987 and was used by about 100 users (Mariani, 1992). A second generation system was installed in 1990 and a new generation, HMM-based system was developed in 1997 which has so far only been used for data collection (Lamel, 2005).

At Panasonic Speech Technology Laboratory in Santa Barbara a biometric terminal has been in service since April 2002 by the building’s main entrance door (Morin and Junqua, 2003). It is a multi-modal access control system where any of the three modes speech, fingerprint or keypad (10-digit account number) can be used individually, or in combination for uncertainty recovery. The speech sub-system is template based and operates on user-selected pass-phrases in an open-microphone mode. Users can speak the pass-phrase at any time from within typically 0.3–3 meters from the terminal. The system has been in use by about 35 enrolled users and was reported to have about 8% FRR and 0.1% FAR (2.8% EER) for the speech mode only. Some of the initial rejections were recovered via another mode reducing the FRR to about 5%. Other results using data collected by this system have been reported in (Bonastre et al., 2003a).

AT&T conducted a six month field trial with an ATM application where a text-prompted, HMM-based speaker verification system was used in addition to regular PIN codes typed on a keyboard (Setlur and Jacobs, 1995). Claimants were asked to repeat random 4-digit phrases into a handset connected to the ATM.

10.2 Development tests

This section describes what data was used for developing the PER system and how it was used.

Table 10.1 shows results from the development experiment to determine empirical values for weights ω_ξ used in combining scores from the HMM and GMM subsystems (Eq. 3.23, p. 47).

The value of the decision threshold θ (Eq. 3.25) was also determined empirically as the same-sex EER threshold with the combined ASV system on the same development test set.

Table 10.1: Equal error rate ϵ_ξ, standard deviation σ_ξ of score distribution and combination weights ω_ξ for the HMM and GMM subsystems as determined from a development experiment on Gandalf data.

subsystem (ξ)	EER (ϵ_ξ)	stdev (σ_ξ)	weight (ω_ξ)
HMM (H)	7.51%	4.017	0.142
GMM (G)	6.11%	0.6747	0.858

10.2.1 Development data

Most experiments behind development decisions in the design of the HMM subsystem were done on various partitions of the Gandalf (Melin, 1996) and Polycost (Hennebert et al., 2000) corpora, e.g. (Melin and Lindberg, 1999b) and (Nordström et al., 1998). With particular development for the PER application in mind, a PER-like development test configuration on Gandalf was created. It was used to optimize the configuration of the GMM subsystem and to determine the *a priori* score fusion weights and the decision threshold used during data collection.

The PER-like development test configuration uses one of two fixed sentences in place of names. Half of the target speakers were assigned one sentence and the other half the other sentence. Enrollment was performed with 10 repetitions of the sentence and 10 five-digit sequences taken from two recording sessions from different handsets (enrollment set d5+fs0x, cf. Table 6.8), while each test was performed with a single repetition of the same sentence and an aurally prompted string of four digits (test set 1fs+1r4-fs0x, cf. Table 6.9). All impostor tests used in development experiments were same-sex attempts. True-speaker test sessions were recorded from up to 10 different handsets per target, but at least half of the sessions came from one of the target's enrollment handsets. Impostor test sessions were generally *not* recorded from one of the target's enrollment handsets. Even though this development test configuration was designed to simulate the PER application as well as possible given the constraints of the already existing Gandalf corpus, it differs in several aspects from real PER data as summarized in Table 10.2 for the telephone version of PER. The on-site version of PER naturally adds the differences already identified between the two PER versions (Table 5.1).

Background models were trained on subsets of files from 960 speakers in the Swedish landline FDB5000 SpeechDat corpus (Elenius, 2000). Background models in the HMM subsystem were trained on a *digits* subset composed by five files per speaker that may contain pronunciations of isolated or connected digits (corpus and item identifiers with parentheses): a random 10-digit sequence (`B1`), a 10 or 7 digit prompt sheet number (`C1`), an 8-12 digit phone number (`C2`), a 16-digit credit card number (`C3`), and a 6-digit PIN-code (`C4`). Background models in the GMM subsystem were trained on a *mixed* subset composed by six files per speaker: a random 10-digit sequence (B1), three phonetically rich sentences (`S1`-`S3`), and two phonetically rich words (`W1`, `W2`). None of the 960 speakers occur in the Gandalf or PER corpora.

Table 10.2: Main differences between the PER-like development set on Gandalf and telephone subset of PER evaluation data.

Aspect	Gandalf development	PER evaluation
Elicitation	recording	use of ASV system
Enrollment data	two session, two different handsets	single session
Test data (per target)	multiple handsets; cross-handset impostors	single handset; same-handset impostors
Impostors	random pseudo-impostors	dedicated impostors
Vocabulary	sentence+digits	proper name+digits
Passphrase variation	1 sentence/20 targets	1 name/1 target

Acoustic models for speech recognition were trained on 4016 speakers (gender-balanced) in the referred SpeechDat corpus, including all files from each speaker with the exception of files transcribed with truncated signal, mispronunciations, unintelligible speech or phonetic letter pronunciations (Lindberg et al., 2000). The number of used speakers is less than 5000 because 500 speakers were withheld for testing, 37 more because they were included in the Gandalf corpus, and 10% of the remaining speakers were set aside for development testing. Hence, there is no speaker overlap between this data and the Gandalf data. There is also no speaker overlap between used SpeechDat data and PER data. The total duration of speech segments in this training data is approximately 120 hours.

Six of the subjects (M1003, M1005, M1015, 1032, F1025 and F1031) in the PER test group participating as clients (five in group E and one (F1031) in group L) and impostors are also included in the development set of the Gandalf corpus, together with three subjects (M1002, M1166 and F1009) participating as impostors only in the PER collection (one with gate-only data, the other two with gate and telephone data). The unfortunate overlap between subjects in PER evaluation data with respect to Gandalf development data is thus 11% of the 54 clients and 9% of the 98 impostors in the PER gate-only test set and 19% of the 27 clients and 16% of the 51 impostors in the condition-parallel test sets. More details about subjects participating both in Gandalf and PER can be found in Section 6.4.

Table 10.3: Statistics on the average number of attempts per enrollment item and gross duration (*minutes:seconds*) of enrollment sessions, based on complete enrollment sessions included in enrollment sets E2a_*c*.

Condition, *c*	#Sessions	Attempts			Duration		
		Min	Avg	Max	Min	Avg	Max
gate/hall	54	1.1	1.9	4.5	1:42	3:41	8:44
landline/office	54	1.0	1.1	2.1	1:35	2:18	4:32
mobile/office	29	1.0	1.2	1.7	1:49	2:26	3:48
mobile/hall	29	1.0	1.4	3.9	2:02	3:08	8:28

10.3 Field test results

10.3.1 Enrollment

During the data collection period, 56 subjects started enrollment. 54 of them succeeded to complete the enrollment sessions they were asked to do (enrollment in two conditions for client group L and four conditions for client group E). Table 10.3 shows statistics on how many attempts per item they made and the total duration of the sessions. Durations are measured from session start to completed enrollment, including time for system prompts, system delays, etc. For all telephone enrollment sessions this includes the entry of a 7-digit enrollment code by voice for authorization, and for sessions from a mobile phone it also includes a sub-dialog to determine if the call was made from the office or the hall. Attempts statistics are based on the average number of attempts per enrollment item and session, e.g. 1.1 in the Min-column for the gate/hall condition means the session with the least number of attempts had 11 attempts total since there were ten items. Attempts are counted from the system point of view, disregarding whether users actually made an attempt to speak an enrollment item or not.

In the longer enrollment sessions, users typically experienced problems with a few of the enrollment items, which they had to repeat many times before the speech recognizer was able to recognize their utterance correctly, or they opted to skip the item. The skip-possibility was introduced as described in Section 5.5 (p. 84) as an attempt to limit user frustration in these cases and to allow the enrollment process to be completed despite such problems. Within the enrollment sessions that were eventually completed, eight subjects (15%) skipped one item and one subject (2%) skipped two items in the gate/hall condition, while a single subject (2%) skipped one item in the landline/office condition. In the two mobile conditions, no items were skipped.

39 of the 54 subjects who completed their requested enrollment sessions (72%) completed all their enrollment session at the first attempt, while the remaining 15 (28%) had one or more failed or aborted enrollment sessions before the complete ones. In failed sessions for nine of the latter, the actual enrollment procedure was never started because either subjects had not enabled enrollment through the

intranet or the enrollment window had expired (four cases); their name was incorrectly recognized (seven cases); or they did not have the enrollment code available (two cases). Six subjects terminated the enrollment procedure of one or more enrollment sessions pre-maturely. Four of the six terminated one session each (three in the gate/hall and one in the mobile/hall condition), probably after feeling disturbed by other people passing through the gate or otherwise making noise in the hall. One of the six, it appeared, had removed his last name through the web interface so the ASR grammar contained only his first name while he was still speaking his full name. After correcting this, his enrollment sessions were immediately successful. The last of the six had severe problems with getting the speech recognizer to recognize his utterances. He terminated three enrollment sessions in the gate/hall condition and one in the landline/office condition before succeeding with enrollment. The source to the system's problems with this subject appears to have been a combination of the subject being a non-native speaker of Swedish and him speaking very loudly to the system.

The remaining two of the 56 subjects (3.6%), one male and one female subject, failed to complete any enrollment session. The female subject, a non-native speaker of Swedish, tried to enroll in both the gate/hall condition and the landline/office condition, with similar results in both cases: The speech recognizer consistently failed on digit sequences including the digit 7, probably caused by her non-native pronunciation of this digit (a typical Swedish pronunciation would be [ɧʉː] or [ʃʉː]). The male subject terminated his first (and only) enrollment session in the gate/hall condition after being disturbed by noise from other people passing through the gate at the time. He suggested to try another time, but never did so.

10.4 Simulation results

Results in this section are from off-line simulations of speech recognition and speaker verification operations using the PER corpus (with recordings from actual use of the PER system; cf. Section 6.3). Results are presented in terms of DET curves and EER.

10.4.1 Baseline system

The original speech recognition and speaker verification components of the PER system (as described in Chapter 5) used to collect data, without the use of a speech detector, is designated as the baseline system.

Results for the baseline system in the gate/hall and landline/office conditions using E2a_*c* and S2b_*c* enrollment and test sets for the respective condition, are shown by the dashed DET curves in Figure 10.1. EERs are 6.4% for gate/hall and 4.0% for landline/office. Error rates are lower in the telephone case as was expected since both acoustic models (ASR) and background models (ASV) were developed on telephone data. However, test sets S2b_G8 and S2b_LO are not

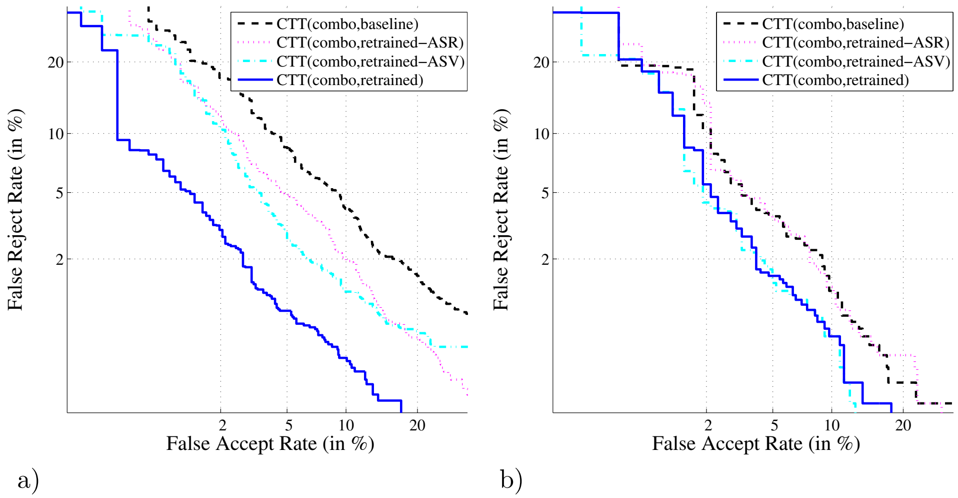


Figure 10.1: DET curves for baseline and retrained systems in the a) gate/hall (S2b_G8) and b) landline/office (S2b_LO) conditions. Baseline is with the original speech recognition and speaker verification components used during data collection, while in the retrained case both components have been adapted to condition-dependent data from background speakers. The remaining two plots show results where only one of the speech recognition or speaker verification components has been adapted.

directly comparable since they are based on different number of subjects, etc. A more fair comparison is shown in Figure 10.2 using the condition-parallel test sets S2b_Q:*c*. The comparison is more fair because, firstly, every test in a given condition has a corresponding test in all other conditions (Section 6.3.4.4, p. 113), and secondly, a name and four digits is used per test in all four conditions, with a digit in a random position having been omitted from every test in the gate/hall data. Figure 10.2 confirms the lower error rates for landline/office than for gate/hall, however with a smaller difference than in Figure 10.1, even though one digit less per utterance is used in the gate/hall condition.

Figure 10.2 also indicates the operating points corresponding to the *a priori* decision threshold determined using the EER point on the Gandalf development test configuration. The resulting operating points are near the *a posteriori* EER point in the telephone conditions, while it is clearly far-off in the gate/hall condition.

10.4.2 Retrained system

Models of the original (baseline) PER system were adapted to PER-specific data from background speakers to create new, *retrained*, condition-dependent systems. These systems are expected to perform better in the PER application than the

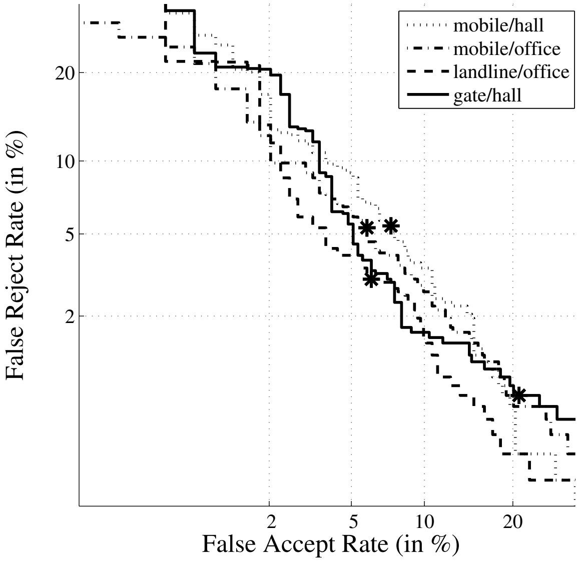


Figure 10.2: A comparison between conditions using the condition-parallel test sets (S2b_Q:c) and the baseline system. A name plus four digits is used in all conditions. EERs are 6.4% (MH), 5.8% (MO), 4.3% (LO) and 5.1% (G8). Asterisks (*) mark the operating points determined by the *a priori* threshold.

baseline system, but since background data was collected in parallel to evaluation data, the retrained systems have only been tested using off-line simulations on recorded data.

Acoustic models in the speech recognition component were not only trained on the new data, but their structure were also changed in two respects: models were made gender-dependent and the number of terms in the Gaussian mixture was reduced from eight to four. The new models were created with the following procedure. Gender-independent models with four terms per state were created with the same procedure as the original (eight-term) models. The four-term models were then cloned into male and female gender-dependent models and background speaker files were tagged as male or female. Mean vectors of the gender-dependent models were then adapted to the new data by a gender-independent Maximum Likelihood Linear Regression (MLLR) transform followed by a single MAP iteration using HTK (Young et al., 1999). The MLLR transform was made with a single transformation matrix for both male and female models, while the MAP adaptation was made with gender-dependent data.

Background models of the speaker verification component (both the HMM and GMM subsystems) were adapted to new data with three iterations of the EM-algorithm and the ML criterion, updating means, variances and mixture weights.

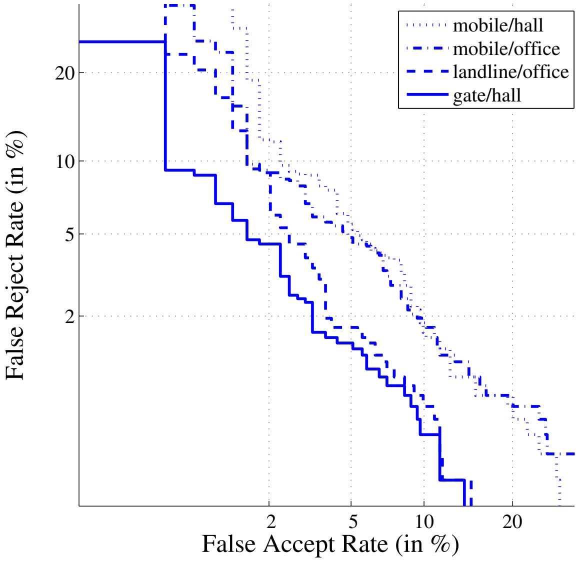


Figure 10.3: A comparison between conditions using the condition-parallel test sets (S2b_Q:c) and the retrained, condition-dependent systems. A name plus four digits is used in all conditions. EERs are 5.3% (MH), 4.8% (MO), 3.5% (LO) and 2.6% (G8).

Original models were used as the starting point for the first iteration. The HMM subsystem was trained on the digits subset of background speaker data, and the GMM subsystem on the name and digits subset.

The solid lines in the DET plots of Figure 10.1 show results for the retrained systems where both speech recognition and speaker verification components have been retrained, while dotted and dash-dotted lines indicate the contribution from retraining the individual components. The figure shows that adapting the speech recognition component improves performance considerable in the gate/hall condition while no effect can be seen in the landline/office condition, while adapting background models in the speaker verification component reduces error rates in both conditions. EER for the solid lines in Figure 10.1 is 2.4% for gate/hall and 3.1% for landline/office. This corresponds to a 63 % relative reduction in EER for gate/hall compared to baseline, and 23% for landline/office. Figure 10.3 shows DET curves for the condition-parallel test sets and the retrained systems.[1] Note that with the retrained systems, performance is better in the gate/hall condition than in the landline/office condition.

[1] See also Figure 7.15 (p. 150) for an alternative comparison using parametric DET curves.

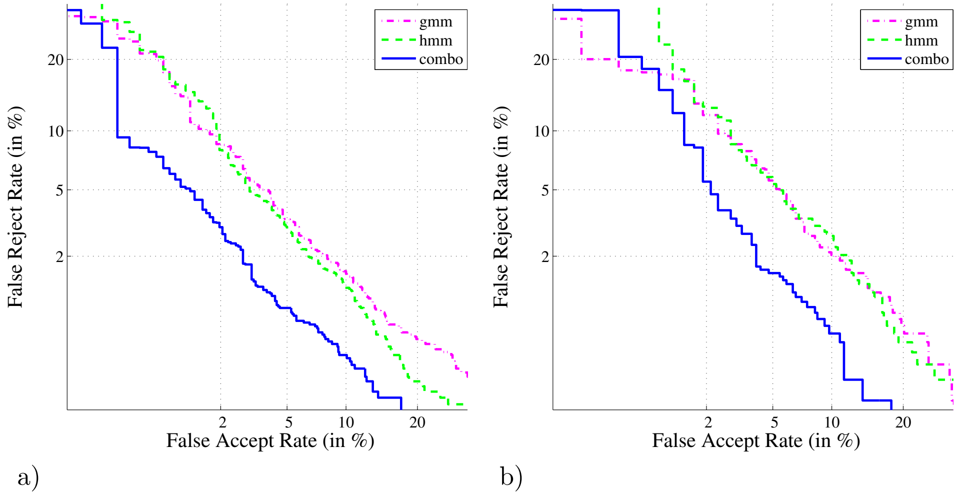


Figure 10.4: DET curves for the retrained system and its individual subsystems in a) gate/hall and b) landline/office conditions.

10.4.3 Fusion

Figure 10.4 shows DET curves for the individual HMM and GMM subsystems along with the combined system, all retrained on PER background speakers. Score combination weights are the *a priori* weights computed on Gandalf data. Clients are enrolled using the full enrollment session (E2a_*c*) and test sets are the single-condition test sets S2b_*c*. The GMM and HMM subsystems exhibit similar error rates in both the gate/hall and landline/office conditions, but note that the GMM subsystem uses more speech data than the HMM subsystem since it uses both the name and the digits. EER in the gate/hall condition is 4.2% and 4.0% respectively for the GMM and HMM subsystems and 2.4% for the combined system; 5.2% for both subsystems and 3.1% for the combined system in the landline/office condition.

10.4.4 Enrollment length

All of the above results were produced using target models trained on the full enrollment session represented by enrollment sets E2a_*c*. This includes 10 repetitions of name and digits for most targets, and 8 or 9 repetitions for a few targets where one or two enrollment utterances were skipped (cf. Section 10.3.1, p. 195). Figure 10.5 compares these results to the case with half of the enrollment data, exactly five repetitions per target, and the retrained systems. Note that background models were the same in both cases. They were trained on full enrollment sessions from each background speaker. EER is 2.4% and 5.3% in the gate/hall condition and 3.1% and 8.8% in the landline/office condition, i.e. the EER is more than doubled

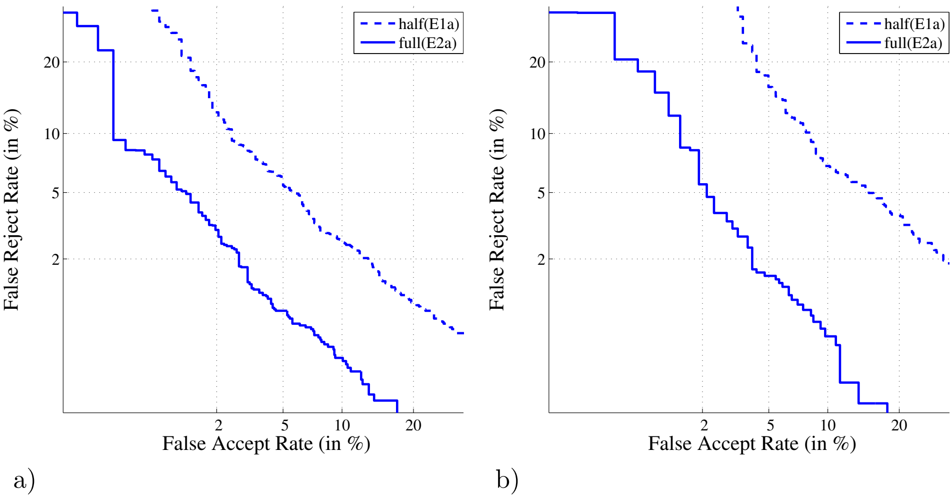


Figure 10.5: Client enrollment using the full enrollment session (E2a_c) and the first half of it (E1a_c) with the retrained system for the a) gate/hall and b) telephone/office conditions. Test sets are the single-condition sets S2b_c.

in the former condition with the reduction in enrollment data, and almost tripled in the latter condition.

10.4.5 Test utterance length

Test utterances collected in the gate/hall condition contain name plus five digits, while those in telephone conditions contain one digit less. Results for single-condition test sets are based on those test utterances directly, and thus the gate/hall condition has a slight advantage over telephone conditions, offered by the use of a display to prompt passphrases. To focus on speaker verification system performance in comparison between conditions, results on condition-parallel test sets in this chapter are produced with one digit removed from every test utterance in the gate/hall condition (with the exception of Figure 10.8 where all five digits were used with the commercial system).

Figure 10.6 displays the effect of the test utterance length directly. In the gate/hall condition, it compares DET curves for the retrained system with a name and two, three, four or five digits. To produce test utterances with less than five digits, digits in one or more random positions within each test utterance have been ignored in the feature vector stream, i.e. delta parameters in feature vectors were computed from the complete waveform to avoid discontinuities. The EER increases from 2.4% for the full test utterance to 2.9%, 3.2% and 4.0% when dropping one, two and three digits, respectively. These results for test utterances with less than five digits should be interpreted as approximate estimates of error rates for real

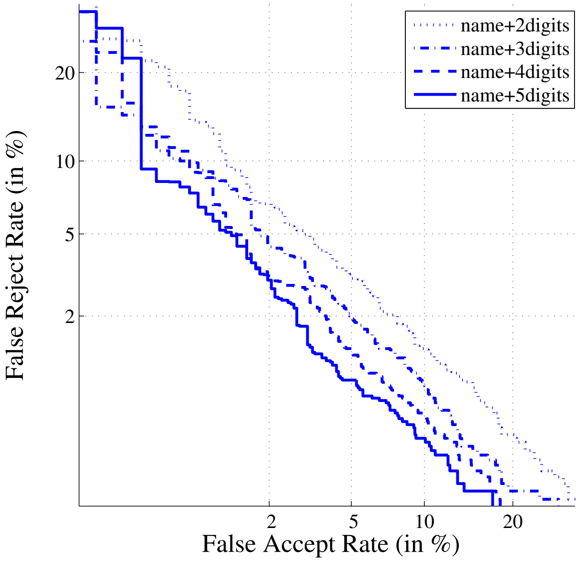


Figure 10.6: DET plots for the retrained system with the gate/hall single-condition test set S2b_G8 and including the name plus two, three, four or five digits in each test.

test utterances with the same number of digits, since synthetic short digit string utterances created by omitting digits from a longer utterance cannot be expected to be exactly equivalent to corresponding real utterances. Naturally, the prosody of the synthetic utterances will not be correct, but it may also be that digits in short strings are pronounced more clearly than longer strings. However, we believe the influence on presented results is small because the ASV system does not explicitly model sentence prosody or word context dependency.

10.4.6 Commercial system

Figure 10.7 shows DET curves for the commercial system for the single-condition test sets S2b_*c* and the gate/hall and landline/office conditions. Results are presented with the full and half session enrollment. EERs are 6.8% and 8.4% in the gate/hall condition and 6.0% and 7.6% in the landline/office condition (24% and 27% relative increase in EER for the two conditions with the reduction in enrollment data). Operating points marked with asterisks in the figure correspond to the EER-threshold determined from the Gandalf development experiment.

A comparison to Figure 10.5 shows that the commercial system performs better with less enrollment data relative to the retrained research systems.

Figure 10.8 compares the four conditions with the commercial system with full-

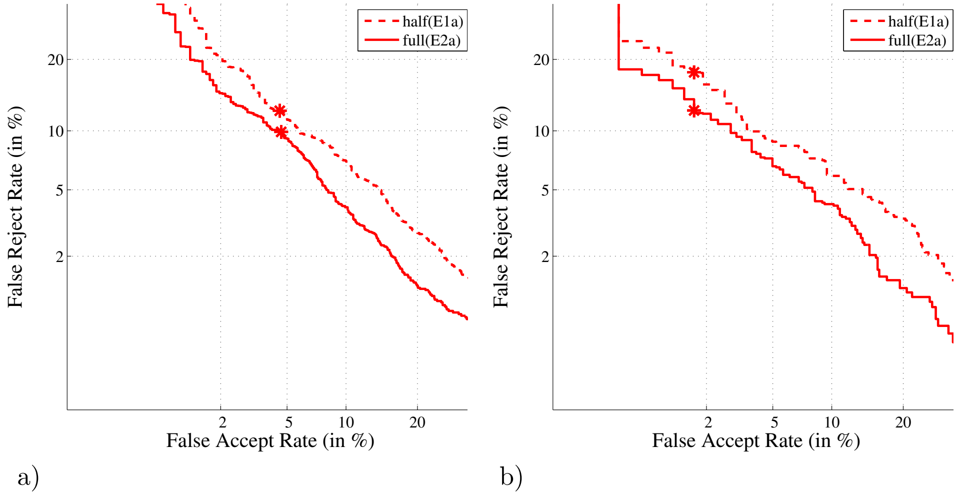


Figure 10.7: DET plots for the commercial system and client enrollment using the full enrollment session (E2a_c) and the first half of it (E1a_c) for the a) gate/hall and b) telephone/office conditions. Test sets are the single-condition sets S2b_c. Asterisks (*) mark the operating points determined by the *a priori* threshold. The speaker adaptation feature is turned off.

session enrollment and condition-parallel test sets. It also includes operating points determined from the EER point on Gandalf development data. As for the baseline research system (Figure 10.2), the operating point for the gate/hall condition is further to the lower right relative to those for the telephone conditions. However, all four points are shifted to the upper left compared to the same system.

10.4.6.1 Speaker adaptation

The commercial system has a speaker adaptation feature that allows a target model to be adapted to a test utterance if the verification score is greater than an adaptation threshold. Figure 10.9 shows DET curves for the commercial system on single-condition test sets (S2b_c) with tests run in a random order, the full enrollment session (E2a_c), and with the adaptation feature turned on. Since the adaptation threshold is specified relative to the decision threshold, an ideal decision threshold for the EER point was determined *a posteriori* for each condition from a previous run on the exact same test data with the adaptation feature turned off. This decision threshold was then used together with the default value on the adaptation threshold. EER with adaptation turned on is 3.2% in the gate/hall condition and 4.0% in the landline/office condition. This is a 53% relative reduction in EER for gate/hall and 27% for landline/office, compared to not using adaptation.

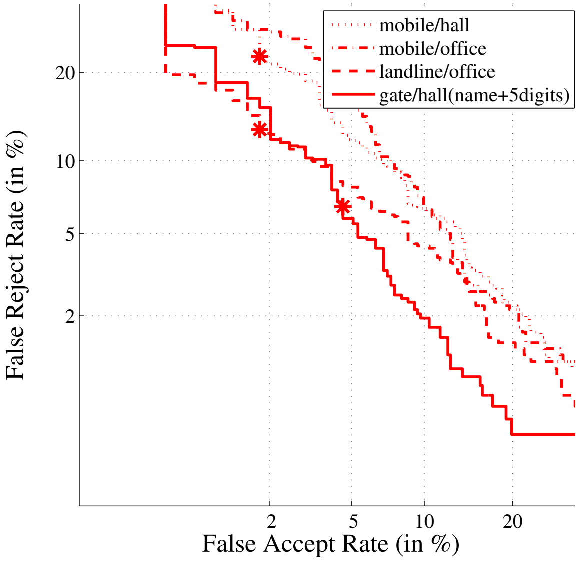


Figure 10.8: A comparison between conditions using the commercial system without speaker adaptation, full enrollment sessions (E2a_c), and the condition-parallel test sets (S2b_Q:c). A name plus four digits is used in telephone conditions and name plus five digits in the gate/hall condition. EERs are 8.4% (MH), 8.7% (MO), 6.4% (LO) and 5.3% (G8). Asterisks (*) mark the operating points determined by the *a priori* threshold.

With speaker adaptation the order of tests is relevant (e.g. Fredouille et al., 2000). In Figure 10.9 two cases were tested: *random* where all tests were run in a random order, and *optimistic* where all true-speaker tests were run before any impostor test. The latter case is an idealized situation for a speaker verification system, and was meant to estimate a lower bound on error rates with speaker adaptation. However, it turned out in Figure 10.9b that error rates are lower with the random order test than with the optimistic.

Table 10.4 shows how many of true-speaker and impostor tests resulted in a model adaptation (for each test the name and digits file were concatenated to form a single file per test).

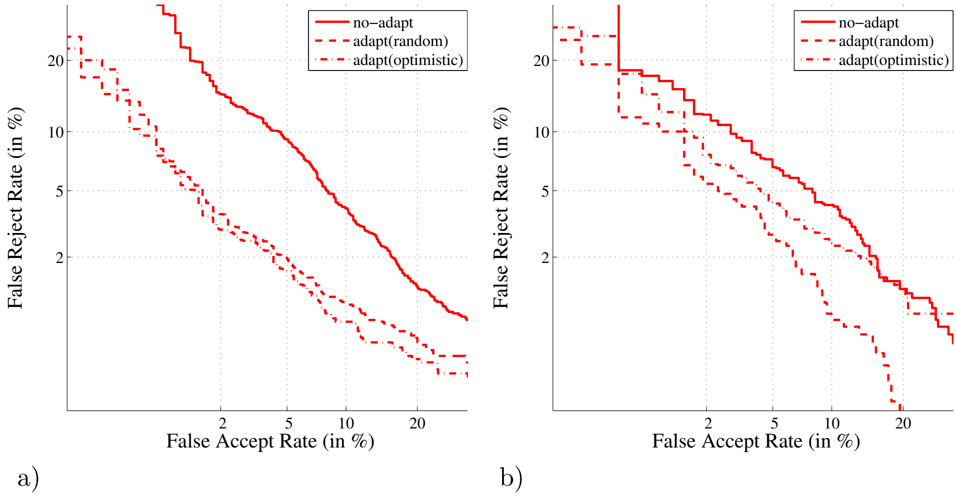
10.5 Discussion

10.5.1 Statistical significance

Table 10.5 summarizes EERs found in this chapter in the gate/hall and landline/office single-condition test sets. Table 10.6 show corresponding results for

Table 10.4: Proportion of true-speaker and impostor tests that resulted in model adaptation and the corresponding false reject (FRR) and false accept rates (FAR) in the experiments presented in Figure 10.9.

Cond.	Adapt	Test order	True-speaker tests	Impostor tests	FRR	FAR
G8	off	-	-	-	6.7%	6.8%
	on	random	97.4%	3.3%	0.75%	12.8%
	on	optimistic	98.1%	4.2%	0.39%	14.9%
LO	off	-	-	-	6.0%	6.2%
	on	random	96.7%	4.5%	0.57%	13.8%
	on	optimistic	96.2%	6.0%	1.55%	16.4%



a)

b)

Figure 10.9: DET plots for the commercial system with and without its speaker adaptation feature turned on for the a) gate/hall and b) telephone/office conditions.

condition-parallel test sets. However, to allow the computation of post-trial confidence intervals (CI) based on Section 2.5.2, where we only considered false reject rates, we present EERs as if they were FRRs and compute CIs for the FRR. Basing a CI on an FRR this way is not statistically sound since EER is based on an *a posteriori* decision threshold determined after observing all the true-speaker test scores we are analyzing, plus a series of impostor test scores. The *a posteriori* threshold introduces a dependency between observations of decision errors. By treating observations of EER as observations of FRR we have basically assumed that previous experiments on development data resulted in a threshold that exactly meets an EER criterion on evaluation data (this was obviously not the case in most of our experiments). We claim the method still gives an idea of the uncertainty in our results.

The tables present 95% post-trial CIs for the "true" overall false reject rate given an observation of a fraction of errors $\hat{p} = x/N$ using two different methods. In both methods, intervals are computed from the binomial distribution as defined by (2.15). The methods differ in how the value for N in the binomial distribution (2.14) is determined:

- **Method 1**: N equals N' as determined by Eq. (2.18), i.e. such that the variance $\hat{p}(1-\hat{p})/N'$ of the fraction of errors predicted by the binomial equals the variance $s_{\hat{p}}^2$ in the estimate of $\hat{p}$ estimated from Eq. (2.22). In the computation of $\hat{s}_{\hat{p}}^2$, false reject rate p_i for target i is simply the fraction of errors observed for this target (defined as the non-parametric ML method in Eq. (7.1), p. 134; $p_i = \text{FRRd}(i)$). This is the "best practice" approach suggested by Mansfield and Wayman (2002), but we use the binomial directly to compute intervals, instead of its normal approximation.

- **Method 2**: N is fixed for a given test set and equals N' computed according to Eq. (6.1) with $\rho = 0.2$.

Since the variance estimation step in Method 1 can be viewed as a way to determine ρ, the resulting values of ρ are included in the tables. After estimating the variance $\hat{s}_{\hat{p}}^2$ with (2.22), ρ was computed relative to the adjusted total number of tests N^* (defined in Section 6.3.4.6) using (2.20) and (2.19) with N substituted by N^* and n substituted by $\lfloor \bar{n} \rfloor^*$, i.e. by solving for ρ in

$$1 + (\lfloor \bar{n} \rfloor^* - 1)\rho = \frac{N^* \hat{s}_{\hat{p}}^2}{\hat{p}(1-\hat{p})}. \tag{10.1}$$

Figure 10.10 shows the binomial distributions behind the confidence intervals for the retrained research (combo) system and single-condition test sets. Appendix F provides corresponding plots for condition-parallel test sets for the retrained combo system (Table F.3) and for the baseline research system and the commercial system (Table F.4 and F.5).

Table 10.5: Summary of observed FRR (%) with 95% confidence intervals computed from the binomial distribution with Methods 1 and 2 to select N. In all cases, the threshold equals the *a posteriori* EER (EERd) threshold. Systems above the dashed line within each method section of the table have been retrained on PER-specific, condition-dependent background data.

Test set	T2b_G8			T2b_LO		
ASV system	FRR	interval	ρ^a	FRR	interval	ρ^b
<Method 1>						
Combo, retrained						
- full enrollment	2.4	(1.2–3.6)	0.066	3.1	(0.0–6.1)	0.285
- half enrollment	5.3	(3.5–7.2)	0.076	8.8	(3.8–14.3)	0.267
GMM, retrained						
- full enrollment	4.2	(2.4–6.2)	0.101	5.2	(1.8–10.2)	0.259
HMM, retrained						
- full enrollment	4.0	(2.4–5.8)	0.093	5.2	(2.1–8.7)	0.141
Combo, baseline						
- full enrollment	6.4	(3.8–9.3)	0.156	4.0	(0.8–7.8)	0.240
Commercial system						
- full enrollment	6.8	(4.0–9.9)	0.181	6.0	(1.1–11.2)	0.320
- half enrollment	8.4	(5.1–11.9)	0.214	7.6	(1.6–14.3)	0.463
<Method 2>	$\rho = 0.2$ ($k = 10.8$)			$\rho = 0.2$ ($k = 8.8$)		
Combo, retrained						
- full enrollment	2.4	(0.8–4.4)	0.200	3.1	(0.7–6.6)	0.200
- half enrollment	5.3	(2.8–8.4)	0.200	8.8	(4.4–14.0)	0.200
GMM, retrained						
- full enrollment	4.2	(2.0–6.8)	0.200	5.2	(1.5–9.5)	0.200
HMM, retrained						
- full enrollment	4.0	(1.6–6.4)	0.200	5.2	(1.5–9.5)	0.200
Combo, baseline						
- full enrollment	6.4	(3.6–9.7)	0.200	4.0	(0.7–7.4)	0.200
Commercial system						
- full enrollment	6.8	(4.0–10.0)	0.200	6.0	(2.2–10.3)	0.200
- half enrollment	8.4	(5.2–12.1)	0.200	7.6	(3.7–12.5)	0.200

[a] $N^* = 2700$ in calculations of ρ (cf. Table 6.12)

[b] $N^* = 1200$ in calculations of ρ (30 targets with $\lfloor \bar{n} \rfloor = 40$ true-speaker tests/target)

Table 10.6: Summary of observed FRR (%) with 95% confidence intervals computed from the binomial distribution with Methods 1 and 2 to select N. In all cases, the threshold equals the *a posteriori* EER (EERd) threshold. Systems above the dashed line within each method section of the table have been retrained on PER-specific, condition-dependent background data.

Test set	T2b_Q:G8[a]			T2b_Q:LO			T2b_Q:MO			T2b_Q:MH		
ASV system	FRR	interval	ρ	FRR	interval	ρ	FRR	interval	ρ	FRR	interval	ρ
<Method 1>												
Combo, retrained												
- full enrollment	2.6	(0.0–5.5)	0.277	3.5	(0.0–7.4)	0.314	4.8	(1.0–9.4)	0.261	5.3	(2.7–8.0)	0.067
Combo, baseline												
- full enrollment	5.1	(2.0–8.2)	0.113	4.3	(1.1–8.9)	0.280	5.8	(2.3–9.1)	0.130	6.4	(3.6–9.9)	0.096
Commercial system												
- full enrollment	5.3[b]	(2.5–8.5)	0.110	6.4	(1.2–12.0)	0.306	8.7	(4.2–14.4)	0.207	8.4	(3.0–14.1)	0.252
<Method 2>	$\rho = 0.2$ $(k = 8)$			$\rho = 0.2$ $(k = 8)$			$\rho = 0.2$ $(k = 8)$			$\rho = 0.2$ $(k = 8)$		
Combo, retrained												
- full enrollment	2.6	(0.0–5.0)	0.200	3.5	(0.8–7.5)	0.200	4.8	(1.6–9.1)	0.200	5.3	(1.6–10.0)	0.200
Combo, baseline												
- full enrollment	5.1	(1.6–9.1)	0.200	4.3	(0.8–8.3)	0.200	5.8	(1.6–9.9)	0.200	6.4	(2.5–10.8)	0.200
Commercial system												
- full enrollment	5.3[c]	(1.6–9.9)	0.200	6.4	(2.5–10.8)	0.200	8.7	(4.1–14.1)	0.200	8.4	(4.1–13.3)	0.200

[a] using name and four digits, where not otherwise specified

[b] using name and five digits

[c] using name and five digits

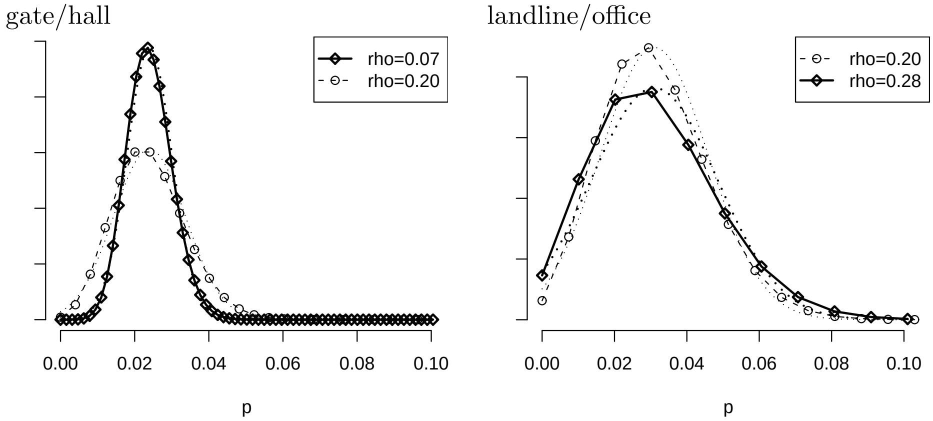


Figure 10.10: Binomial distributions used to compute confidence intervals for the retrained research system, test sets T2b_G8 and T2b_LO and full enrollment (E2a_c). ρ (rho) for solid lines with diamonds are computed with Method 1, while the distributions for $\rho = 0.20$ (dashed lines with circles) correspond to Method 2 with an *a posteriori* choice of ρ. The normal approximation to each binomial is shown as a dotted line.

For Method 2, we have chosen[2] a constant intra-speaker correlation coefficient $\rho = 0.2$ corresponding to values for k between 8 and 11 for the different test sets. Our prior belief about k was that a constant $k = 2$ would be a good value. Compared to values for ρ (and k) resulting from Method 1 based on an estimated variance, we chose to show intervals for a constant $\rho = 0.2$ instead, being the average over all ρ values found with Method 1 within Tables 10.5 and 10.6, respectively. Thus, the CIs shown for Method 2 in the table are based on an *a posteriori* choice of ρ.

The potential usefulness of Method 2 lies in predicting pre-trial CIs rather than estimating post-trial CIs. We include results from Method 2 here for comparison. The motivation for a constant ρ in Method 2 is that the “intra-speaker correlation” should depend mainly on the speakers and not so much on the particular test set or ASV system under test.

The evaluation strategy of Bolle et al. (2004) (applied to fingerprint data) should be applied also to ASV data to evaluate post-trial CI estimation methods. With this strategy, a corpus is randomly divided into two disjunct halves. CIs are estimated with each method on one half and compared to the “true” error rate estimated on the other half. The procedure is then repeated a number of times to estimate the coverage, i.e. the probability that a CI covers the true error rate[3].

[2] this choice is discussed in Section 11.2

[3] ideally, a 95% confidence interval should have a coverage of 95%.

Given confidence intervals from Method 1 in Tables 10.5 and 10.6, we can get an idea of which experimental differences observed in this chapter are statistically supported, and which are not, by comparing confidence intervals. Well separated, non-overlapping confidence intervals indicate strong support for the difference, while intervals that overlap to a great extent indicate no support for a difference. Note that to make formal conclusions about differences being statistically significant or not, the results normally require more rigorous analysis. In particular, our confidence intervals are derived to say something about measurements on one ASV system compared to some underlying "true" value. Comparing two systems on the same speech data, or comparing the performance of a single system on different types of data, requires other types of statistical tests, for example McNemar's test (e.g. Siegel, 1956).

Informally comparing[4] confidence intervals in the tables, we find for example:

- A positive effect from retraining the (combined) research system on PER-specific, condition-dependent tuning data is well supported in the gate/hall condition by results on the single-condition test set T2b_G8, while it is not supported in the telephone conditions. On the condition-parallel gate/hall test set (T2b_Q:G8) this difference is weakly supported.
- Performance degradation from halving the amount of enrollment data with the retrained (combined) research system is supported in the gate/hall and landline/office conditions.
- An improvement from combining the (retrained) HMM system with the GMM system (including the additional use of proper names for verification) is weakly supported in the gate/hall condition, and not support in the landline/office condition.
- Difference in performance of the retrained (combined) research system between the four conditions are not supported, except for the difference between the gate/hall and mobile/hall condition which is weakly supported.

10.5.1.1 McNemar tests

McNemar's test for the significance of changes (e.g. Siegel, 1956) is a non-parametric test that can be applied to pair-wise related measures on a nominal scale (labeled data). To apply this test in speaker verification with good theoretical justification, FRR and FAR should be treated jointly somehow (Bengio and Mariéthoz, 2004b). For simplicity, however, we will take the same approach as above and compare false reject error rates only, at a global *a posteriori* EER threshold. The problem is

[4] We used the following definitions: Call two cases under comparison case A and case B, and the estimated false reject rates and confidence intervals from the two cases $\hat{p}_{\mathrm{A}}$, $\hat{p}_{\mathrm{B}}$, $\hat{\mathrm{CI}}_{\mathrm{A}}$ and $\hat{\mathrm{CI}}_{\mathrm{B}}$. A difference is *well supported* when $\hat{\mathrm{CI}}_{\mathrm{A}}$ and $\hat{\mathrm{CI}}_{\mathrm{B}}$ are non-overlapping; *supported* when $\hat{p}_{\mathrm{A}} \notin \hat{\mathrm{CI}}_{\mathrm{B}}$ and $\hat{p}_{\mathrm{B}} \notin \hat{\mathrm{CI}}_{\mathrm{A}}$; *weakly supported* when $\hat{p}_{\mathrm{A}} \notin \hat{\mathrm{CI}}_{\mathrm{B}}$ but $\hat{p}_{\mathrm{B}} \in \hat{\mathrm{CI}}_{\mathrm{A}}$; and *not supported* if $\hat{p}_{\mathrm{A}} \in \hat{\mathrm{CI}}_{\mathrm{B}}$ and $\hat{p}_{\mathrm{B}} \in \hat{\mathrm{CI}}_{\mathrm{A}}$.

again that FRR observations are then dependent through the threshold and also depend on impostor tests.

To compare two cases, say A and B, with the McNemar test, we compare individual FRR for each target speaker and determine if the FRR is higher or lower in case B than in case A, assuming each target has the same number of tests in both cases. Denote as $p_{\mathrm{A}i}$ and $p_{\mathrm{B}i}$ the FRR for target i in the two cases. Denote as M_{AB} the number of targets for which $p_{\mathrm{A}i} < p_{\mathrm{B}i}$ (better result in case A than in case B)[5], and as M_{BA} the number of targets for which $p_{\mathrm{B}i} < p_{\mathrm{A}i}$. Designate as the null hypothesis H_0 that there is no difference between cases A and B. Under H_0, expected values of both M_{AB} and M_{BA} would then equal $(M_{\mathrm{AB}} + M_{\mathrm{BA}})/2$. The McNemar test tests if observed values M_{AB} and M_{BA} are sufficiently different from their expected values. It proceeds by computing the test statistic

$$T_{\chi^2} = \frac{(|M_{\mathrm{AB}} - M_{\mathrm{BA}}| - 1)^2}{M_{\mathrm{AB}} + M_{\mathrm{BA}}} \tag{10.2}$$

and the probability of the value T_{χ^2}, or a more extreme value, under the χ^2-distribution with one degree of freedom ($df = 1$). If this probability is less than $(1-\alpha)/2$ (two-sided test), H_0 is rejected in favor of the alternative hypothesis, that there is a difference between cases A and B. We use the same level of significance $\alpha = 0.05$ as with confidence intervals above.

Table 10.7 shows the results of applying McNemar's test to some of the comparisons made in this chapter. Note that the results from McNemar are consistent with findings from our comparisons of confidence intervals above. All differences that were found to be at least *weakly supported* in the comparison of confidence intervals were found statistically significant with the McNemar test, while differences that McNemar tests did not find statistically significant were found *not supported* by confidence interval comparison.

Note that the McNemar test does not take into account the magnitude of differences in individual FRR between the two cases, only the sign. Since our measures are ordinal (FRR differences can be ranked with respect to their magnitude), the Wilcoxon matched-pair signed-ranks test (e.g. Siegel, 1956) could also be used, which does take the magnitude of differences into account. This is a more powerful statistical test. However, since our approach of comparing FRR at case-dependent *a posteriori* thresholds introduces dependencies between speakers, and thus the assumptions behind both tests are not quite true, we decided to use the more "blunt" McNemar test instead.

10.5.2 Length of enrollment and test data

It is clear from Figure 10.5 and confidence intervals in Table 10.5 that the (retrained) research system benefits from the rather large number of repetitions of name and

[5] M_{AB} is equivalently the number of targets for which fewer false reject errors are observed in case A than in case B, given our assumption about an equal number of tests per case for each target.

Table 10.7: Results of McNemar's test of differences at 5% level of significance.

case A	case B	common	test set	p[a]	diff[b]
Effect 1: *retraining on PER condition-specific background data*					
baseline	retrained	combo system, full enrollment	T2b_G8	<0.001	x
			T2b_LO	1.0	-
			T2b_Q:G8	0.016	x
			T2b_Q:LO	1.0	-
			T2b_Q:MO	0.24	-
			T2b_Q:MH	0.45	-
Effect 2: *reducing enrollment data by a factor two*					
full	half	combo system, retrained	T2b_G8	<0.001	x
			T2b_LO	0.002	x
		commercial system	T2b_G8	0.015	x
			T2b_LO	0.34	-
Effect 3: *combining HMM subsystem with GMM subsystem*					
HMM	combo	retrained systems, full enrollment	T2b_G8	0.004	x
			T2b_LO	0.043[c]	-
Effect 4: *changing PER condition*					
G8	LO	combo system, retrained	T2b_Q:c	1.0	-
G8	MO			0.30	-
G8	MH			0.006	x
LO	MO			0.15	-
LO	MH			0.015	x
MO	MH			0.33	-

[a] probability that test statistic x has observed value T_{χ^2} or greater ($P_{\chi^2}(x >= T_{\chi^2})$)

[b] 'x' indicates a statistically significant difference detected by a two-sided test at $\alpha = 0.05$

[c] difference would have been significant with a one-sided test

digits in the full enrollment session, since cutting it to half more than doubled the EER. The same is not true for the commercial system, for which the EER increased by only about 25%. This difference between the two systems can be partly explained by target model size: gross model size is about five times larger for the research system than for the commercial system after compressing each model set using Lempel-Ziv coding to partly compensate for an ineffective storage format used with the research system. The research system was dimensioned to operate with rather large amounts of enrollment data.

Test utterances in this study are an order of magnitude shorter than the total length of enrollment and consist of a name and a string of digits. It was argued in Section 5.3 (p. 81) that it is difficult to collect digit strings with more than four digits from users in a telephone application through aural prompts, but collecting longer digit strings through visual prompts as in the gate case should be feasible. In this study we collected only five digits per utterance in the gate/hall condition, and simulated the use of two, three and four digits per utterance with the results in Section 10.4.5 (p. 201) and Figure 10.6. Figure 10.11 shows a prediction of EER for test utterances with longer digit strings, based on an exponential fit to the EERs of Figure 10.6 extended with the EER for the corresponding experiments with a name only, and a name plus a single digit. It suggests that the EER with a name plus six digits would be 1.8%, a 37% relative reduction compared to 2.8% for a name plus four digits. The prediction of 1.1% EER for a name plus eight digits is uncertain because, firstly, it is not evident that the exponential prediction model is valid for longer digit strings; and, secondly, it is also not evident that users would accept such long strings, and it is likely that they would generate significantly more disfluencies, such as substitutions, hesitations, repairs, etc. Such disfluencies are likely to generate errors in the speech recognition process and the resulting segmentation used by the speaker verification system.

A complementary approach for collecting more test data efficiently is to rely on a sequential decision strategy such as a heuristic method (Furui, 1981; Naik and Doddington, 1986) or one based on Wald's sequential probability ratio test principle (Lund and Lee, 1996; Surendran, 2001).

10.5.3 Effects of fusion

Fusion results in Section 10.4.3 (p. 200) show a large error rate reduction from each of the individual systems to their combination. This may be surprising since both subsystems are based on similar features, classifiers and normalization techniques, and their output score should therefore be correlated and not be very good candidates for score fusion. However, one major difference is their use of data: the GMM subsystem uses both names and digits, while the HMM subsystem ignores the name and uses the digits only. To understand/explain the underlying factors, we tested the separate and combined systems on the individual and combined parts of the gate/hall test utterances. In all cases were clients enrolled using the full name plus

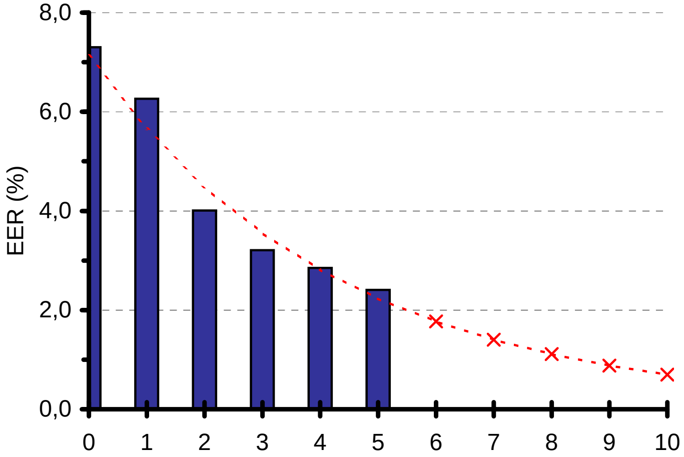


Figure 10.11: The EER values for the retrained system with the gate/hall single-condition test set S2b_G8 and including the name plus zero through five digits in each test (bars) and a prediction (dashed line) of EER values for a different number of digits using an exponential model (cf. Figure 10.6).

digits enrollment set (E2a_c). EERs are shown in Table 10.8, where cases C, D and F match the DET curves included in Figure 10.4a.

Using the cases in the table, the formation of the final result (case F) can be illustrated with the two alternative paths of information fusion shown in Figure 10.12. Each path consists of conceptual information fusion along two axes: score fusion of separate systems and vocabulary fusion of the name and the digit parts of the test utterance. The lower path (via case E) consists of one fusion step along each axis: a system fusion on the digits part of the test utterance, followed by fusion of the two parts of the utterance. Hence, each step combines independent sources of information. The upper path (via case C) more closely reflects the actual structure of the ASV system, but it contains the fusion (C,D)→F with simultaneous fusion along both axes, since it combines system scores based on different parts of the test utterance, and hence combines information sources that are not independent. We propose that the lower path better explains the formation of the system output (from a conceptual point of view).

10.5.4 On-site vs. telephone use of ASV

Is there a greater potential for well-performing ASV in an on-site application than in a telephone application? This is a very general question, and of course we don't have a foundation to answer it in a general sense, but we do have some clues for our particular application instances.

Table 10.8: EER for the individual subsystems and their combination applied to digits-only or name-only subsets of the S2b_G8 test set, or the complete test set using both name and digits. The enrollment set is the full (name and digits) E2a_G8. The combined system in case E uses the same score combination weights as the system in case F.

Case	System	Vocabulary	EER
A	gmm	name	7.3%
B	gmm	digits	6.9%
C	gmm	name, digits	4.2%
D	hmm	digits	4.0%
E	combo	digits	3.4%
F	combo	name, digits	2.4%

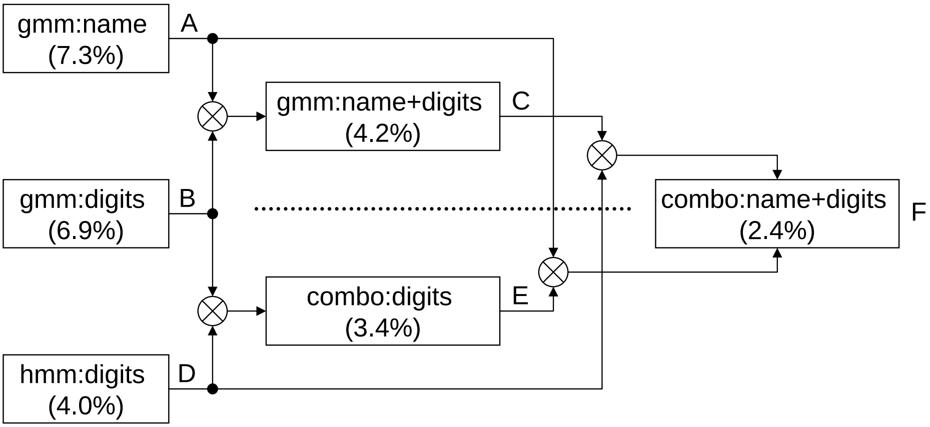


Figure 10.12: Two alternative conceptual information fusion paths that explain how the output of the ASV system is formed. Numbers within parentheses are the measured EERs for the gate/hall condition with the system and test utterance content represented by each box.

We believe our comparison between ASV in the on-site and telephone version of PER is fair. First, the design differences introduced between the on-site and telephone versions of PER are well founded. For example, verification based on a client's proper name and a digit sequence collected in a single utterance, where the digits are visually prompted, works well in the on-site application, while the name and digits must be separated in the telephone case. Aural prompts are the only alternative with most telephones (since they don't have a display). Second, our data collection procedure and design of the condition-parallel test sets based on series of sessions recorded in chronological proximity in the four conditions, allow for similar prerequisites in the four conditions. If a subject suffered a head cold during a gate session, the same was true in telephone sessions within the same series. If there was a noisy background, it was probably there both during a gate session and a corresponding mobile telephone session in the hall. Random between-session variation in for example background noise will naturally have occurred, but such variation can only be excluded by stereo recordings, and stereo recordings in our case would have meant a more artificial context for the recordings. Systematical differences between conditions may also have occurred, however. For example, in many series the gate/hall session was made before telephone sessions because it was recorded when the subject arrived to work in the morning. Since a number of steps had to be climbed to reach the gate, (true-speaker) subjects might have been more out-of-breath at the gate than after arriving in the office and sitting down to make the landline/office call. There is also the possible difference in motivation in subjects, since the gate version of PER could actually open the door, while telephone calls were made for recording purposes only.

Results from the condition-parallel test sets indicate that, provided acoustic models in the speech recognition and speaker verification components are tuned using proper development data, ASV error rate may be lower in the on-site condition than in all three telephone conditions, though a statistically significant advantage was measured only relative to the mobile/hall condition (Table 10.7). With acoustic models trained on a general-purpose telephone corpus, little or no difference was seen between the conditions. The performance difference introduced by tuning on proper development data was large for the on-site application and non-significant for the telephone application. This highlights an important difference between using a variety of ubiquitous telephone handsets vs. using a single microphone in a particular room in an on-site application: the need for dedicated tuning data is larger for the on-site application than for telephone applications. It should be easier to create an ASV system that will perform consistently at a near optimum error rate between instances of telephone applications without tuning it for every particular application, while tuning data will be important for any on-site application. To achieve the best possible performance, however, tuning data from the application will usually be needed in either case.

Our point estimates of EER in the gate/hall and landline/office conditions were 2.6% vs. 3.5% on the condition-parallel test sets using the same number of digits in the test utterance. While this corresponds to a 25% relative reduction for the on-site application, the statistical uncertainty in the estimates is large and we can not infer a difference between the two conditions based on these measurements alone. But that was with the same number of digits. In our on-site version of PER we used five digits that caused no apparent trouble for subjects, and we believe six digits would have worked well too. Considering Figure 10.11, we would expect a 37% relative reduction in EER for six digits compared to four, suggesting the 2.6% EER for the on-site application could be reduced to 1.6%[6]. We are then up to a 54% reduction for gate/hall relative to the landline/office condition. The corresponding reductions are 67% relative to mobile/office and 70% relative to mobile/hall.

We can further speculate into factors we have not tested. In our experiments with on-site data, we used a downsampled 8 kHz version of the original wide-band audio recordings made at 16 kHz. Given good development data for wide-band speech and a proper modification of the system's speech feature representation to operate with 16 kHz data, it should be possible to achieve a further reduction in error rate in the on-site system. Furthermore, we saw in Section 6.3.4.5 that the proportion of different-number calls (test calls from a different telephone number than the target's enrollment call) was higher in the impostor part of the telephone condition test sets (around 25%) than in the true-speaker part, suggesting that in a fully same-channel test set error rate in telephone conditions could have been higher than what we saw in our data.

To conclude the discussion on the potential for well-performing ASV in an on-site application vs. in a telephone application, our data suggests that, given the availability of application-specific tuning data, ASV error rate may be less than half in an on-site application than in a corresponding telephone application.

[6] Note that Figure 10.11 and the previous discussion on test utterance lengths are based on the single-condition test set S2b_G8, while in this paragraph we look at the condition-parallel test set S2b_Q:G8.

Part V

Concluding remarks

Chapter 11

Concluding remarks

11.1 Summary

The introductory part of the thesis gave an overview of the field of speaker recognition, to serve as a background. A basic taxonomy of speaker recognition tasks was given and basic conditions for recognizing individuals by means of their voices were discussed around the basic concepts of inter-speaker and intra-speaker variability. The most common techniques proposed for speaker recognition were then surveyed from a somewhat historical perspective, including feature extraction, classification and score normalization techniques. A background for assessment of speaker recognition system performance was also given, along with some basic requirements on corpora for speaker recognition research and development.

The central parts II–IV of the thesis dealt with three broad aspects related to the task of speaker verification: system and application design, assessment methods and practical experiments.

Part II was concerned with system and application design on different levels, from detailed description of speaker verification methods up to an example of a complete application where speaker verification was used. First, two speaker verification systems were described in detail from a mathematical point-of-view. A number of aspects of the text-dependent HMM-based system were the focus of the author's work during 1995-1999. Aspects included the overall structure of the system as well as more detailed aspects such as methods for estimating target-dependent variances, the choice of cohort method and background models, and methods for segmenting utterances into words for use with the word-dependent HMMs. Except for the investigation on segmentation methods (Melin, 1998), all these aspects were covered more or less explicitly in the thesis. The design and implementation of the GMM system was the work of Daniel Neiberg under the author's supervision. Its description is included for completeness since it was used in other parts of the thesis and because not all parts of the description were published elsewhere. The chapter on speaker verification systems was concluded with a description of the

GIVES software framework which has been used to implement the systems.

The next level of system design was covered by ATLAS, a software framework for building prototype applications with speech technology. The purpose of the framework was to include many parts of a speech technology-based system in an application-independent way, such that it provides a high-level speech technology interface to the creator of the actual application. Speech technology components connected to ATLAS and example applications built on it were listed.

Finally, a complete application using speaker verification was described, the PER application. The application was implemented on top of the ATLAS framework and used GIVES and a score-level combination of the two described ASV systems to verify the claimed identities of its users. The application was implemented in two versions, an on-site access control system securing a barred gate in a reverberant stairway, and a mock-up telephone version. The telephone version was created to support a parallel data collection of on-site and telephone speech.

Part III covered tools and methods for assessment of speaker verification systems. First, two Swedish speaker verification corpora were described, the Gandalf and the PER corpora. Gandalf is a telephone corpus with 86 client speakers and 30 background speakers containing a mixture of text-dependent and text-independent material recorded with a tape-recorder metaphor, while PER is a text-dependent corpus of parallel on-site and telephone speech containing 54 client speakers and 79 background speakers recorded during actual use of an ASV system. The design of baseline experiments for the European-English Polycost database was then reviewed in retrospect by surveying work that has used the database. A chapter was then dedicated to estimation methods for false reject and false accept error rates in speaker verification. In particular, the parametric approximation of score distributions by means of normal distributions and its possible use in robust error rate estimation were proposed and compared to the standard non-parametric estimation method on PER data. ML and MAP implementations of the two methods were derived and it was shown that conjugate prior distributions used with the MAP variants fit well to observed PER data for our speaker verification system. In particular, it was found that the distribution of false reject and false accept rates over target speakers are well described by beta distributions. Applications of the parametric method include estimation of error rate for individual target speakers and incremental estimation of error rates.

Part IV comprised three chapters on speaker verification experiments.

The first experiments-chapter compared the use of aural vs. visual prompting of digit strings to claimants using Gandalf data. It was found that aural prompts resulted in more speaking-errors in user responses than visual prompts, in particular when prompts included five instead of four digits. It was also found that, given visually prompted (read) enrollment data, visually prompted test utterances resulted in a slightly lower error rate than aurally prompted test utterances, suggesting there is a difference in how digit sequences are spoken in response to the two types of prompts.

The second chapter compared several variance estimation methods in the context of word-level target speaker HMMs using three different telephone corpora. In particular, a number of variance flooring methods with different levels of resolution were proposed. While the best variance flooring methods resulted in slightly lower error rates than using target-independent variances copied from a gender-dependent background model, the latter method was found to be more robust, and it is much simpler to use.

The third chapter on ASV experiments gathered many results from the PER system. Some of the results originated directly from live use of the system, while most presented results were from off-line simulations using well-defined data sets from the PER corpus. Results include comparisons of the original (baseline) ASV system used during the data collection and systems retrained on application-specific tuning data; comparison of the performance on on-site, landline and mobile telephone data; effects of the amounts of enrollment and test data; and on system fusion. Beside our research systems, an anonymous commercial ASV system was also used, to calibrate the recognition tasks and to validate the research systems as state-of-the-art recognition systems. EERs around 5% were found for the retrained research system on mobile telephone data, 3.5% on landline telephone data and 2.4% on the on-site data. It was argued that performance in the on-site application could be improved further by using wide-band audio and by increasing the length of visually prompted passphrases, to EERs below 2%.

A large part of the work described in this thesis involved creating research tools, including speaker verification corpora with corresponding data set definitions, and software platforms for creating speaker verification systems and speech technology based applications in general. The tools were then used more or less extensively in experiments also included in the thesis, but the "pay-off" in our own research work is not very large compared to the amount of work invested in creating the tools. However, the tools have also been used by others as cited in the thesis, for research and educational purposes and in one case for commercial purposes. To summarize work done by others with our tools, GIVES has been used in ten MSc theses projects and one research project; ATLAS in one PhD project (Pakucs), eight MSc or BSc theses projects and three research projects (including two EU projects); and Gandalf in one PhD project (Olsen) and five MSc projects. The total number of MSc or BSc projects where at least one of the tools were used is 17. They were spread over the years 1999–2004 and constituted 25% of all completed MSc/BSc projects at the Speech group in the Department during the same period.

11.2 General discussion

11.2.1 Performance prospects

Research presented in this thesis has been on automatic speaker verification in the context of its commercial applications, characterized by co-operative users, user-friendly interfaces, and requirements for small amounts of enrollment and test data.

A logical question at this point is: What performance can be expected from ASV in the near future?

Much of the work has been directed towards answering this question. As we pointed out in the introductory overview chapter on speaker recognition, a universal answer does not make much sense, since performance depends on many factors related to the application, for example signal quality, text-dependence modes, amounts of enrollment and test data, and the particular user. What we have done is instead to collect speech data representative of some conceivable applications in terms of such factors, to build a state-of-the-art ASV system for evaluation, and to measure what error rates we could achieve. We also developed one particular application (PER) that we implemented and let users try "for real", using a text-dependent mode and what we think are reasonable amounts of enrollment and test data. As quoted in the above summary, error rates corresponding to an overall EER of approximately 2.4% were found for the on-site version of this application, with a potential for further improvements to below 2%. The corresponding EER for the telephone versions were a few percent-units higher.

We will now discuss the validity of these error rates in relation to the question on what performance can be expected from ASV in the near future. Pertinent issues are subject selection, competitiveness of our ASV system, and the relevance of application factors.

A key limitation in our studies is the biased selection of subjects. Subjects in both the Gandalf and the PER corpora are biased in that people with high education in general, and with an education or profession related to speech and language in particular, are over-represented. We would expect a user group not biased in this respect to experience higher error rates on average, at least during an initial learning phase. For example, compared to experienced users of speech technology, unfamiliar users may be expected to be less patient with system mistakes, speak less consistently from session to session, over-articulate, etc.

Technically, lower error rates than those found in our studies can be expected by increasing the amount of enrollment data per user, and even more so for the amount of test data. For example, the Dialogues Spotlight Consortium (2001) found an average same-sex half total error rate (HTER; roughly comparable in magnitude to EER if the operating point in the HTER case is near the *a posteriori* EER operating point) as low as 0.9% using telephone test data consisting of nine utterances (a total of 19 digits and five non-digit words) and large amounts of mixed-type enrollment data in English, using the Nuance Verifier. We believe these amounts of enrollment and test data are not viable in a commercial application, at least not if requested from users in every session and if the goal of the application is to serve a general population. Example results from the same study with quite realistic amounts of data are a HTER of 1.6% using three repetitions of an eight-digit account number for enrollment and a single repetition of the same number for test, or 1.9% similarly using a nine-digit membership number. (These error rates are about half of what we found for the landline telephone version of PER; this is discussed below.) In this example, the error rate (HTER) was reduced to about a half by using a lot more data — at the cost of users having to spend more time on enrollment and test. However, using a good incremental decision strategy, where most decisions are taken after one or two utterances, and more is needed only in a few cases, it may be possible to reach down towards the 0.9% error rate and still keep users happy, at least the security-aware ones.

Is our ASV system state-of-the-art, or are there better systems? Except for the use of a downsampled signal in the on-site version of PER, we claim our system is near state-of-the-art, and that our results are representative of what error rates can be achieved today for the test cases we used. Of course, there is always room for improvement, and there are still many proposed methods in the literature we have not tried. Especially, improvements can be expected from fusing multiple systems (as we already saw an example of in this thesis). However, we also tested a competitive commercial system on our data, with comparable results.

We then need to relate to the lower error rates found in the Dialogues Spotlight Consortium study cited above. They found average same-sex HTER around 1.5% using less enrollment data (about one third of a full PER enrollment session) and about the same amount of test data as in the PER study. We identify the main difference between the two studies in that the Spotlight test case is more text-dependent since it is based on fixed passphrases. In the PER case, the proper name is fixed and the order of digits is randomized. We have seen in other tests (e.g. Nordström et al., 1998) that Nuance Verifier performs better with fixed phrases than with randomized word strings. We think that, in general, the use of fixed passphrases during enrollment and test allows for higher precision in speaker modeling, and therefore potentially lower verification error rates. The drawback is a potential risk that an ASV system can be defeated using plain recordings of clients' passphrases. We also suspect users may trust a fixed-phrase system less because they would at least partly associate security with the secrecy of their passphrase, and would not like to speak it in public. With randomized prompts, it is easier to convince users that the important thing is *how* the passphrase is spoken, not *what* it contains, and that recordings in the hands of others is not an issue. In the end, this suspected trust problem can perhaps be solved with proper priming material and marketing of the application.

Can ASV system performance be expected to improve in the near future, say within the next five years? In our view, marginal improvements in text-dependent ASV error rate can be expected, but not radical ones. During recent years, much of ASV research effort has been spent on text-independent verification techniques, and substantial improvements in text-independent ASV have also been observed during the annual NIST speaker recognition evaluations (Przybocki and Martin, 2004). It is possible, but not obvious, that at least some of the improvements can be re-applied successfully in the text-dependent domain. We would also like to point out Support Vector Machines (SVM) as a relatively new and promising classifier technique. SVMs and other new techniques may well be used in combination with current state-of-the-art techniques to improve performance.

Looking at the distribution of FRR for individual target speakers (at the corresponding global EER threshold), we found that only a few speakers show FRR above the average, while most speakers fall below the average. That is, as has been well known since previous work by others (e.g. Doddington et al., 1998), errors are unequally distributed among users of an ASV system, with a small fraction of users contributing to a large fraction of the errors. However, we also found that our empirical distributions of FRR and FAR over target speakers fit well to the family of beta distributions. A consequence of this uneven distribution of errors over users may be that those who experience high error rates stop using the associated application (or use alternative means of identity verification if such are offered), and only those who experience low error rates remain. For applications where such drop-out is acceptable, the result would be reduced overall error rates among users of the application. We have seen examples of this with PER.

Let us summarize the discussion so far. We have results from the PER corpus, 2.4% EER for the on-site version and 3.5% for landline telephones. Our subjects were biased towards lower error rate, suggesting error rates with a general population to be *higher*. We claimed our research system is near state-of-the-art, but use of wide-band signal processing and an increased length of passphrases should *reduce* the on-site error rate; and a good incremental decision strategy may be used to *reduce* error rates at only a small increase in the average length of test data. We expected state-of-the-art text-dependent ASV techniques to improve over the near future, but not radically so. This would by definition be accompanied by an additional *reduction* in error rate. We expected a transition to using fixed passphrases instead of randomized digit strings to buy another *reduction* in error rate, perhaps at the cost of decreased resistance against recordings and lower user trust (or increased need for user "education").

We thus have one argument as to why state-of-the-art ASV error rate in a commercial application in the near future would be *higher* than those we found with the PER systems, and three arguments for *lower* error rates, with an additional fourth argument for a *lower* error rate in the on-site version. Without indulging ourselves in trying to quantify each of the expected changes in error rate, we believe the expected error rate increase from a non-biased group of users may be larger than each of the individual expected reductions. It is our feeling that on-site overall EER can reach below 2% but not below 1%, and telephone error rates can reach below 3% but not below 1.5%.

11.2.2 Statistical significance

Pre-trial confidence intervals for observed error rates given a hypothesized "true" error rate can be useful in determining the required sample size for an experiment, for example when planning a corpus collection or a live service trial. Given a hypothesis about the error rate of an ASV system under scrutiny, and a requirement for statistical significance in testing the hypothesis, the number of required target speakers and impostor subjects can be approximately calculated (Higgins et al., 1991; Dialogues Spotlight Consortium, 2000; Dass and Jain, 2005). Counting only test trials from distinct targets as independent, by setting the intra-speaker correlation coefficient ρ in Eq. (2.20)[1] equal to 1, such an estimate of the number of required speakers will usually be very high. In practice, funding and the availability of subjects and resources for data collection may well limit the number of samples that can be collected, in particular the number of speakers. On the other extreme, assuming multiple trials from each speaker as independent ($\rho = 0$) results in gross under-estimation of the required number of speakers. Considering that in the case of speaker verification we want access to multiple sessions from each target to capture intra-speaker variability, and that if the number of available subjects is limited, it would be beneficial if it were possible to determine a more advantageous value of ρ that reflects the actual statistical dependency between data collected through multiple sessions from a set of target speakers.

For our experiments on the PER corpus, we computed post-trial confidence intervals for false reject rates based on the distribution of individual error rate. While these intervals are only estimates of statistical uncertainty, at least they are based on observations and not only guesses. Counting "backwards" they correspond to ρ-values of 0.06–0.22 for the largest test set (T2b_G8; 54 target speakers and an adjusted[2] average number of trials per target of 50) and 0.06–0.47 for the smaller test sets (27 targets, 36–40 trials per target on average). The average ρ over all cases was 0.2 (without having proved that an average makes sense in this case). We hypothesize that as a simple method for predicting pre-trial confidence intervals for overall false reject rates (in experiments not yet having been carried out) on the PER corpus, the choice of a fixed $\rho = 0.2$ will give reasonable results, while $\rho = 0.3$ will be on the "safe side". At the least, using any of the two values should result in better estimates of uncertainly than using $\rho = 0$ or $\rho = 1$. However, these choices of ρ are based on observations on the existing corpus.

We set out by stating that pre-trial confidence intervals can be useful in determining the required sample size for an experiment, for example when planning a corpus collection or a live service trial. To allow this, we need to predict confidence intervals for new, unseen data. Then, the question is: Do the ρ-values observed for the PER corpus carry over to other corpora? To answer this, further investigation is required. We think that they do to some degree, especially if the properties of the corpora are similar.

11.3 Future work

We think the estimation of uncertainty in ASV results need more attention. While the basic method for estimating confidence intervals (CI) based on the binomial distribution can be applied to results on corpora with a large number of speakers, such as corpora used in NIST speaker recognition evaluations, many ASV experiments are performed on corpora with a smaller number of speakers (say 50-100)

[1] p. 31

[2] cf. Section 6.3.4.6, p. 116

but many trials from each speaker. Such corpora are important in the ASV field, in particular for research on text-dependent techniques in “small” languages like Swedish: it is less costly in terms of subject recruitment and instruction overhead to collect a smaller number of speakers many times than a large number of speakers a few times; multiple trials from each speaker are required anyway to capture intra-speaker variability; and existing large English corpora may not be an alternative, in particular not the text-independent ones. One problem with such corpora is that estimation of uncertainty is more difficult because of dependencies between trials from the same speaker. Improved theoretical models are needed, and should be evaluated on real ASV data together with data-driven methods for CI estimation.

The PER and Gandalf corpora were designed to incorporate more factors than were eventually investigated in this thesis. Some of the remaining factors should be studied. Side-information about speakers and sessions in the Gandalf corpus should be used for a more detailed analysis of ASV results on this corpus (Section 6.2.3.2). Use of wide-band recordings in PER would give some insight into what additional speaker-dependent information is available in the 4–8 kHz band, and a better indication of what performance can be expected from ASV in on-site applications. The Swedish Speecon corpus (Iskra et al., 2002) could be used to support such experiments with training of wide-band background models. Cross-condition experiments on PER data would be a challenging task. We believe good performance on such a task, with for example enrollment by telephone and access on-site, is an important prerequisite to enabling wide-spread use of ASV in the future, under the motto “enroll once - access anywhere”.

In terms of technical advances in ASV in commercial-type applications, we believe improvements in the near future will be seen from improved support vector machine classifiers and fusion of multiple systems. Much would be gained in terms of overall error rate from improving ASV performance for the small fraction of the user population (“goats”) that contribute to the majority of errors. While the use of multiple systems and data fusion is a “brute force” approach that does not provide a better understanding of the real recognition problem, it has a potential for improving results. While not given much attention in this thesis, good incremental decision strategies and methods for robust selection of operating points for an ASV system will also be important for the potential future success of automatic speaker verification in the commercial market.

References

Altincay, H. and Demirekler, M. (1999). On the use of supra model information from multiple classifiers for robust speaker identification. In *Proc. 6th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 971–974, Budapest, Hungary.

Altincay, H. and Demirekler, M. (2002). Post-processing of classifier outputs in multiple classifier systems. In Roli, F. and Kittler, J., editors, *Proc. Third International Workshop Multiple Classifier Systems (MCS)*, volume 2364, pages 159–168, Cagliari, Italy. Springer.

Ambikairajah, E. and Hassel, P. (1996). Speaker identification on the Polycost database using line spectral pairs. In *Proc. COST 250 Workshop on Applications of Speaker Recognition Techniques in Telephony*, pages 47–54, Vigo, Spain.

Angelidis, J. (2003). Development of the Hörstöd application. Master's thesis, KTH/TMH, Stockholm, Sweden.

Anguita, J., Hernando, J., and Abad, A. (2005). Improved jacobian adaptation for robust speaker verification. *IEICE Transactions on Information and Systems*, E88-D(7):1767–1770.

Ariyaeeinia, A. and Sivakumaran, P. (1997). Analysis and comparison of score normalisation methods for text-dependent speaker verification. In *Proc. 5th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1379–1382, Rhodes, Greece.

Armerén, E. (1999). Site access controlled by speaker verification. Master's thesis, KTH/TMH, Stockholm, Sweden.

Ashbourn, J. and Savastano, M. (2002). Biometric sensors evaluation: User pshychology as a key factor in determining the success of an application. In *Proc. COST 275 Workshop - The Advent of Biometrics on the Internet*, pages 115–118, Rome, Italy.

Ashour, G. and Gath, I. (1999). Characterization of speech during imitation. In *Proc. 6th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1187–1190, Budapest, Hungary.

Atal, B. (1972). Automatic speaker recognition based on pitch contours. *The Journal of The Acoustical Society of America*, 52(6):1687–1697.

Atal, B. (1974). Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *The Journal of The Acoustical Society of America*, 55(6):1304–1312.

Atal, B. (1976). Automatic recognition of speakers from their voices. *Proceedings of the IEEE*, 64(4):460–475.

Auckenthaler, R., Carey, M., and Lloyd-Thomas, H. (2000). Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10:42–54.

Barras, C. and Gauvain, J.-L. (2003). Feature and score normalization for speaker verification of cellular data. In *Proc. 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 49–52, Hong Kong.

Barras, C., Meignier, S., and Gauvain, J.-L. (2004). Unsupervised online adaptation for speaker verification over the telephone. In *Proc. Odyssey04 - The Speaker and Language Recognition Workshop*, pages 157–160, Toledo, Spain.

Bayer, S., Doran, C., and George, B. (2001). Dialogue interaction with the DARPA Communicator infrastructure: the development of useful software. In *Notebook proceedings, First International Conference on Human Language Technology Research*, pages 179–181, San Diego CA, USA.

Ben, M., Betser, M., Bimbot, F., and Gravier, G. (2004). Speaker diarization using bottom-up clustering based on a parameter-derived distance between adapted GMMs. In *Proc. 2004 International Conference on Spoken Language Processing (ICSLP)*, pages 2329–2332, Jeju Island, Korea.

Ben, M., Blouet, R., and Bimbot, F. (2002). A monte-carlo method for score normalization in automatic speaker verification using kullback-leibler distances. In *Proc. 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 689–692, Orlando FL, USA.

Ben-Yacoub, S., Abdeljaoued, Y., and Mayoraz, E. (1999). Fusion of face and speech data for person identity verification. *IEEE Transactions on Neural Networks*, 10(5):1065–1074.

Bengio, S. and Mariéthoz, J. (2004a). The expected performance curve: a new assessment measure for person authentication. In *Proc. Odyssey04 - The Speaker and Language Recognition Workshop*, pages 279–284, Toledo, Spain.

Bengio, S. and Mariéthoz, J. (2004b). A statistical significance test for person authentication. In *Proc. Odyssey04 - The Speaker and Language Recognition Workshop*, pages 237–244, Toledo, Spain.

BenZeghiba, M. and Bourlard, H. (2002). User-customized password speaker verification based on HMM/ANN and GMM models. In *Proc. 2002 International Conference on Spoken Language Processing (ICSLP)*, pages 1325–1328, Denver CO, USA.

Bernasconi, C. (1990). On instantaneous and transitional spectral information for text-dependent speaker verification. *Speech Communication*, 9(2):129–139.

Beskow, J. (1995). Rule-based visual speech synthesis. In *Proc. 4th European Conference on Speech Communication and Technology (EUROSPEECH)*, volume 2, pages 299–302, Madrid, Spain.

Bickley, C. and Hunnicutt, S. (2000). ENABL - enabler for engineering software using language and speech. *TMH-QPSR*, 41(1):1–11.

Bimbot, F., Blomberg, M., Boves, L., Chollet, G., Jaboulet, C., Jacob, B., Kharroubi, J., Koolwaaij, J., Lindberg, J., Mariethoz, J., Mokbel, C., and Mokbel, H. (1999). An overview of the PICASSO project research activities in speaker verification for telephone applications. In *Proc. 6th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1963–1966, Budapest, Hungary.

Bimbot, F., Blomberg, M., Boves, L., Genoud, D., Hutter, H.-P., Jaboulet, C., Koolwaaij, J., Lindberg, J., and Pierrot, J.-B. (2000). An overview of the CAVE project research activities in speaker verification. *Speech Communication*, 31(2-3):155–180.

Bimbot, F. and Chollet, G. (1997). Assessment of speaker verification systems. In Gibbon, D., Moore, R., and R., W., editors, *Handbook of Standards and Resources for Spoken Language Systems*. Mouton de Gruyter.

Bimbot, F., Chollet, G., and Paoloni, A. (1994). Assessment methodology for speaker identification and verification systems. In *Proc. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 75–82, Martigny, Switzerland.

Bimbot, F. and Mathan, L. (1994). Second-order statistical measures for text-independent speaker identification. In *Proc. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 51–54, Martigny, Switzerland.

Bjork, E. and Healy, A. (1974). Short-term order and item retention. *Journal of Verbal Learning and Verbal Behaviour*, 13:80–97.

Bjurling, M. (2004). Röststyrt gränssnitt för omgivningskontroll; test, analys och konstruktion av ett trådlöst system (in Swedish). Master's thesis, KTH/TMH, Stockholm, Sweden.

Blouet, R., Mokbel, C., Mokbel, H., Soto, E., Chollet, G., and Greige, H. (2004). BECARS: a free software for speaker verification. In *Proc. Odyssey04 - The Speaker and Language Recognition Workshop*, pages 145–148, Toledo, Spain.

Bolle, R., Ratha, N., and Pankanti, S. (2004). An evaluation of error confidence interval estimation methods. In *Proc. 17th International Conference on Pattern Recognition (ICPR)*, volume 3, pages 103–106, Cambridge, UK.

Bonastre, J.-F., Bimbot, F., Boë, L.-J., Campbell, J., Reynolds, D., and Magrin-Chagnolleau, I. (2003a). Person authentication by voice: A need for caution. In *Proc. 8th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 33–36, Geneva, Switzerland.

Bonastre, J.-F., Morin, P., and Junqua, J.-C. (2003b). Gaussian dynamic warping (GDW) method applied to text-dependent speaker detection and verification. In *Proc. 8th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 2013–2016, Geneva, Switzerland.

Bonastre, J.-F., Wils, F., and Meignier, S. (2005). ALIZE, a free toolkit for speaker recognition. In *Proc. 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 737–740, Philadelphia PA, USA.

Bonastre, L.-F., Meloni, H., and Langlais, P. (1991). Analytical strategy for speaker identification. In *Proc. 2nd European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 435–438, Genova, Italy.

Bonifas, J.-L., Hernaez, R. I., Etxebarria, G. B., and Saoudi, S. (1995). Text-dependent speaker verification using dynamic time warping and vector quantization of LSF. In *Proc. 4th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 359–362, Madrid, Spain.

Boves, L., Bogaart, T., and Bos, L. (1994). Design and recording of large data bases for use in speaker verification and identification. In *Proc. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 43–46, Martigny, Switzerland.

Broun, C., Campbell, W., Pearce, D., and Kelleher, H. (2001). Speaker recognition and the ETSI standard distributed speech recognition front-end. In *Proc. Speaker Odyssey - The Speaker Recognition Workshop*, pages 121–124, Crete, Greece.

Burges, C. (1998). A tutorial on support vector machines for patterns recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

Burton, D. (1987). Text-dependent speaker verification using vector quantization source coding. *IEEE Transactions on Speech and Audio Processing*, 35(2):133–143.

Caminero, J., Gonzalez-Rodriguez, J., Ortega-Garcia, J., Tapias, D., Ruz, P., and Sola, M. (2002). A multilingual speaker verification system: Architecture and performance evaluation. In *Proc.International Conference on Language Resources and Evaluation (LREC)*, pages 626–631, Las Palmas, Spain.

Campbell, J. (1997). Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85(9):1437–1462.

Campbell, J. and Reynolds, D. (1999). Corpora for the evaluation of speaker recognition systems. In *Proc. 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 829–832, Phoenix AZ, USA.

Campbell, Jr., J. (1995). Testing with the YOHO CD-ROM voice verification corpus. In *Proc. 1995 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 341–344, Detroit MI, USA.

Campbell, W. (2002). Generalized linear discriminant sequence kernels for speaker recognition. In *Proc. 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 161–164, Orlando FL, USA.

Campbell, W., Assaleh, K., and Broun, C. (2002). Speaker recognition with polynomial classifiers. *IEEE Transactions on Speech and Audio Processing*, 10(4):205–212.

Campbell, W., Campbell, J., Reynolds, D., Jones, D., and Leek, T. (2004a). High-level speaker verification with support vector machines. In *Proc. 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 73–76, Montreal, Canada.

Campbell, W., Reynolds, D., and Campbell, J. (2004b). Fusing discriminative and generative methods for speaker recognition: Experiments on switchboard and nfi/tno field data. In *Proc. Odyssey04 - The Speaker and Language Recognition Workshop*, pages 41–44, Toledo, Spain.

Carey, M., Parris, E., Bennett, S., and Lloyd-Thomas, H. (1997). A comparison of model estimation techniques for speaker verification. In *Proc. 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1083–1086, Munich, Germany.

Carey, M., Parris, E., and Bridle, J. (1991). A speaker verification system using alpha-nets. In *Proc. 1991 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 397–400, Toronto, Canada.

Carey, M., Parris, E., Lloyd-Thomas, H., and Bennett, S. (1996). Robust prosodic features for speaker identification. In *Proc. 1996 International Conference on Spoken Language Processing (ICSLP)*, pages 1800–1803, Philadelphia PA, USA.

Carlson, R., Granström, B., and Hunnicutt, S. (1982). A multi-language text-to-speech module. In *Proc. 1982 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages 1604–1607, Paris, France.

Carlson, R., Granström, B., and Karlsson, I. (1991). Experiments with voice modeling in speech synthesis. *Speech Communication*, 10(5–6):481–489.

Chao, Y.-H., Wang, H.-M., and Chang, R.-C. (2005). GMM-based Bhattacharyya kernel Fisher discriminant analysis for speaker recognition. In *Proc. 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 649–652, Philadelphia PA, USA.

Chaudhari, U., Navrátil, J., and Maes, S. (2003). Multigrained modeling with pattern specific maximum likelihood transformations for text-independent speaker recognition. *IEEE Transactions on Speech and Audio Processing*, 11(1):61–69.

Chen, S. and Gopinath, R. (2001). Gaussianization. In Leen, T., Dietterich, T., and Tresp, V., editors, *Advances in Neural Information Processing Systems, Papers from Neural Information Processing Systems (NIPS) 2000*, volume 13, pages 423–429, Denver CO, USA. The MIT Press.

Chetouani, M., Faundez-Zanuy, M., Gas, B., and Zarader, J. (2004). A new non-linear feature extraction algorithm for speaker verification. In *Proc. 2004 International Conference on Spoken Language Processing (ICSLP)*, pages 1405–1408, Jeju Island, Korea.

Cohen, A. and Vaich, T. (1994). On the identification of twins by their voices. In *Proc. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 213–216, Martigny, Switzerland.

Cole, R., Noel, M., and Noel, V. (1998). The CSLU speaker recognition corpus. In *Proc. 1998 International Conference on Spoken Language Processing (ICSLP)*, pages 3167–3170, Sydney, Australia.

Colombi, J., Anderson, T., Rogers, S., Ruck, D., and Warhola, G. (1993). Auditory model representation for speaker recognition. In *Proc. 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 700–703, Minneapolis MN, USA.

Darsinos, V., Galanis, D., and Kokkinakis, G. (1995). A method for fully automatic analysis and modelling of voice source characteristics. In *Proc. 4th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 413–416, Madrid, Spain.

Dass, S. and Jain, A. (2005). Effects of user correlation on sample size requirements. In *Proc. SPIE Defense and Security Symposium (DSS)*, pages 226–231, Orlando FL, USA.

Dau, T., Kollmeier, B., and Kohlrausch, A. (1997). Modeling auditory processing of amplitude modulation (parts i and ii). *The Journal of The Acoustical Society of America*, 102(5):2892–2919.

Dau, T., Puschel, D., and Kohlrausch, A. (1996). A quantitative model of the "effective" signal processing in the auditory system (part i). *The Journal of The Acoustical Society of America*, 99(6):3615–3622.

Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monsyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366.

Dialogues Spotlight Consortium (2000). Large scale evaluation of automatic speaker verification technology. Technical report, The Centre for Communication Interface Research, The University of Edinburgh.

Dialogues Spotlight Consortium (2001). Evaluation of Nuance v7.0.4 speaker verification performance on the Dialogues Spotlight UK English database. Technical report, The Centre for Communication Interface Research, The University of Edinburgh.

Doddington, G. (1985). Speaker recognition - identifying people by their voices. *Proceedings of the IEEE*, 73(11):1651–1664.

Doddington, G. (1998). Speaker recognition evaluation methodology: An overview and perspective. In *Proc. Workshop on Speaker Recognition and its Commercial and Forensic Applications (RLA2C)*, pages 60–66, Avignon, France.

Doddington, G., Liggett, W., Martin, A., Przybocki, M., and Reynolds, D. (1998). SHEEP, GOATS, LAMBS and WOLWES - a statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation. In *Proc. 1998 International Conference on Spoken Language Processing (ICSLP)*, pages 1351–1354, Sydney, Australia.

Durou, G. and Jauquet, F. (1998). Cross-language text-independent speaker identification. In *Proc.Europeach Signal Processing Conference (EUSIPCO)*, Rhodes, Greece.

Dutoit, T., Pagel, V., Pierret, N., Batialle, F., and van der Vreken, O. (1996). The MBROLA project: towards a set of high-quality speech synthesizers free of use for non-commercial purposes. In *Proc. 1996 International Conference on Spoken Language Processing (ICSLP)*, pages 1393–1396, Philadelphia PA, USA.

Edlund, J. and Hjalmarsson, A. (2005). Applications of distributed dialogue systems: the KTH Connector. In *Proc. ISCA Tutorial and Research Workshop on Applied Spoken Language Interaction in Distributed Environments (ASIDE 2005)*, Aalborg, Denmark.

Edlund, J., Skantze, G., and Carlson, R. (2004). Higgins – a spoken dialogue system for investigating error handling techniques. In *Proc. 2004 International Conference on Spoken Language Processing (ICSLP)*, pages 229–231, Jeju Island, Korea.

Ejarque, P. and Hernando, J. (2005). Variance reduction by using separate genuine-impostor statistics in multimodal biometrics. In *Proc. 9th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 785–788, Lisbon, Portugal.

Elenius, D. (2001). Härmning - ett hot mot talarverifieringssystem? (in Swedish). Master's thesis, KTH/TMH, Stockholm, Sweden.

Elenius, D. and Blomberg, M. (2002). Characteristics of a low reject mode speaker verification system. In *Proc. 2002 International Conference on Spoken Language Processing (ICSLP)*, pages 1385–1388, Denver CO, USA.

Elenius, K. (2000). Experiences from collecting two Swedish telephone speech databases. *International Journal of Speech Technology*, 3:119–127.

Fant, G. (1960). *Acoustic Theory of Speech Production*. Mouton, The Hague.

Fant, G. (1986). Glottal flow: models and interaction. *Journal of Phonetics*, 14:393–399.

Farrell, K., Mammone, R., and Assaleh, K. (1994). Speaker recognition using neural networks and conventional classifiers. *IEEE Transactions on Speech and Audio Processing*, 2(1):194–205.

Fatma, K. and Cetin, E. (1999). Design of gaussian mixture models using matching persuit. In *Proc. IEEE-EURASIP Workshop on Non-Linear Signal and Image Processing (NSIP)*, Antalya, Turkey.

Feix, W. and DeGeorge, M. (1985). A speaker verification system for access-control. In *Proc. 1985 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 399–402, Tampa, Florida.

Feller, W. (1968). *An introduction to probability theory and its applications*, volume 1. Wiley, New York, 3 edition.

Fette, B., Broun, C., Campbell, W., and Jaskie, C. (2000). CipherVOX: Scalable low-complexity speaker verification. In *Proc. 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3634–3637, Istanbul, Turkey.

Filipsson, M. and Bruce, G. (1997). LUKAS - a preliminary report on a new Swedish speech synthesis. In *Working Papers*, volume 46, pages 47–56, Lund University, Dept. of Linguistics, Lund, Sweden.

Fine, S., Navrátil, J., and Gopinath, R. (2001). Enhancing GMM scores using SVM "hint". In *Proc. 7th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1757–1760, Aalborg, Denmark.

Fredouille, C., Mariéthoz, J., Jaboulet, C., Hennebert, J., Bonastre, J.-F., Mokbel, C., and Bimbot, F. (2000). Behaviour of a bayesian adaptation method for incremental enrollment in speaker verification. In *Proc. 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1197–1200, Istanbul, Turkey.

Fujisaki, H. and Ljungqvist, M. (1986). Proposal and evaluation of models for the glottal source waveform. In *Proc. 1986 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1605–1608, Tokyo, Japan.

Furui, S. (1981). Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(2):254–272.

Furui, S. (1986). Research on individuality features in speech waves and automatic speaker recognition techniques. *Speech Communication*, 5(2):183–197.

Furui, S. (1997). Recent advances in speaker recognition. In *Proc. Audio- and Video-Based Biometric Person Authentication (AVBPA)*, pages 237–252, Crans-Montana, Switzerland.

Gagnon, L., Stubley, P., and Mailhot, G. (2001). Password-dependent speaker verification using quantized acoustic trajectories. In *Proc. 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Salt Lake City UT, USA.

Ganchev, T., Fakotakis, N., Tasoulis, D., and Vrahatis, M. (2004a). Generalized locally recurrent probabilistic neural networks for text-independent speaker verification. In *Proc. 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 41–44, Montreal, Canada.

Ganchev, T., Tasoulis, D., Vrahatis, M., and Fakotakis, N. (2004b). Locally recurrent probabilistic neural networks with application to speaker verification. *GESTS International Transactions on Speech Science and Engineering*, 1(2):1–13.

Ganchev, T., Tsopanoglou, A., Faktoakis, N., and Kokkinakis, G. (2002). Probabilistic neural networks combined with GMMs for speaker recognition over telephone channels. In *Proc. International Conference on Digital Signal Processing (DSP)*, pages 1081–1084, Santorini, Greece.

Garcia Torcelly, J. (2002). Speaker verification based on speech encoder parameters. Master's thesis, KTH/TMH, Stockholm, Sweden.

Gauvain, J.-L., Lamel, L., and Prouts, B. (1995). Experiments with speaker verification over the telephone. In *Proc. 4th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 651–654, Madrid, Spain.

Gauvain, J.-L. and Lee, C.-H. (1992). Bayesian learning for hidden markov model with gaussian mixture state observation densities. *Speech Communication*, 11(2-3):205–213.

Genoud, D. and Chollet, G. (1999). Deliberate imposture: A challenge for automatic speaker verification systems. In *Proc. 6th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1971–1974, Budapest, Hungary.

Gersho, A. and Gray, R., editors (1992). *Vector Quantization and Signal Compression*. Springer.

Gravier, G., Bonastre, J.-F., Geoffrois, E., Galliano, S., Mc Tait, K., and Choukri, K. (2004). The ESTER evaluation compaign for the rich transcription of French broadcast news. In *Proc.International Conference on Language Resources and Evaluation (LREC)*, pages 885–888, Lisbon, Portugal.

Gu, Y., Jongebloed, H., Iskra, D., Os, E., and Boves, L. (2000). Speaker verification in operational environments – monitoring for improved service operation. In *Proc. 2000 International Conference on Spoken Language Processing (ICSLP)*, pages 450–453, Beijing, China.

Gu, Y. and Thomas, T. (2001). A text-independent speaker verification system using support vector machines classifier. In *Proc. 7th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1765–1768, Aalborg, Denmark.

Gupta, S. and Savic, M. (1992). Text-independent speaker verification based on broad phonetic segmentation of speech. *Digital Signal Processing*, 2:69–79.

Gustafson, J., Bell, L., Beskow, J., Boye, J., Carlson, R., Edlund, J., Granström, B., House, D., and Wirén, M. (2000). AdApt - a multimodal conversational dialogue system in an apartment domain. In *Proc. 2000 International Conference on Spoken Language Processing (ICSLP)*, volume 2, pages 134–137, Beijing, China.

Gustafson, J., Lindberg, N., and Lundeberg, M. (1999). The august spoken dialogue system. In *Proc. 6th European Conference on Speech Communication and Technology (EUROSPEECH)*, volume 3, pages 1151–1154, Budapest, Hungary.

Gustafsson, Y. (2000). Hidden markov models with applications in speaker verification. Master's thesis, KTH/Mathematical Statistics, Stockholm, Sweden.

Gyllensvärd, P. (2003). Röststyrning på bilbesiktningen? (in Swedish). Master's thesis, KTH/TMH, Stockholm, Sweden.

Hennebert, J., Melin, H., Petrovska, D., and Genoud, D. (2000). POLYCOST: A telephone-speech database for speaker recognition. *Speech Communication*, 31(2-3):265–270.

Hermansky, H. (1990). Perceptual linear predictive (plp) analysis of speech. *The Journal of The Acoustical Society of America*, 87(4):1738–1752.

Hermansky, H., Morgan, N., Bayya, A., and Kohn, P. (1991). Compensation for the effect of the communication channel in auditory-like analysis of speech (RASTA-PLP). In *Proc. 2nd European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1367–1370, Genova, Italy.

Hernando, J. and Nadeu, C. (1998). Speaker verification on the Polycost database using frequency filtered spectral energies. In *Proc. 1998 International Conference on Spoken Language Processing (ICSLP)*, pages 129–132, Sydney, Australia.

Higgins, A. and Bahler, L. (2001). Password-based voice verification using SpeakerKey. In *Proc. Speaker Odyssey - The Speaker Recognition Workshop*, pages 31–32, Crete, Greece.

Higgins, A., Bahler, L., and Porter, J. (1991). Speaker verification using randomized phrase prompting. *Digital Signal Processing*, 1:89–106.

Higgins, A., Bahler, L., and Porter, J. (1993). Voice identification using nearest-neighbor distance measure. In *Proc. 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 375–378, Minneapolis MN, USA.

Homayounpour, M. and Chollet, G. (1995). Discrimination of voices of twins and siblings for speaker verification. In *Proc. 4th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 345–348, Madrid, Spain.

Huang, X., Acero, A., and Hon, H.-W. (2001). *Spoken Language Processing - A Guide to Theory, Algorithm, and System Development.* Prentice Hall PTR.

Hébert, M. and Boies, D. (2005). T-norm for text-dependent commercial speaker verification applications: Effect of lexical mismatch. In *Proc. 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 729–732, Philadelphia PA, USA.

Hébert, M. and Peters, D. (2000). Improved normalization without recourse to an impostor database for speaker verification. In *Proc. 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1213–1216, Istanbul, Turkey.

Höge, H., Tropf, H., Winski, R., van den Heuvel, H., Haeb-Umbach, R., and Choukri, K. (1997). European speech databases for telephone applications. In *Proc. 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1771–1774, Munich, Germany.

Ihse, M. (2000). Usability study of a speech controlled telephone banking system. Master's thesis, KTH/TMH, Stockholm, Sweden.

Iskra, D., Grosskopf, B., Marasek, K., van den Huevel, H., Diehl, F., and Kiessling, A. (2002). Speecon – speech databases for consumer devices: Database specification and validation. In *Proc.International Conference on Language Resources and Evaluation (LREC)*, Las Palmas, Spain.

Isobe, T. and Takahashi, J. (1999). A new cohort normalization using local acoustic information for speaker verification. In *Proc. 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 841–845, Phoenix AZ, USA.

Jain, A., Bolle, R., and Pankanti, S., editors (1999). *Biometrics: Personal Identification in Networked Society.* Kluwer Academic, Dordrecht, The Netherlands.

Johansson, M. (2002). Phoneme recognition as a hearing aid in telephone communication. Master's thesis, Dept. of Linguistics, Uppsala University, Uppsala, Sweden.

Johansson, M., Blomberg, M., Elenius, K., Hoffsten, L.-E., and Torberger, A. (2002). Phoneme recognition for the hearing impaired. *TMH-QPSR*, 44:109–112.

Karlsson, I., Banziger, T., Dankovicová, J., Johnstone, T., Lindberg, J., Melin, H., Nolan, F., and Scherer, K. (1998). Within-speaker variability due to speaking manners. In *Proc. 1998 International Conference on Spoken Language Processing (ICSLP)*, pages 2379–2382, Sydney, Australia.

Karlsson, I., Banziger, T., Dankovicová, J., Johnstone, T., Lindberg, J., Melin, H., Nolan, F., and Scherer, K. (2000). Speaker verification with elicited speaking styles in the VeriVox project. *Speech Communication*, 31(2-3):121–129.

Katz, M., Krüger, S., Schafföner, M., Andelic, E., and Wendemuth, A. (2006a). Speaker identification and verification using support vector machines and spares kernel logistic regression. In *Proc. The International Workshop on Intelligent Computing in Pattern Analysis/Synthesis*, Xian, China.

Katz, M., Schafföner, M., Andelic, E., Krüger, S., and Wendemuth, A. (2006b). Sparse kernel logistic regression using incremental feature selection for text-independent speaker identification. In *Proc. IEEE Odyssey 2006: The Speaker and Language Recognition Workshop*, San Juan, Puerto Rico.

Kharroubi, J., Petrovska-Delacrétaz, D., and Chollet, G. (2001). Text-independent speaker verification using support vector machines. In *Proc. Speaker Odyssey - The Speaker Recognition Workshop*, pages 51–54, Crete, Greece.

Konig, Y., Heck, L., Weintraub, M., and Sonmez, K. (1998). Nonlinear discriminant feature extraction for robust text-independent speaker recognition. In *Proc. Workshop on Speaker Recognition and its Commercial and Forensic Applications (RLA2C)*, pages 72–75, Avignon, France.

Koolwaaij, J. and Boves, L. (2000). Local normalization and delayed decision making in speaker detection and tracking. *Digital Signal Processing*, 10:113–132.

Koolwaaij, J., Boves, L., Os, E., and Jongebloed, H. (2000). On model quality and evaluation in speaker verification. In *Proc. 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3759–3762, Istanbul, Turkey.

Lamel, L. (2005). Personal communication.

Lamel, L. and Gauvain, J.-L. (1998). Speaker verification over the telephone. In *Proc. Workshop on Speaker Recognition and its Commercial and Forensic Applications (RLA2C)*, pages 76–79, Avignon, France.

LcLaughlin, J., Pitton, J., and Kirchhoff, K. (2002). University of Washington 2002 NIST speaker ID evaluations. In *NIST Speaker Recognition Workshop, Informal proceedings*, Vienna VA, USA.

Lee, C.-H. and Gauvain, J.-L. (1993). Speaker adaptation based on MAP estimation of HMM parameters. In *Proc. 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 558–561, Minneapolis MN, USA.

Lee, C.-H. and Gauvain, J.-L. (1996). Bayesian adaptive learning and MAP estimation of HMM. In Lee, C.-H., Soong, F., and Paliwal, K., editors, *Automatic Speech and Speaker Recognition: Advanced Topics*, pages 83–107. Kluwer.

Lee, P. (1989). *Bayesian Statistics: An Introduction.* Oxford University Press, New York.

Leung, K.-Y., Mak, M.-W., and Kung, S.-Y. (2004). Applying articulatory features to telephone-based speaker verification. In *Proc. 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 85–88, Montreal, Canada.

Leung, K.-Y., Mak, M.-W., Siu, M., and Kung, S.-Y. (2005). Speaker verification using adapted articulatory feature-based conditional pronunciation modeling. In *Proc. 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 181–184, Philadelphia PA, USA.

Li, H., Haton, J.-P., and Gong, Y. (1995). On MMI learning of gaussian mixture for speaker models. In *Proc. 4th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 363–366, Madrid, Spain.

Li, Q. (2004). Discovering relations among discriminative training objectives. In *Proc. 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 33–36, Montreal, Canada.

Li, Q. and Juang, B.-H. (2003). Fast discriminative training for sequential observations with application to speaker identification. In *Proc. 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Hong Kong.

Li, Q., Juang, B.-W., Zhou, Q., and Lee, C.-H. (2000). Automatic verbal information verification for user authentication. *IEEE Transactions on Speech and Audio Processing*, 8(5):585–596.

Li, X., Chang, E., and Dai, B. (2002). Improving speaker verification with figure of merit training. In *Proc. 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 693–696, Orlando FL, USA.

Lindberg, B., Johansen, F., Warakagoda, N., Lehtinen, G., Kacic, Z., Zgank, A., Elenius, K., and Salvi, G. (2000). A noise robust multilingual reference recogniser based on SpeechDat(II). In *Proc. 2000 International Conference on Spoken Language Processing (ICSLP)*, pages 370–373, Beijing, China.

Lindberg, J. and Blomberg, M. (1999). Vulnerability in speaker verification - a study of technical impostor techniques. In *Proc. 6th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1211–1214, Budapest, Hungary.

Lindberg, J. and Melin, H. (1997). Text-prompted versus sound-prompted passwords in speaker verification systems. In *Proc. 5th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 851–854, Rhodes, Greece.

Lindblom, T. (2003). Text independent speaker verification using a hybrid of support vector machines and gaussian mixture models. Master's thesis, KTH/TMH, Stockholm, Sweden.

Linville, S. (2001). *Vocal Aging.* Singular Thomson Learning, San Diego.

Liu, C., Tin, M., Wang, W., and Wang, H. a. (1990). Study of line spectrum pair frequencies for speaker recognition. In *Proc. 1990 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 277–280, Albuquerque NM, USA.

Liu, C.-S., Lee, C.-H., Chou, W., Juang, B.-H., and Rosenberg, A. (1995). A study on minimum error discriminative training for speaker recognition. *The Journal of The Acoustical Society of America*, 97(1):637–648.

Luck, J. (1969). Automatic speaker verification using cepstral measurements. *The Journal of The Acoustical Society of America*, 46(4):1026–1032.

Lund, M. and Lee, C. (1996). A robust sequential test for text-independent speaker verification. *The Journal of The Acoustical Society of America*, 99(1):609–621.

Lundgren, S. (2003). Atlas talsyntesbrygga med (J)SAPI stöd; testad med Babel-Infovox BrightSpeech (in Swedish). Master's thesis, KTH/TMH, Stockholm, Sweden.

Magrin-Chagnolleau, I. and Durou, G. (1999). Time-frequency principal components of speech: application to speaker identification. In *Proc. 6th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 759–762, Budapest, Hungary.

Magrin-Chagnolleau, I. and Durou, G. (2000). Application of vector filtering to pattern recognition. In *Proc. 15th International Conference on Pattern Recognition (ICPR)*, volume 3, pages 433–436, Barcelona, Spain.

Magrin-Chagnolleau, I., Durou, G., and Bimbot, F. (2002). Application of time-frequency principal component analysis to text-independent speaker identification. *IEEE Transactions on Speech and Audio Processing*, 10(6):371–378.

Magrin-Chagnolleau, I., Gravier, G., Seck, M., Boeffard, O., Blouet, R., and Bimbot, F. (2000). A further investigation of speech features for speaker characterization. In *Proc. 2000 International Conference on Spoken Language Processing (ICSLP)*, volume 3, pages 1029–1032, Beijing, China.

Mansfield, A. and Wayman, J. (2002). Best practices in testing and reporting performance of biometric devices, version 2.01. Technical report, UK Biometric Working Group (BWG).

Mariani, J. (1992). Spoken language processing in the framework of human-machine communication at LIMSI. In *Proc. 5th DARPA Speech and Natural Language Workshop*, pages 55–60, Harriman NY, USA.

Mariethoz, J. and Bengio, S. (2005). A unified framework for score normalization techniques applied to text-independent speaker verification. *IEEE Signal Processing Letters*, 12(7):532–535.

Martin, A., Doddington, G., Kamm, T., Ordowski, M., and Przybocki, M. (1997). The DET curve in assessment of detection task performance. In *Proc. 5th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1895–1898, Rhodes, Greece.

Martin, A. and Przybocki, M. (2000). The NIST 1999 speaker recognition evaluation - an overview. *Digital Signal Processing*, 10:1–18.

Martin, A. and Przybocki, M. (2001). The NIST speaker recognition evaluations: 1996-2001. In *Proc. Speaker Odyssey - The Speaker Recognition Workshop*, pages 39–43, Crete, Greece.

Matrouf, D., Bonastre, J.-F., and Fredouille, C. (2006). Effect of speech transformation on impostor acceptance. In *Proc. 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 933–936, Toulouse, France.

Matsui, T. and Furui, S. (1990). Text-independent speaker recognition using vocal tract and pitch information. In *Proc. 1990 International Conference on Spoken Language Processing (ICSLP)*, pages 137–140, Kobe, Japan.

Matsui, T. and Furui, S. (1993). Concatenated phoneme models for text-variable speaker recognition. In *Proc. 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 391–394, Minneapolis MN, USA.

Matsui, T. and Furui, S. (1994). Similarity normalization method for speaker verification based on a posteriori probability. In *Proc. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 59–62, Martigny, Switzerland.

Melin, H. (1996). Gandalf - a Swedish telephone speaker verification database. In *Proc. 1996 International Conference on Spoken Language Processing (ICSLP)*, pages 1954–1957, Philadelphia PA, USA.

Melin, H. (1998). On word boundary detection in digit-based speaker verification. In *Proc. Workshop on Speaker Recognition and its Commercial and Forensic Applications (RLA2C)*, pages 46–49, Avignon, France.

Melin, H. (1999). Databases for speaker recognition: working group 2 final report. In Falcone, M., editor, *COST250 Speaker Recognition in Telephony, Final Report (CD-ROM)*. European Commission, DG XIII-B, Brussels.

Melin, H., Ariyaeeinia, A., and Falcone, M. (1999). The COST250 speaker recognition reference system. In Falcone, M., editor, *COST250 Speaker Recognition in Telephony, Final Report (CD-ROM)*. European Commission, DG XIII-B, Brussels.

Melin, H., Koolwaaij, J., Lindberg, J., and Bimbot, F. (1998). A comparative evaluation of variance flooring techniques in HMM-based speaker verification. In *Proc. 1998 International Conference on Spoken Language Processing (ICSLP)*, pages 1903–1906, Sydney, Australia.

Melin, H. and Lindberg, J. (1996). Guidelines for experiments on the Polycost database (version 1.0). In *Proc. COST 250 Workshop on Applications of Speaker Recognition Techniques in Telephony*, pages 59–69, Vigo, Spain.

Melin, H. and Lindberg, J. (1999a). Guidelines for experiments on the Polycost database (version 2.0). In Falcone, M., editor, *COST250 Speaker Recognition in Telephony, Final Report (CD-ROM)*. European Commission, DG XIII-B, Brussels.

Melin, H. and Lindberg, J. (1999b). Variance flooring, scaling and tying for text-dependent speaker verification. In *Proc. 6th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1975–1978, Budapest, Hungary.

Melin, H., Sandell, A., and Ihse, M. (2001). CTT-bank: A speech controlled telephone banking system - an initial evaluation. *TMH-QPSR*, 42(1):1–27.

Mengusoglu, E. (2003). Confidence measure based model adaptation for speaker verification. In *Proc. the IASTED International Conference Connunications, Internet, and Information Technology (CIIT)*, Scottsdale AZ, USA.

Michaels, R. and Boult, T. (2001). Efficient evaluation of classification and recognition systems. In *Proc.International Conference on Computer Vision and Pattern Recognition (CVPR)*, Kauai Marriott, Hawaii, USA.

Moreno, P. and Ho, P. (2003). A new SVM approach to speaker identification and verification using probabilistic distance kernels. In *Proc. 8th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 2965–2968, Geneva, Switzerland.

Moreno, P. and Ho, P. (2004). SVM kernel adaptation in speaker classification and verification. In *Proc. 2004 International Conference on Spoken Language Processing (ICSLP)*, pages 1413–1416, Jeju Island, Korea.

Morgan, N. and Hermansky, H. (1992). RASTA extentions: Robustness to additive and convolutional noise. In *Proc. ESCA Workshop on Speech Processing in Adverse Conditions*, pages 115–118, Cannes, France.

Morin, P. and Junqua, J.-C. (2003). A voice-centric multimodal user authentication system for fast and convenient physical access control. In *Proc. Workshop on Multimodal User Authentication*, pages 19–24, Santa Barbara CA, USA.

Murray, I. and Arnott, L. (1993). Toward the simulation of emotion in synthetic speech: A review of the literature on human vocal emotion. *The Journal of The Acoustical Society of America*, 93(2):1097–1108.

Naik, J. and Doddington, G. (1986). High performance speaker verification using principal spectral components. In *Proc. 1986 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 881–884, Tokyo, Japan.

Naik, J., Netsch, L., and Doddington, G. (1989). Speaker verification over long distance telephone lines. In *Proc. 1989 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 524–527, Glasgow, UK.

Navrátil, J., Chaudhari, U., and Ramaswamy, G. (2001). Speaker verification using target and background dependent linear transformations and multi-system fusion. In *Proc. 7th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1389–1392, Aalborg, Denmark.

Navrátil, J. and Ramaswamy, G. (2002). DETAC: A discriminative criterion for speaker verification. In *Proc. 2002 International Conference on Spoken Language Processing (ICSLP)*, pages 1349–1352, Denver CO, USA.

Navrátil, J. and Ramaswamy, G. (2003). The awe and mystery of t-norm. In *Proc. 8th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 2009–2012, Geneva, Switzerland.

Neiberg, D. (2001). Text independent speaker verification using adapted gaussian mixture models. Master's thesis, KTH/TMH, Stockholm, Sweden.

Netsch, L. and Doddington, G. (1992). Speaker verification using temporal decorrelation post-processing. In *Proc. 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 181–184, San Francisco CA, USA.

Newman, M., Gillick, L., Ito, Y., McAllister, D., and Peskin, B. (1996). Speaker verification through large vocabulary continuous speech recognition. In *Proc. 1996 International Conference on Spoken Language Processing (ICSLP)*, pages 2419–2422, Philadelphia PA, USA.

NIST, editor (1998). *NIST Speaker Recognition Workshop, Informal proceedings*, College Park MD, USA. NIST.

NIST, editor (2002). *NIST Speaker Recognition Workshop, Informal proceedings*, Vienna VA, USA. NIST.

Nordqvist, P. and Leijon, A. (2004). An efficient robust sound classification algorithm for hearing aids. *The Journal of The Acoustical Society of America*, 115(6):3033–3041.

Nordström, T., Melin, H., and Lindberg, J. (1998). A comparative study of speaker verification systems using the Polycost database. In *Proc. 1998 International Conference on Spoken Language Processing (ICSLP)*, pages 1359–1362, Sydney, Australia.

Oglesby, J. (1995). What's in a number? moving beyond the equal error rate. *Speech Communication*, 17(1-2):193–208.

Olsen, J. (1997). Speaker verification based on phonetic decision making. In *Proc. 5th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1375–1378, Rhodes, Greece.

Olsen, J. (1998a). Speaker recognition based on discriminative projection models. In *Proc. 1998 International Conference on Spoken Language Processing (ICSLP)*, pages 1919–1922, Sydney, Australia.

Olsen, J. (1998b). Speaker verification with ensemble classifiers based on linear speech transforms. In *Proc. 1998 International Conference on Spoken Language Processing (ICSLP)*, pages 1915–1918, Sydney, Australia.

Olsen, J. (1998c). Using ensemble techniques for improved speaker modelling. In *Proc. Nordic Signal Processing Symposium (NORSIG)*, pages 85–88, Vigsø. Denmark.

Olsen, J. and Lindberg, B. (1999). Algorithms & parameters for speaker recognition: activities in COST250 working group 3. In Falcone, M., editor, *COST250 Speaker Recognition in Telephony, Final Report (CD-ROM)*. European Commission, DG XIII-B, Brussels.

Olsson, J. (2002). Text dependent speaker verification with a hybrid HMM/ANN-system. Master's thesis, KTH/TMH, Stockholm, Sweden.

Ortega-Garcia, J. and Bousono-Crespo, C. (2005). Report on existing biometric databases. Technical Report IST-2002-507634-D1.1.1, BioSecure.

Ortega-Garcia, J., Gonzalez-Rodriguez, J., and Marrero-Aguiar, V. (2000). AHUMADA: A large speech corpus in spanish for speaker characterization and identification. *Speech Communication*, 31(2-3):255–264.

Pakucs, B. (2003). Sesame: A framework for personalized and adaptive speech interfaces. In *Proc. EACL-03 Workshop on Dialogue Systems: interaction, adaptation and styles of management*, pages 95–102, Budapest, Hungary.

Parthasarathy, S. and Rosenberg, A. (1996). General phrase speaker verification using sub-word background models and likelihood-ratio scoring. In *Proc. 1996 International Conference on Spoken Language Processing (ICSLP)*, pages 2403–2406, Philadelphia PA, USA.

Payton, K. (1988). Vowel processing by a model of the auditory periphery: A comparison to eighth-nerve responses. *The Journal of The Acoustical Society of America*, 83(1):145–162.

Pelecanos, J. and Sridharan, S. (2001). Feature warping for robust speaker verification. In *Proc. Speaker Odyssey - The Speaker Recognition Workshop*, pages 213–218, Crete, Greece.

Pellom, B. and Hansen, J. (1999). An experimental study of speaker verification sensitivity to computer voice-altered imposters. In *Proc. 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 837–840, Phoenix AZ, USA.

Plumpe, M., Quatieri, T., and Reynolds, D. (1999). Modeling of the glottal flow derivative waveform with application to speaker identification. *IEEE Transactions on Speech and Audio Processing*, 7(5):569–586.

Politis, D. (1998). Computer-intensive methods in statistical analysis. *IEEE Acoustics, Speech, and Signal Processing Magazine*, 15(1):39–55.

Poritz, A. (1982). Linear predictivee hidden Markov models and the speech signal. In *Proc. 1982 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 7, pages 1291–1294, Paris, France.

Potamianos, A., Kuo, H.-K., Lee, C.-H., Pargellis, A., Saad, A., and Zhou, Q. (1999). Design principles and tools for multimodal dialog systems. In *Proc of ESCA Workshop on Interactive Dialogue in Multi-Modal Systems*, pages 169–172, Kloster Irsee, Germany.

Przybocki, M. and Martin, A. (2004). NIST speaker recognition evaluation chronicles. In *Proc. Odyssey04 - The Speaker and Language Recognition Workshop*, pages 15–22, Toledo, Spain.

Quatieri, T., Malyska, N., and Sturim, D. (2003). Auditory signal processing as a basis for speaker recognition. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 111–114, New Palts, NY, USA.

Rabiner, L., Rosenberg, A., and Levinson, S. (1978). Considerations in dynamic time warping algorithm for discrete word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(6):575–582.

Reynolds, D. (1994). Speaker identification and verification using gaussian mixture speaker models. In *Proc. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 27–30, Martigny, Switzerland.

Reynolds, D. (1995). Speaker identification and verification using gaussian mixture speaker models. *Speech Communication*, 17(1-2):91–108.

Reynolds, D. (1997a). Comparison of background normalization methods for text-independent speaker verification. In *Proc. 5th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 963–966, Rhodes, Greece.

Reynolds, D. (1997b). HTIMIT and LLHDB: Speech corpora for the study of handset transducer effects. In *Proc. 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1535–1538, Munich, Germany.

Reynolds, D. (2003). Channel robust speaker verification via feature mapping. In *Proc. 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 53–56, Hong Kong.

Reynolds, D., Andrews, W., Campbell, J., Navratil, J., Peskin, B., Adami, A., Jin, Q., Klusacek, D., Abramson, J., Mihaescu, R., Godfrey, J., Jones, D., and Xiang, B. (2003). The SuperSID project: Exploiting high-level information for high-accuracy speaker recognition. In *Proc. 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 784–787, Hong Kong.

Reynolds, D., Quatieri, T., and Dunn, R. (2000). Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10:19–41.

Rose, R. and Reynolds, D. (1990). Text-independent speaker identification using automatic acoustic segmentation. In *Proc. 1990 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 293–296, Albuquerque NM, USA.

Rosenberg, A. (1976). Automatic speaker verification: A review. *Proceedings of the IEEE*, 64(4):475–487.

Rosenberg, A., DeLong, J., Lee, C.-H., Juang, B.-H., and Soong, F. (1992). The use of cohort normalized scores for speaker verification. In *Proc. 1992 International Conference on Spoken Language Processing (ICSLP)*, pages 599–602, Banff, Canada.

Rosenberg, A., Lee, C.-H., and Gokcen, S. (1991). Connected word talker verification using whole word hidden markov models. In *Proc. 1991 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 381–384, Toronto, Canada.

Rosenberg, A., Lee, C.-H., and Soong, F. (1990). Sub-word unit talker verification using hidden markov models. In *Proc. 1990 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 269–272, Albuquerque NM, USA.

Rosenberg, A., Lee, C.-H., and Soong, F. (1994). Cepstral channel normalization techniques for HMM-based speaker verification. In *Proc. 1994 International Conference on Spoken Language Processing (ICSLP)*, pages 1835–1838, Yokohama, Japan.

Rosenberg, A., Siohan, O., and Parthasarathy, S. (1998). Speaker verification using minimum verification error training. In *Proc. 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 105–108, Seattle WA, USA.

Rosenberg, A. and Soong, F. (1987). Evaluation of vector quantization talker recognition system in text independent and text dependent modes. *Computer Speech and Language*, 2(3/4).

Rua, E., Agulla, E., Mateo, C., and Florez, O. (2004). User verification in a BioVXML framework. In *Proc. Odyssey04 - The Speaker and Language Recognition Workshop*, pages 141–144, Toledo, Spain.

Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.

Salvi, G. (1998). Developing acoustic models for speech recognition. Master's thesis, KTH/TMH, Stockholm, Sweden.

Sambur, M. (1975). Selection of acoustic features for speaker identification. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(2):176–182.

Savic, M. and Gupta, S. (1990). Variable parameter speaker verification system based on hidden markov modeling. In *Proc. 1990 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 281–284, Albuquerque NM, USA.

Schalk, H., Reininger, H., and Euler, S. (2001). A system for text dependent speaker verification - field trial evaluation and simulation results. In *Proc. 7th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 783–786, Aalborg, Denmark.

Schalk, T. (1991). Speaker verification over the telephone network. *Speech Technology*, 5(3).

Schmidt, M. and Gish, H. (1996). Speaker identification via support vector classifiers. In *Proc. 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 105–108, Atlanta GE, USA.

Schuckers, M. (2003a). Estimation and sample size calculations for matching performance of biometric identification devices. Technical report, Center for Identification Technology Research (CITeR), http://www.citer.wvu.edu/.

Schuckers, M. (2003b). Using the beta-binomial distribution to assess performance of a biometric identification device. *International Journal of Image and Graphics*, 3(3):523–529.

Schuckers, M., Hawley, A., Livingstone, K., and Mramba, N. (2004). A comparison of statistical methods for evaluating matching performance of a biometric identification device: a preliminary report. *Proceedings of SPIE Biometric Technology for Human Identification*, 5404:144–155.

Seneff, S., Hurley, E., Lau, R., Pao, C., Schmid, P., and Zue, V. (1998). Galaxy-II: A reference architecture for conversational system development. In *Proc. 1998 International Conference on Spoken Language Processing (ICSLP)*, volume 3, pages 931–934, Sydney, Australia.

Senia, F. and van Velden, J. (1997). Specification of orthographic transcription and lexicon conventions. Technical Report LE2-4001-SD1.3.2, SpeechDat.

Setlur, A. and Jacobs, T. (1995). Results of a speaker verification service trial using HMM models. In *Proc. 4th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 639–642, Madrid, Spain.

Seward, A. (2000). A tree-trellis n-best decoder for stochastic context-free grammars. In *Proc. 2000 International Conference on Spoken Language Processing (ICSLP)*, volume 4, pages 282–285, Beijing, China.

Shriberg, E., Ferrer, L., Kajarekar, S., Venkataraman, A., and Stolcke, A. (2005). Modeling prosodic feature sequences for speaker recognition. *Speech Communication*, 46(3-4):455–472.

Siafarkas, M., Ganchev, T., and Fakotakis, N. (2004). Wavelet packet based speaker verification. In *Proc. Odyssey04 - The Speaker and Language Recognition Workshop*, pages 257–264, Toledo, Spain.

Siegel, S., editor (1956). *Nonparametric statistics for the behavioral sciences.* McGraw-Hill, New York.

Siohan, O., Lee, C.-H., Surendran, A., and Li, Q. (1999). Background model design for flexible and portable speaker verification systems. In *Proc. 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 825–829, Phoenix AZ, USA.

Sjölander, K. and Beskow, J. (2000). Wavesurfer - an open source speech tool. In *Proc. 2000 International Conference on Spoken Language Processing (ICSLP)*, volume 4, pages 464–467, Beijing, China.

Slyh, R., Hansen, E., and Anderson, T. (2004). Glottal modeling and closed-phase analysis for speaker recognition. In *Proc. Odyssey04 - The Speaker and Language Recognition Workshop*, pages 315–322, Toledo, Spain.

Snedecor, G. and Cochran, W. (1967). *Statistical methods.* Iowa State University Press, Ames, Iowa, 6th edition.

Solomonoff, A., Campbell, W., and Boardman, I. (2005). Advances in channel compensation for SVM speaker recognition. In *Proc. 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 629–632, Philadelphia PA, USA.

Soong, F. and Rosenberg, A. (1986). On the use of instantaneous and transitional spectral information in speaker recognition. In *Proc. 1986 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 877–880, Tokyo, Japan.

Ström, N. (1996). Continuous speech recognition in the WAXHOLM dialogue system. *TMH-QPSR*, 37(4):67–96.

Surendran, A. (2001). Sequential decisions for faster and more flexible verification. In *Proc. 7th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 763–766, Aalborg, Denmark.

Sutton, S., Cole, R., Villiers, J., Schalkwyk, J., Vermeulen, P., Macon, M., Yan, Y., Kaiser, E., Rundle, B., Shobaki, K., Hosom, P., Kain, A., Wouters, J., Massaro, D., and Cohen, M. (1998). Universal speech tools: the CSLU toolkit. In *Proc. 1998 International Conference on Spoken Language Processing (ICSLP)*, volume 7, pages 3221–3224, Sydney, Australia.

Svanfeldt, G. (2003). A speech interface demonstrator for pre-operative planning within orthopaedic surgery. Master's thesis, KTH/TMH, Stockholm, Sweden.

Söderquist, H. (2002). An application-independent speaker adaptation service. Master's thesis, KTH/TMH, Stockholm, Sweden.

Sönmetz, K., Heck, L., Weintraub, M., and Shriberg, E. (1997). A lognormal tied mixture model of pitch for prosody-based speaker recognition. In *Proc. 5th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1391–1394, Rhodes, Greece.

Testi, D., Zannoni, C., Petrone, M., Clapworthy, G., Neiberg, D., Tsagarakis, N., Caldwell, D., and Viceconti, M. (2005). A multimodal and multisensorial preoperative planning environment for total hip replacement. In *Proc.IEEE Medical Information Visualisation (MediVis 05)*, pages 25–29, London, England. IEEE Computer Society Press.

Teunen, R., Shahshahani, B., and Heck, L. (2000). A model-based transformational approach to robust speaker recognition. In *Proc. 2000 International Conference on Spoken Language Processing (ICSLP)*, pages 495–498, Beijing, China.

Thompson, J. and Mason, J. (1994). The pre-detection of error-prone class members at the enrollment stage of speaker recognition systems. In *Proc. ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 127–130, Martigny, Switzerland.

Thumernicht, C. (2002). Evaluation of an auditory model for recognition of speech in noise. Master's thesis, KTH/TMH, Stockholm, Sweden.

Thévenaz, P. and Hügli, H. (1995). Usefulness of the LPC-residue in text-independent speaker verification. *Speech Communication*, 17(1-2):145–157.

Tran, D. and Wagner, M. (2001). A generalised normalisation method for speaker verification. In *Proc. Speaker Odyssey - The Speaker Recognition Workshop*, pages 73–76, Crete, Greece.

Turunen, M. and Hakulinen, J. (2000). Jaspis - a framework for multilingual adaptive speech applications. In *Proc. 2000 International Conference on Spoken Language Processing (ICSLP)*, volume 2, pages 719–722, Beijing, China.

Vapnik, V., editor (1995). *The Nature of Statistical Learning Theory.* Springer.

Wagner, T. and Dieckmann, U. (1995). Sensor-fusion for robust identification of persons: A field test. In *Proc. IEEE International Conference on Image Processing*, volume 3, pages 516–519, Washington D.C., USA.

Wan, V. and Campbell, W. (2000). Support vector machines for speaker verification and identification. In *Proc. IEEE Workshop on Neural Networks for Signal Processing*, pages 775–784, Sydney, Australia.

Wan, V. and Renals, S. (2002). Evaluation of kernel methods for speaker verification and identification. In *Proc. 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 669–672, Orlando FL, USA.

Winkler, R., Brückl, M., and Sendlmeier, W. (2003). The aging voice: an acoustic, electroglottographic and perceptive analysis of male and female voices. In *Proc. 15th International Conference of Phonetic Sciences (ICPhS)*, pages 2869–2872, Barcelona, Spain.

Wolf, J. (1972). Efficient acoustic parameters for speaker recognition. *The Journal of The Acoustical Society of America*, 51(2):2044–2055.

Xiang, B., Chaudhari, U., Navratil, J., Ramaswamy, G., and Gopinath, R. (2002). Short-time Gaussianization for robust speaker verification. In *Proc. 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 681–684, Orlando FL, USA.

Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. (1999). *The HTK Book (for HTK version 2.2).* Cambridge University, Cambridge, UK.

Zetterholm, E., Blomberg, M., and Elenius, D. (2004). A comparison between human perception and a speaker verification system score of a voice imitation. In *Proc. the 10th Australian International Conference on Speech Science and Technology*, pages 393–397, Sydney, Australia.

Öhlin, D. (2004). Formantextraktion för datadriven formantsyntes (in Swedish). Master's thesis, KTH/TMH, Stockholm, Sweden.

Appendices

Appendix A

PER photos

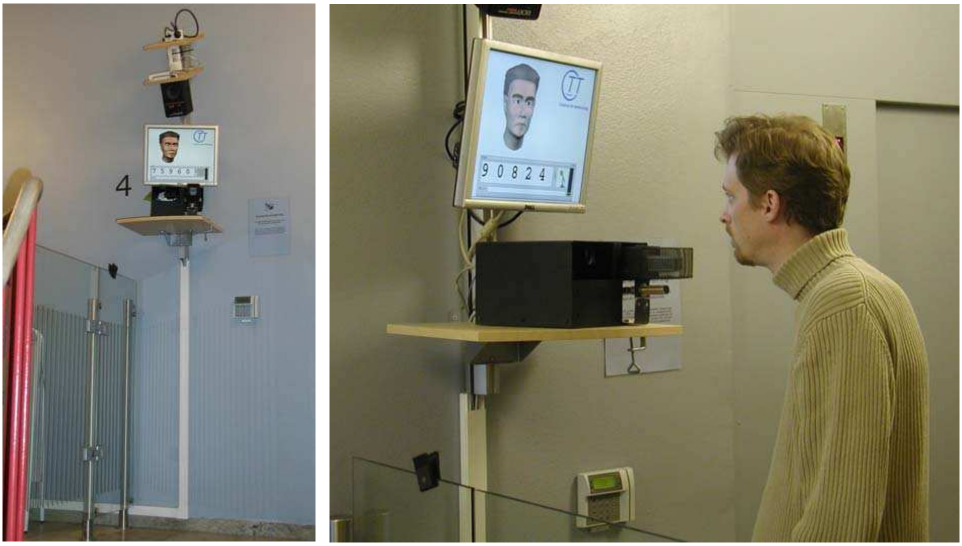


Figure A.1: The new installation of PER at Lindstedtsvägen 24. Photos by Botond Pakucs.

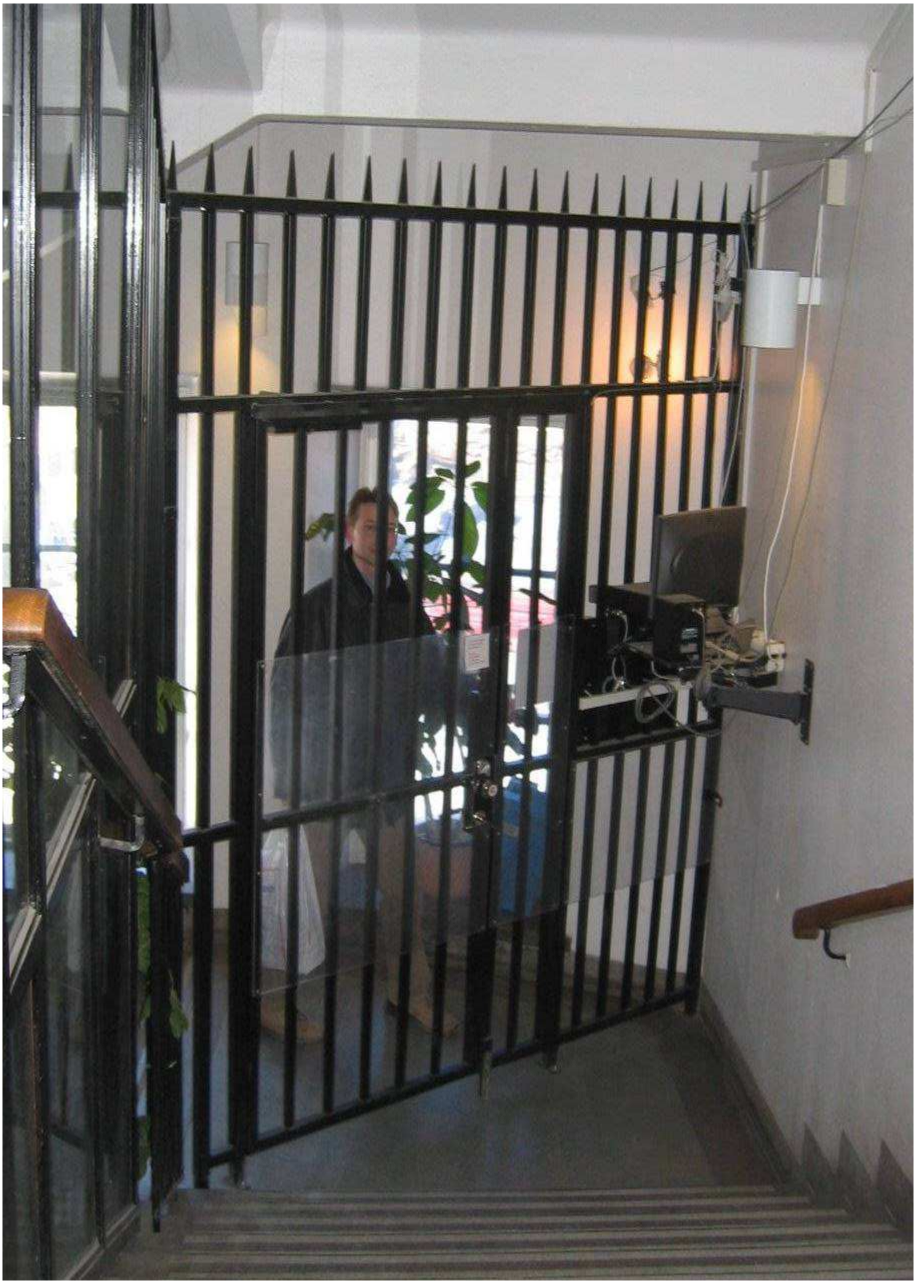


Figure A.2: View of the bar gate with the on-site version of PER at Drottning Kristinas väg 31. Photo by Botond Pakucs.

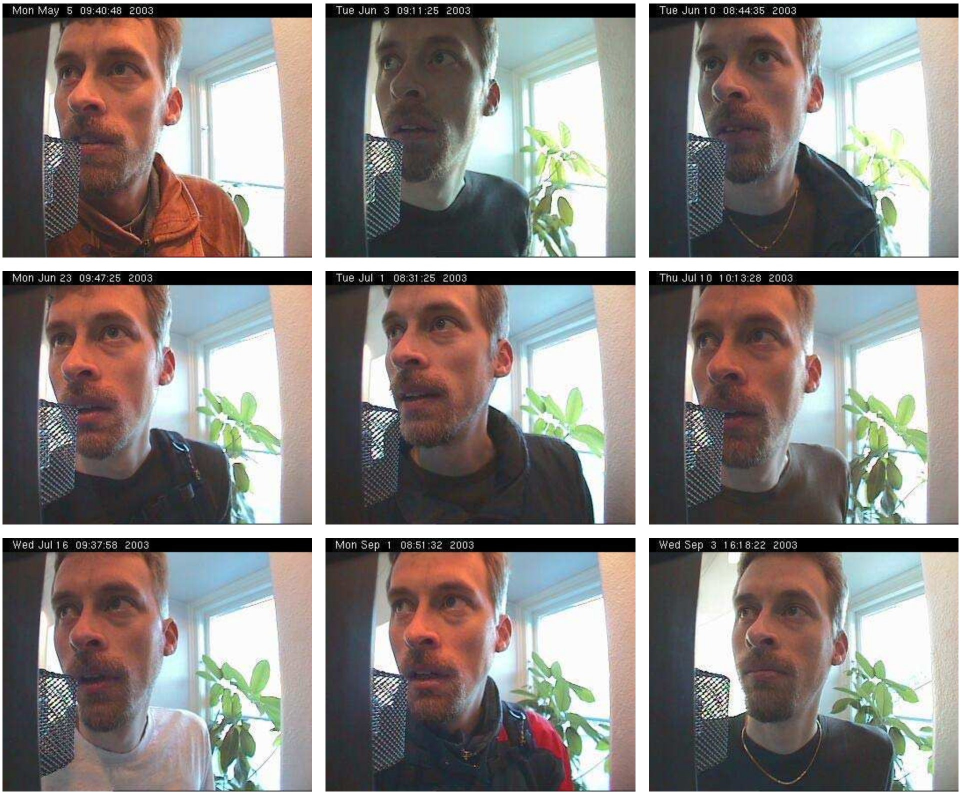


Figure A.3: Photos of the author taken during nine distinct sessions between May and September 2003 by PERs network camera. The images are captured at the time when the photo electric sensor first signals the presence of a person.

Appendix B

Gandalf - data sets and extended statistics

This Appendix presents additional data sets defined for the Gandalf corpus and additional statistics not included in Section 6.2 (p. 89).

B.1 Data sets

This section provides additional details and background information on data sets not included in Section 6.2.4 (p. 99), including data sets that have been defined but were not used in this thesis.

The assignment of subjects into development, evaluation and background set shown in Table B.1 and referred to as Division-1 was created this way:

- Only subjects who allowed other labs to use their recordings were used (this eliminated `F056`, `M066`, `M127`, `F134`, `F151`, `F157`, `M173` and `F186`).
- 15 male and 15 female non-client subjects were assigned to the background speaker set.
- The first half of the female client subjects and the first half of the male client subjects were assigned to the development set to be used as both clients and impostors.
- The remaining client subjects were assigned to the evaluation set to be used as both clients and impostors.
- The remaining non-client subjects were assigned to the evaluation set to be used as impostors only. They were assigned to the evaluation set rather than being split onto the development and evaluation sets to keep the development set smaller (it is run more often) and to provide for better statistical significance in tests on the evaluation set.

Table B.1: Gandalf subjects assigned to the background, development and evaluation sets.

Set	Subjects
Background	Background speakers (15F, 15M): F102, F103, F116, F117, F129, F130, F136, F137, F147, F148, F152, F156, F179, F180, F192, M100, M101, M104, M105, M106, M107, M110, M111, M112, M113, M114, M115, M118, M124, M128
Development	Client and impostor (18F, 22M): F016, F018, F019, F022, F023, F024, F025, F026, F027, F028, F030, F031, F032, F035, F037, F041, F044, F047, M010, M011, M012, M013, M014, M015, M017, M020, M021, M029, M033, M034, M038, M040, M042, M043, M045, M046, M048, M050, M054, M055 Impostor only (1M): M036
Evaluation	Client and impostor (18F, 24M): F049, F051, F052, F053, F062, F063, F064, F065, F070, F071, F072, F073, F074, F083, F084, F089, F091, F095, M057, M058, M059, M060, M061, M068, M069, M075, M076, M077, M078, M079, M080, M081, M082, M085, M086, M087, M088, M090, M092, M093, M094, M096 Impostor only (14F, 34M): F067, F109, F120, F122, F133, F135, F138, F144, F150, F166, F167, F169, F194, F199, M121, M125, M126, M131, M132, M139, M140, M141, M142, M143, M145, M153, M154, M155, M161, M162, M163, M164, M165, M175, M177, M178, M181, M182, M184, M185, M187, M188, M189, M190, M191, M195, M197, M198

- It was then decided to use as clients only subjects who completed all the (enrollment) calls 1, 2 and 99. This disqualified M036 and F067 (who both miss call 99) and they were used as impostors only.

B.2 Extended call statistics

Table B.7 shows the distribution of types of locations that the subjects called from as indicated by call response sheets (CRS).

Subjects were encouraged to make some calls during collection Part 1 from abroad. A total of 33 calls were international calls. They were distributed as follows. Three client subjects (M012, F051 and M092) made four calls each from abroad, four subjects made two calls each and 11 subjects made a single international call. One of the impostor subjects (M143) made both his calls from abroad. Of the 33 international calls, 14 were made from France, 11 from Great Britain and the remaining eight from six other countries. The 31 international calls from client

Table B.2: Gandalf enrollment sets defined on five-digit sequences (d5), varied sentences (vs), and fixed sentence (fs0n, with $n \in \{1, 2\}$). Files are specified on the format session/filename.

Composition	Phrase	t	Files
1s1h	d5	1	`001/D5{01-25}`
		0.5	`001/D5{01-12}`
		0.3	`001/D5{01-07}`
	vs	0.5	`001/VS{01-10}`
		0.25	`001/VS{01-05}`
		0.15	`001/VS{01-03}`
	fs0n	0.25	`001/FS0n{01-05}`
		0.15	`001/FS0n{01-03}`
2s1h[a]	d5	1	`001/D5{01-12}`, `099/D5{13-25}`
		0.5	`001/D5{01-06}`, `099/D5{07-12}`
		0.3	`001/D5{01-04}`, `099/D5{05-07}`
	vs	0.5	`001/VS{01-05}`, `099/VS{06-10}`
		0.25	`001/VS{01-03}`, `099/VS{04-05}`
		0.15	`001/VS{01-02}`, `099/VS03`
	fs0n	0.25	`001/FS0n{01-03}`, `099/FS0n{04-05}`
		0.15	`001/FS0n{01-02}`, `099/FS0n03`
2s2h	d5	1	`001/D5{01-12}`, `002/D5{13-25}`
		0.5	`001/D5{01-06}`, `002/D5{07-12}`
		0.3	`001/D5{01-04}`, `002/D5{05-07}`
	vs	0.5	`001/VS{01-05}`, `002/VS{06-10}`
		0.25	`001/VS{01-03}`, `002/VS{04-05}`
		0.15	`001/VS{01-02}`, `002/VS03`
	fs0n	0.25	`001/FS0n{01-03}`, `002/FS0n{04-05}`
		0.15	`001/FS0n{01-02}`, `002/FS0n03`
	d5+fs0x	1	`001/D5{01-05}`, `002/D5{06-10}`, `001/FS0n{01-05}`), `002/FS0n{01-05}`)[b]

[a] the second of the two sessions (099) is nominally recorded four months into the collection period

[b] $n = 1$ for half the targets and $n = 2$ for the other half

Table B.3: Gandalf test sets based on digit utterances.

Test set	Text	Prompt	Files	Comment
1d4-d40n	4 digits	visual	D40n	the same digit sequence across all targets and calls (four test sets; $n \in \{1 \ldots 4\}$)
1d3	3 digits	visual	D301 or[a] D302 or D303 or D304	the same digit sequence for a given target across all calls
1d4	4 digits	visual	D401 or D402 or D403 or D404	the same digit sequence for a given target across all calls
2d3	6 digits	visual	(D301, D302) or (D302, D303) or (D303, D304) or (D304, D305)	the same digit sequences for a given target across all calls
2d4	8 digits	visual	(D401, D402) or (D402, D403) or (D403, D404) or (D404, D401)	the same digit sequences for a given target across all calls
1r4	4 digits	aural	R401 or R402	sequences are picked at random
2r4	8 digits	aural	R401, R402	sequences are picked at random

[a] D301 is used in all attempts against the first $1/N$ fraction of the target group, D302 against the next $1/N$ fraction, etc., where N is the number of alternatives ($N = 4$ in this case).

subjects correspond to 2.2% of all calls in Part 1, or 1.7% of the client test calls in all three parts.

On the call response sheet (CRS), subjects were asked to specify any illness that may have affected their voice[2] during the call in terms of: "no illness that I think affected my voice" vs. combinations of "a little sore throat" or "very sore throat"; "slight runny nose" or "very runny nose"; "fever"; and "something else that I think affected my voice" (open question). For the purpose of presenting the outcome of collected replies, we here define an *illness* to be in effect if any of the sore throat, runny nose or fever options were checked; and a *severe illness* to be in effect if at least one of "very sore throat", "very runny nose" or "fever" were checked. Comments written to the "something else…" option were very diverse, so they are reported separately below.

[2] the Swedish term used was "röst", which in the speech science community usually refers to the voice source, but we believe lay people relate this term more to a broader sense of how a person's speech sounds.

Table B.4: Gandalf test sets based on sentence utterances and combinations of digit and sentence utterances.

Test set	Text	Prompt	Files	Comment
1vs	1 sent.	visual	VS01 or VS02 or VS03 or VS04	VS01 contains the same sentence across targets for a given call number; impostors use files from call 3
1vs-2[a]	1 sent.	visual	VS01 or VS02 or VS03 or VS04	VS01 contains the same sentence across targets for a given call number
2vs	2 sent.	visual	(VS01, VS02) or (VS02, VS03) or (VS03, VS04) or (VS04, VS01)	VS01 contains the same sentence across targets for a given call number
2vs-2[b]	2 sent.	visual	(VS01, VS02) or (VS02, VS03) or (VS03, VS04) or (VS04, VS01)	VS01 contains the same sentence across targets for a given call number
1rs	1 sent.	aural	RS01	sentences are picked at random; "short" sentences
1rl	1 sent.	aural	RL01	sentences are picked at random; "long" sentences
1fs-fs01	1 sent.	visual	FS01	the same sentence across all targets and calls
1fs-fs02	1 sent.	visual	FS02	the same sentence across all targets and calls
1fs+1r4-fs0x	1 sent., 4 digits	visual, aural	(FS01, R401) or (FS02, R402)	the same sentence across calls; digit sequences are picked at random; designed as development set for PER experiments

[a] Impostors use files from different calls, such that a given impostor use the same sentence against all targets, and no two impostors use the same sentence.

[b] impostors use files from different calls, such that a given impostor use the same pair of sentences against all targets, and no two impostors use the same sentence pair.

Table B.5: Relations among subjects in the Gandalf database. Groups of subjects with similar voices, according to their own opinion, have been underlined. (Referred to on p. 93.)

Relation	Subjects
Identical twins	F083–F084, M079–M080
Siblings	F024–F135, F025–F138, F037–F095, F051–F053, F065–F169, M010–M046, M012–M126–M127, M029–M140–M141–M143, M040–M145, M048–M076, M068–M164, M085–M175, M086–M187, M096–M185
Parent–child (same gender)	F024–F070, F047–F044, F052–F051, F052–F053, F091–F031, F120–F109, F144–F037, F144–F095, F150–F049, F151–F052, F166–F071, F167–F073, F186–F074, M012–M125, M021–M131, M045–M010, M045–M046, M086–M177, M121–M081, M139–M029, M139–M140, M139–M141, M139–M143
Parent–child (different gender)	F035–M029, F044–M010, F044–M046, F072–M033, F122–M081, F133–M131, M012–F070, M055–F051, M055–F053
Cousins (same gender)	M010–M036, M010–M048, M010–M076, M036–M046, M036–M048, M036–M076, M046–M048, M046–M076

Table B.6: Client subjects who called from other handsets than their designated favorite handset during collection parts 2 and 3 (calls 17-28). (Referred to on p. 95.)

Subject	Call range	Number of calls
M013	28	1
F016	17–28	12
F023	18–28	11
F030	20–28	9
F037	26–28	3
M050	23–28	6
F051	17	1
F052	27	1
F065	25–28	4
M068	17–23	7
M078	27–28	2
M082	20–28	9
M087	24–28	5

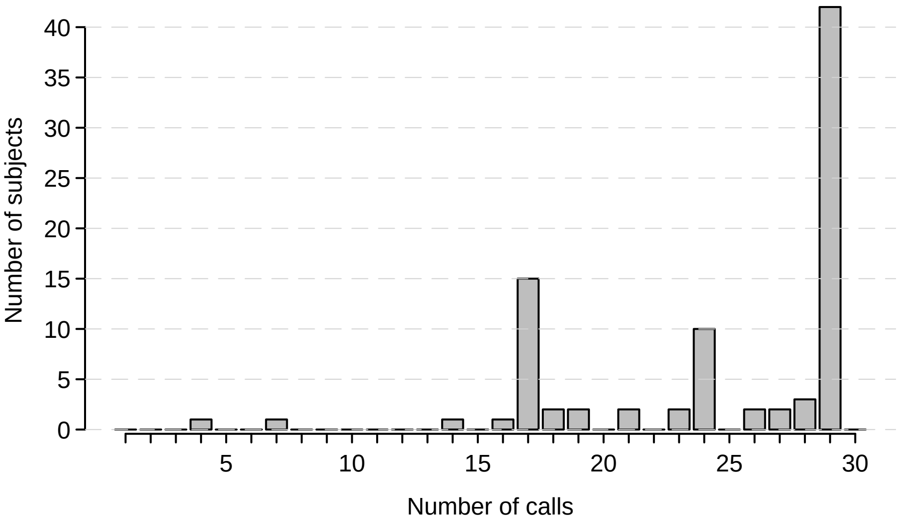


Figure B.1: Histogram on how many client subjects recorded how many calls in Gandalf. The bars at 17, 24 and 29 calls show how subjects completed Parts 1, 2 and 3 of data collection. (Referred to on p. 95.)

Table B.7: Number of calls from different types of locations for Gandalf calls from favorite and non-favorite handsets, respectively. The non-favorite handset section also shows the proportion of calls made from mobile phone and public pay phones.

	Client calls					Impostor calls	
	Favorite handset		Non-favorite handset			Enroll	Test
Location	Subjects	Calls	Calls	Mobile phone	Pay phone	calls	calls
Home[a]	54	846	414	12%		56	58
Office	31	469	227	14%		25	20
Phone booth			51		100%		
Public room[b]			36	11%	58%	1	3
Car			10	100%			
Outdoors			9	78%	11%		
Other	1[c]	21[c]	13[d]	31%	15%		1

[a] including hotel room

[b] The exact term used on the CRS was “stor lokal med många människor” (*large room with many people*).

[c] location specified as an open-plan office

[d] open-plan office(3), computer/copy-machine room(3), restaurant(2), sound-proof room(1), train(1), dressing room(1), unspecified(2)

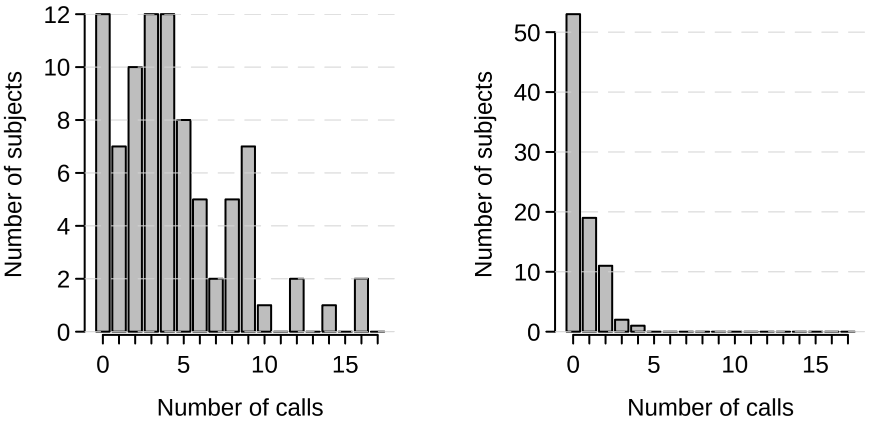


Figure B.2: Histogram on the number of subjects who made how many calls during illness. The left pane includes any degree of illness, while the right pane includes the more severe degrees of illness only (very sore throat, very runny nose or fever).

In 51 of the 2106 calls from client subjects (2.4%) a severe illness was in effect. These calls are distributed as shown in the right pane of Figure B.2. The subject with four such calls is `F016`, while subjects `F052` and `M078` have three calls each. 53 subjects reported no call with severe illness. Five of the severe illness calls were enrollment calls (subject/call): `F024/001` (fever), `F051/001` (sore throat), `F053/099` (runny nose), `F056/001` (runny nose), and `M076/099` (runny nose).

If we also include the slighter versions of illness there were 378 calls from client subjects (18%). The left pane of Figure B.2 shows the histogram on how many such calls were made by how many client subjects.

In 53 of the illness calls (as defined above), a subject *also* specified another source of voice alteration through the “something else that I think affected my voice” option. In 124 other calls (5.9% of client calls), client subjects specified another source of voice alteration *without* checking any of the nose, throat or fever options. Table B.8 shows an attempt to categorize the various sources suggested by subjects for those calls. Note that the table shows what *subjects believed* may have altered their voice. These sources may or may not correspond to voice changes that are actually measurable in the produced speech, but it may still be interesting for the reader to see what factors appear important to subjects. Note also that, naturally, subjects have individual opinions on what factors are influential, and thus, the frequency counts in the table are not likely to indicate probabilities of factor occurrence in the general population.

The call response sheet also requested information about background noise in terms of “it was quiet” vs. combinations of “there was some occasional noise” and

Table B.8: Categorization of client subject responses to the open question "something else that I think affected my voice" after Gandalf calls in cases were subjects did not also check any of the throat, nose or fever options.

Category			
	Subcategory	Examples	Cases
throat related			
	abnormality	hoarseness, laryngitis, other vocal cord problem, dryness, allergic reaction, phlegm, coughing, harking, creaky voice	42
	tiredness	from using voice; from singing	6
	startup	first speaking in morning; have not spoken for a long while	11
nose related		stuffed nose, hay fever	6
mouth related		blister on tongue, oral infection	4
influence from substance		anesthetic (by dentist), alcohol, medicine, smoking (active or passive), eating (while calling), drinking (while calling)	19
mental or physical state		toothache, headache, stomachache, stress, tension, tired, sleepy, physical exercise, distracted, happiness, laughter	34
position		leaning backwards while sitting	1
other		lots of dust in the room	1

"there was persistent noise". The two noise options had sub-options for the noise source: "talk", "music" and "car noise". In addition to checking our pre-specified options, subjects often added their own comments in the margin to specify other noise sources.

Looking at calls from client subjects where the CRS indicated the presence of persistent noise, there were 241 such calls, corresponding to 11% of all 2106 client calls. In 194 calls (80% of 241) a single noise source was specified, while 47 calls had multiple noise sources. Looking at one noise source at the time, there were 72 calls with "talk" (43% as the single source), 89 calls with "music" (63% as the single source), 29 calls with "car noise" (76% as the single source), and 101 calls with another noise source only as indicated by margin comments (84% as the single source). "talk" and "music" occurred together in 30 calls.

In the calls with persistent "car noise", 16 (55%) came from a public payphone and 6 (21%) from a mobile phone. Table B.10 presents percentages of calls with any type of persistent noise made from public payphones, mobile phones vs. other

Table B.9: Categorization of other sources of persistent noise

Category Subcategory	Examples	Cases
Office environment		
computer	computer fan, computer noise	31
other	printer, copy machine, office noise	5
Home environment		
appliances	dish washer, tumble dryer, kitchen machine, boiling water, kitchen fan	10
broadcast	TV, radio	5
other	screaming child	1
Other ventilation	fan, ventilation	26
Public environment	train, traffic, plane, escalator	9
Construction work	pounding, drilling	5
Weather	rain, wind	2
Other	"noise", clatter, metallic sound, door slam, "was interrupted", unknown	7

types of phones.

Table B.9 gives an overview of other types of noise sources than our pre-specified options ("talk", "music" and "car noise") indicated by subjects in the margin of CRSs. Note that almost 60% of the calls listed in this table indicate noise from some kind of fan or ventilation system. This type of noise is usually of a more stationary nature than any of talk, music or car noise, and probably result in a higher signal-to-noise ratio (SNR) in a recorded speech signal, and thus we believe these calls will pose a lesser problem for automatic speech and speaker recognition than the latter types of noise. We also note that modern desktop computers have generally become more quiet since 1996 when Gandalf recordings were concluded.

Turning to calls from client subjects where the CRS indicated the presence of intermittent noise, there were 454 such calls, corresponding to 22% of all client calls. In 392 calls (86% of 454) a single noise source was specified, while 62 calls had multiple noise sources. Looking at one noise source at the time, there were 249 calls with "talk" (78% as the single source), 59 calls with "music" (66% as the single source), 75 calls with "car noise" (79% as the single source), and 103 calls with another noise source only as indicated by margin comments (79% as the single source). "talk" and "music" occurred together in 26 calls.

In the calls with intermittent "car noise", 20 (27%) came from a public payphone

Table B.10: Occurrence of any type of noise (talk, music, car or other) in calls by client subjects as indicated by call response sheets. The *any* column indicates in how many calls either or both intermittent and persistent noise was indicated. The *Other phone* category includes wired and cordless telephones in the landline network.

Telephone type	Calls	Noise occurrence			
		intermittent	persistent	any	both
Public payphone	73	47%	48%	92%	3%
Mobile phone	106	29%	16%	44%	-
Other phone	1952	20%	9.7%	29%	-

and 6 (8%) from a mobile phone. Table B.10 presents percentages of calls with any type of intermittent noise made from public payphones, mobile phones vs. other types of phones. It also indicates co-occurrence of intermittent and persistent noise.

Appendix C

PER - data sets

This Appendix presents additional data sets defined for the PER corpus not included in Section 6.3.4 (p. 110), including data sets that have been defined but were not used in this thesis. To make this Appendix self-contained, definitions already given in Section 6.3.4 are repeated here, in some cases re-formulated to fit a more generic framework.

C.1 Background speaker enrollment sets

Based on background speaker enrollment sessions, two enrollment sets per condition c were defined using text acceptance rule a and 51 male and 28 female background speakers that completed enrollment in all four conditions:

- E2a_c using the same definition as E2a_c set for client speakers but applied to the group of background speakers.
- E5a_c like E2a_c but also including the first repetition with accepted text status of five sentence items.

These enrollment sets can be used for example to create pseudo-impostor models for T-norm.

Enrollment sets E2a-gen_c and E2a-wld_c have been defined to train condition-dependent multi-speaker models. Speech data used in the former is the same as in E2a_c but with pooled data from all male or female background speakers, while in the latter all data were pooled, irrespective of gender, for training a "world" model.

C.2 Single-condition test sets

Separate sets of true-speaker and impostor test sets are defined first. Those are then combined into complete test sets.

Two sets of true-speaker test sets, T1x_c and T2x_c, and two sets of impostor test sets, I1x_c and I2x_c, have been defined. Common to all test sets is that they contain no more than one attempt from any given session, and only from login sessions annotated as valid and complete that contain at least one attempt whose file level transcription meet the conditions of the b-criterion for "accepted text status".

Common to both true-speaker and impostor test sets is that they contain no more than one attempt from any given session, and only from login sessions annotated as valid and complete that contain at least one attempt whose file level transcription meet the conditions of the b-criterion for "accepted text status".

True-speaker test sets include one attempt per session from all true-speaker login sessions. Impostor test sets include one attempt per combination of impostor speaker and target where the impostor speaker has recorded at least one session where (s)he claimed the given target identity. If there is more than one such session, the first one is used. Only same-sex impostor tests are used.

Test sets with index 1 and 2 differ in how attempts are selected from a session in which the PER system accepted the claimant. T1x_c and I1x_c include the last attempt (the attempt that was accepted during data collection), while T2x_c and I2x_c include the first attempt with an accepted text status. From sessions in which the PER system rejected the claimant, all test sets include the first attempt with an accepted text status. Hence, the test sets with index 2 always include the first attempt with an accepted text status, while those with index 1 change selection criterion with the decision of the collection system for each session.

True-speaker and impostor test sets are paired into the following complete test sets:

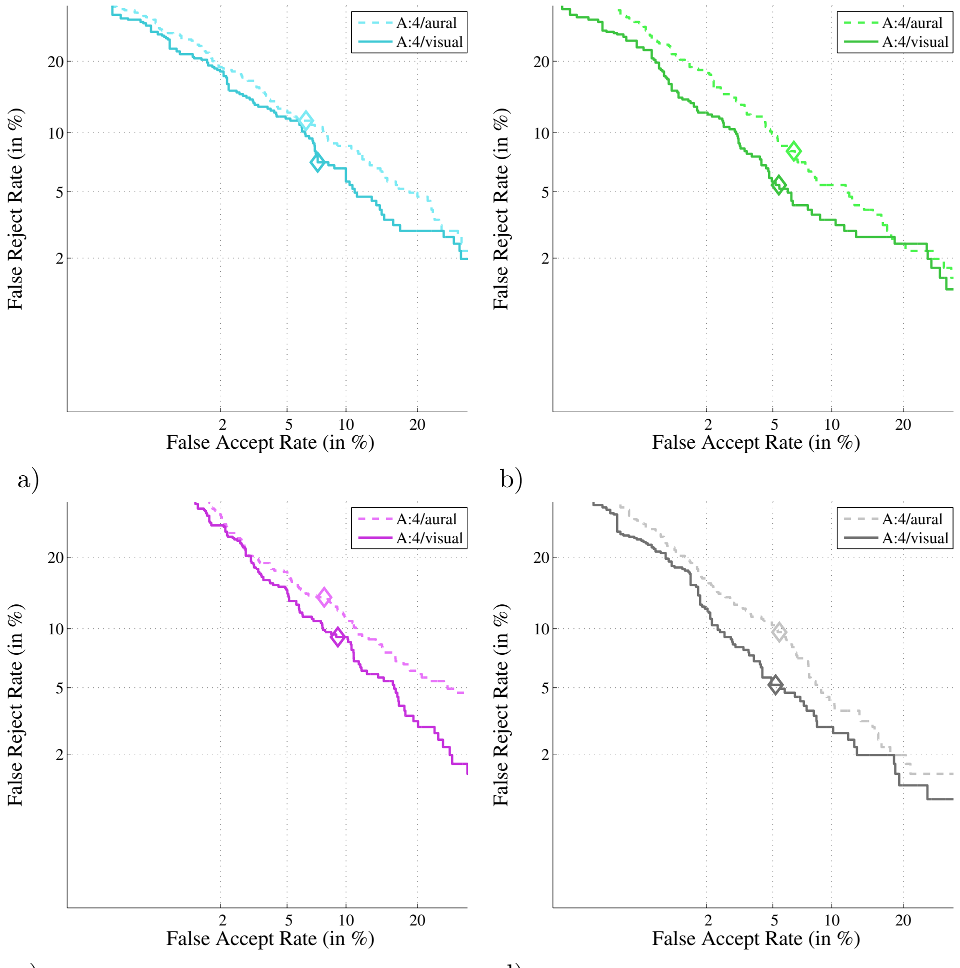
- S1x_c=T1x_c+I1x_c. This set most closely mimics the conditions of the final decision as taken in each login session of the data collection. The conditions include that claimants (both clients and impostors) get a new attempt if the text or voice test fails, up to three attempts total. Note that even if up to three attempts were indeed needed to produce the test utterance (or pair of utterances in telephone conditions), exactly one attempt from each session is included in the test set.

- S2x_c=T2x_c+I2x_c. This test set is like S1x_c, except it simulates that the claimant doesn't get a second attempt if the voice test fails, only if the text test fails. For the verification system used during the collection, this test set is more difficult on true-speaker tests (will lead to higher false reject rates) and easier on impostor tests (will lead to lower false accept rates) than S1x_c. This is because in those true-speaker sessions where S1x_c and S2x_c include different attempts, S2x_c will always include an attempt that was rejected during the data collection (except if the rejection was due to some system failure rather than a reject by the ASV system). Also in impostor sessions where S1x_c and S2x_c include different attempts, will S2x_c include an attempt that was rejected during the collection, but in this case "reject" is the correct decision.

Table 6.10 (p. 112) shows the number of speakers and tests included in the PER test sets used in the thesis, while Figure 6.6 shows how tests are distributed over targets in the G8 true-speaker and impostor test sets.

Appendix D

Prompting methods - extended results



a)

b)

c)

d)

Figure D.1: DET curves from Experiment A in Chapter 8 for a) the HMM-based CAVE system used in (Lindberg and Melin, 1997), b) the HMM subsystem, c) the GMM subsystem and d) the combined HMM and GMM system. The three latter systems are all described in Chapter 3. Each plot compares a pair of test sets with 4-digit strings prompted aurally and visually, respectively. Diamonds indicate the FRR/FAR pair for a threshold determined as the EER threshold on the visually prompted data.

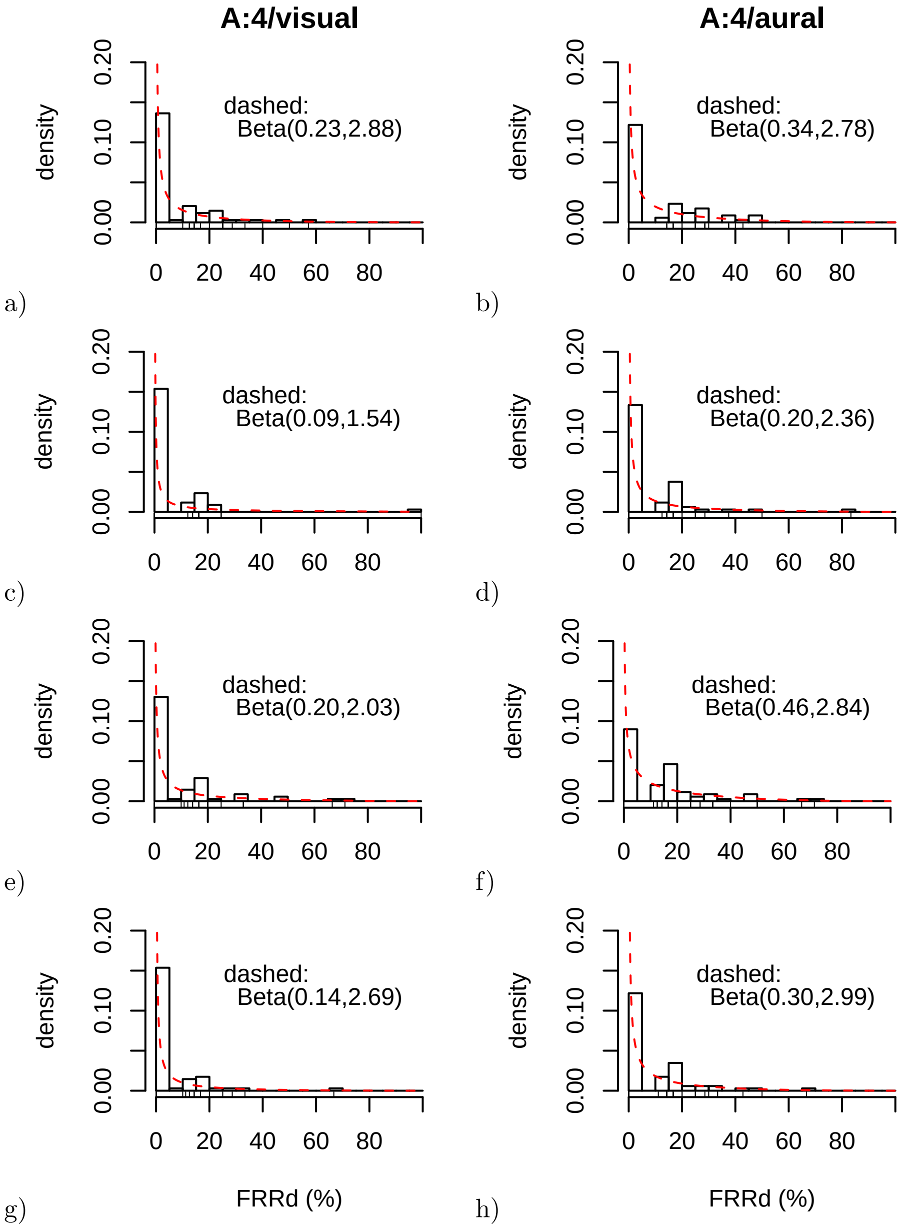


Figure D.2: Distribution of non-parametric FRR (FRRd) at a fixed target-independent threshold $\theta^{\mathrm{v}}_{\mathrm{EERd}}$ determined as the EER threshold on the visually prompted data over targets in Experiment A, with the ASV systems: CAVE system (a,b); HMM subsystem (c,d); GMM subsystem (e,f); and the combined HMM and GMM system (g,h). Dashed lines show fitted beta distributions (cf. Section 7.3.1.2, p. 135).

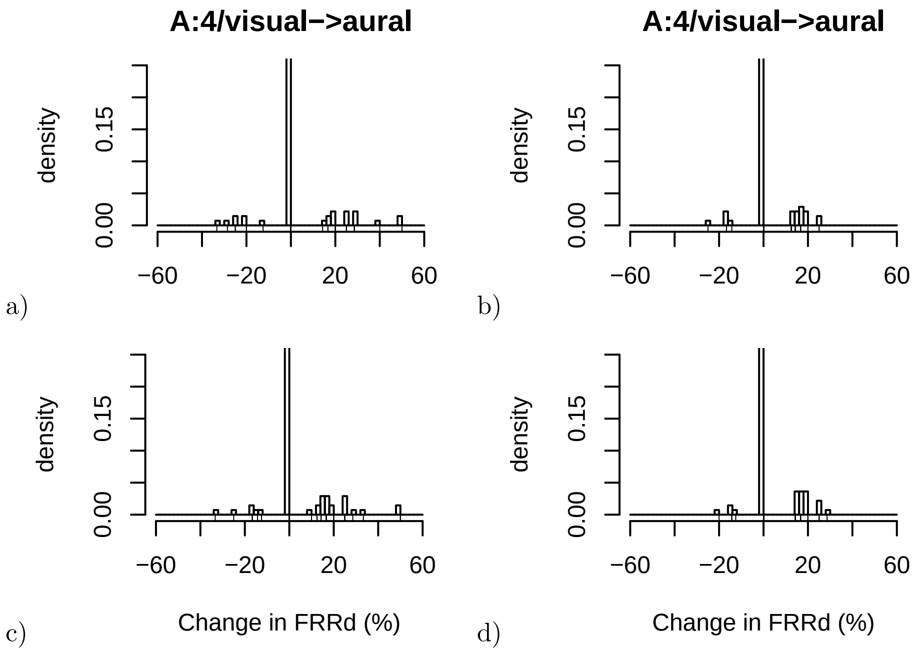


Figure D.3: Distribution over targets of differences in FRRd at a fixed target-independent, system-specific threshold $\theta^{\mathrm{v}}_{\mathrm{EERd}}$ determined as the EER threshold on the visually prompted data, when comparing the two sets in Experiment A. Results are given for four ASV systems: a) CAVE system, b) HMM subsystem, c) GMM subsystem, and d) the combined HMM and GMM system. A positive difference x indicates that the individual FRRd for aurally prompted items was x units higher than for visually prompted items.

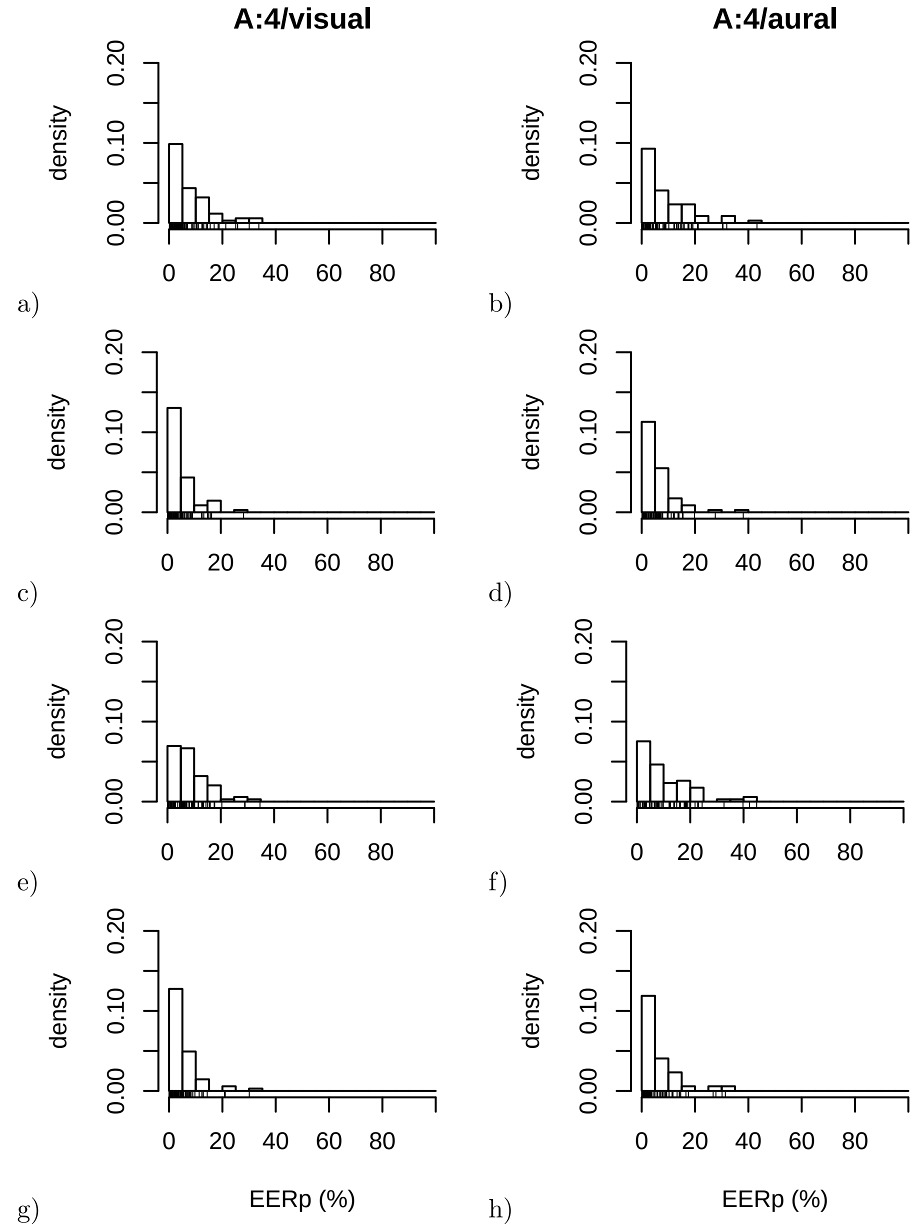


Figure D.4: Distribution of EERp (target-dependent *a posteriori* thresholds) over targets in Experiment A, with four ASV systems: CAVE system (a,b); HMM subsystem (c,d); GMM subsystem (e,f); and the combined HMM and GMM system (g,h).

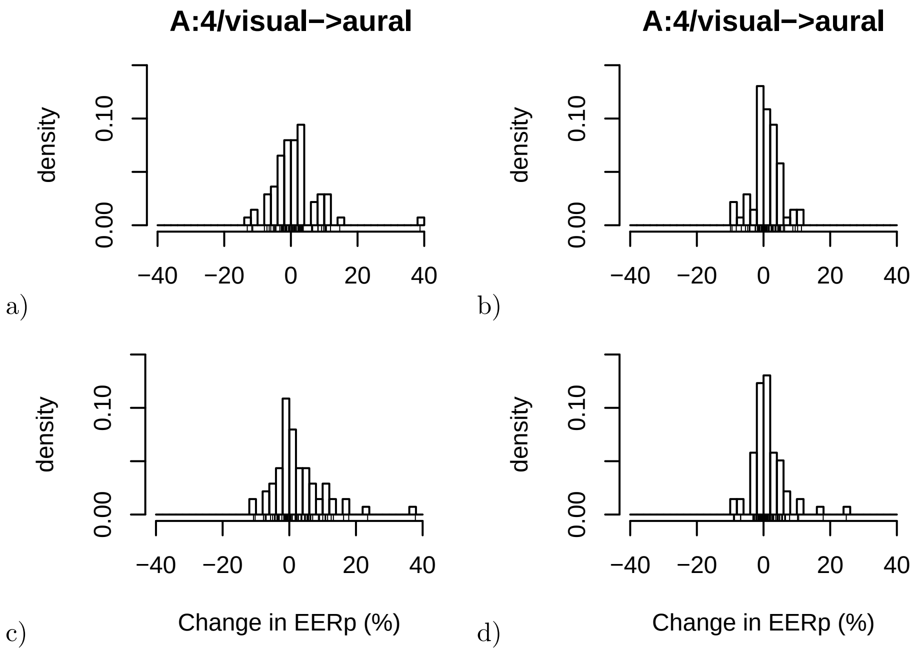


Figure D.5: Distribution of differences in EERp (target-dependent *a posteriori* thresholds) over targets when comparing the two sets in Experiment A. Results are given for four ASV systems: a) CAVE system, b) HMM subsystem, c) GMM subsystem, and d) the combined HMM and GMM system. A positive difference x indicates that the individual EERp for aurally prompted items was x units higher than for visually prompted items.

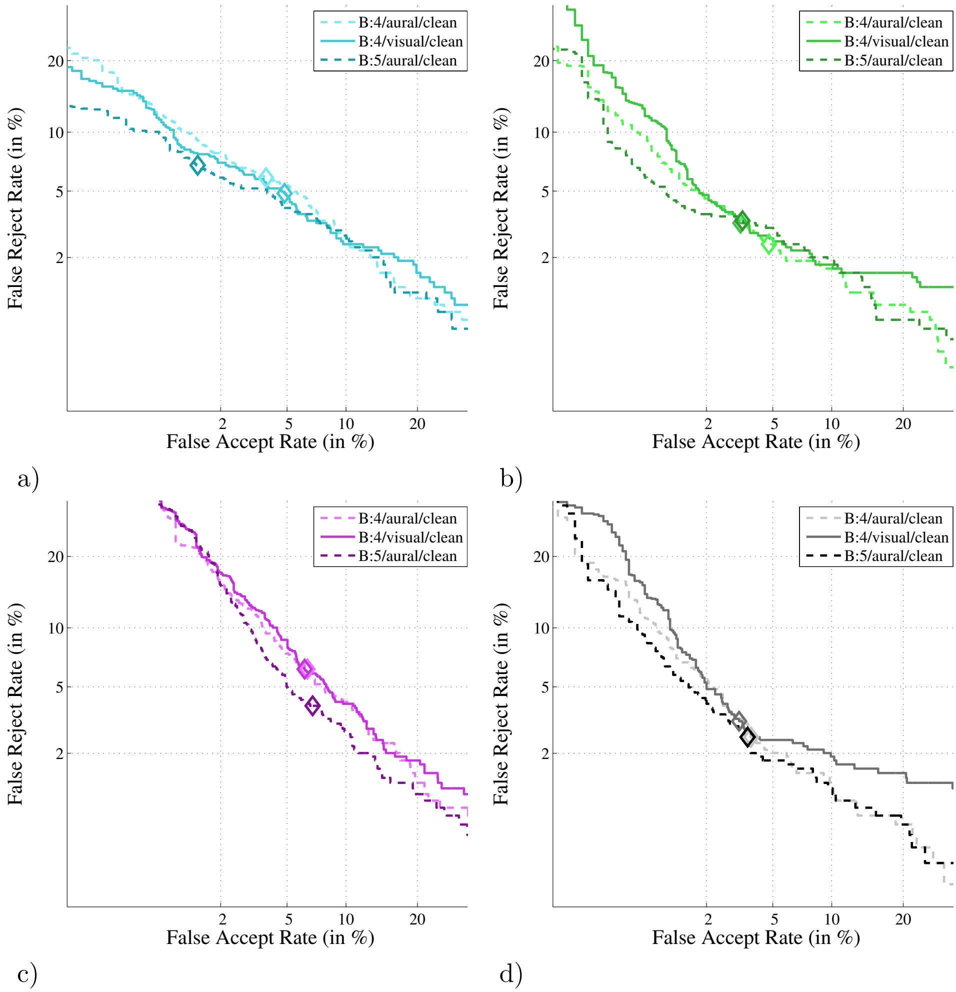


Figure D.6: DET curves from Experiment B/clean in Chapter 8, i.e. with no speaking and recording errors in the true-speaker part of test sets, for a) the HMM-based CAVE system used in (Lindberg and Melin, 1997), b) the HMM subsystem, c) the GMM subsystem and d) the combined HMM and GMM system. The three latter systems are all described in Chapter 3. Each plot compares a triple of test sets with 4-digit visually prompted and 4 and 5-digit aurally prompted strings, respectively. Diamonds indicate the FRR/FAR pair for a threshold determined as the EER threshold on the visually prompted 4-digit data.

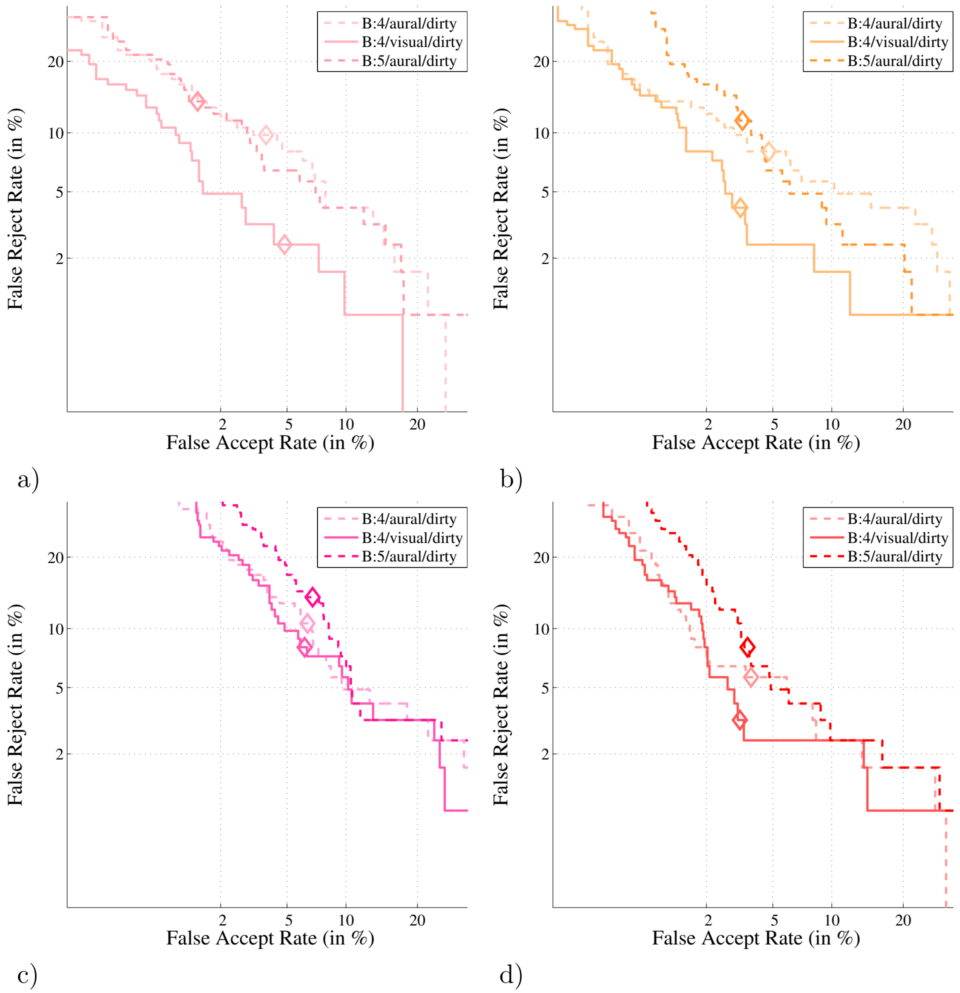


Figure D.7: Same comparisons as in the previous figure but with the complementary B/dirty test set, i.e. true-speaker parts of test sets contain only triples with at least one speaking and recording error. Diamonds indicate the FRR/FAR pair for a threshold determined as the EER threshold on the visually prompted 4-digit data in B/clean (with speaking and recording errors *excluded*).

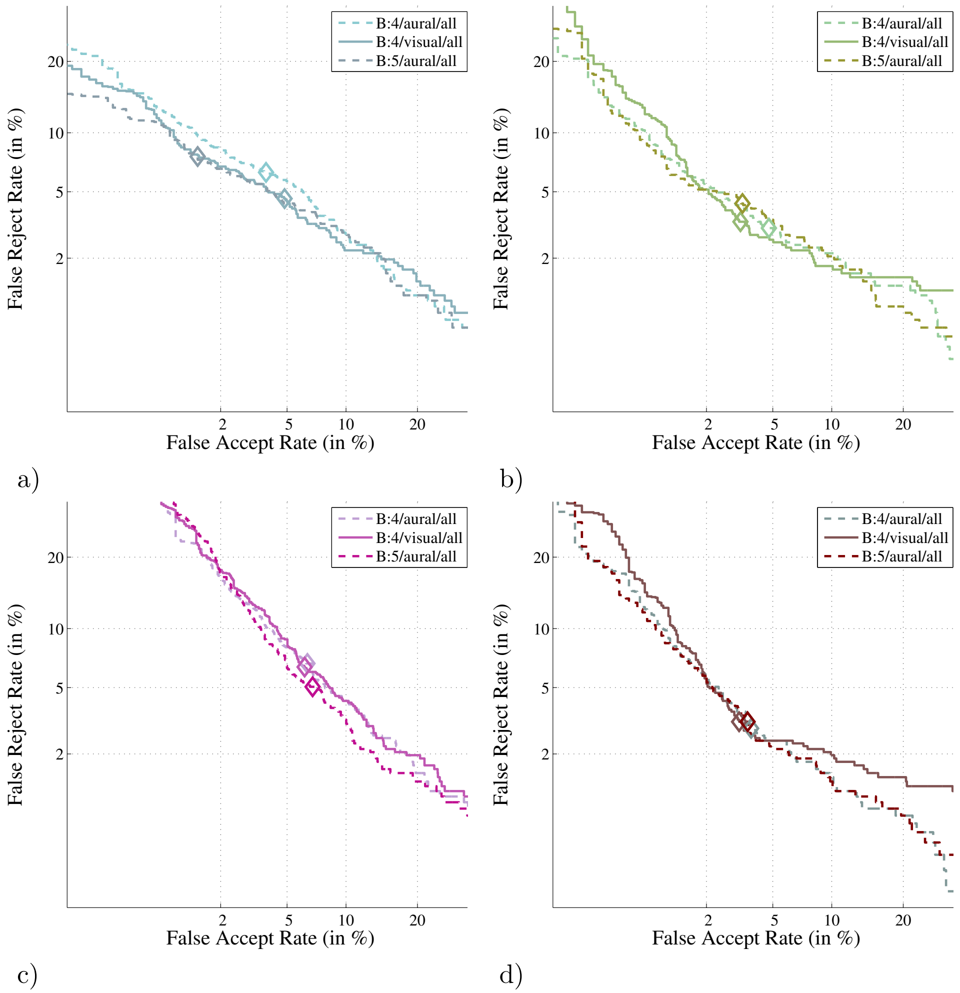
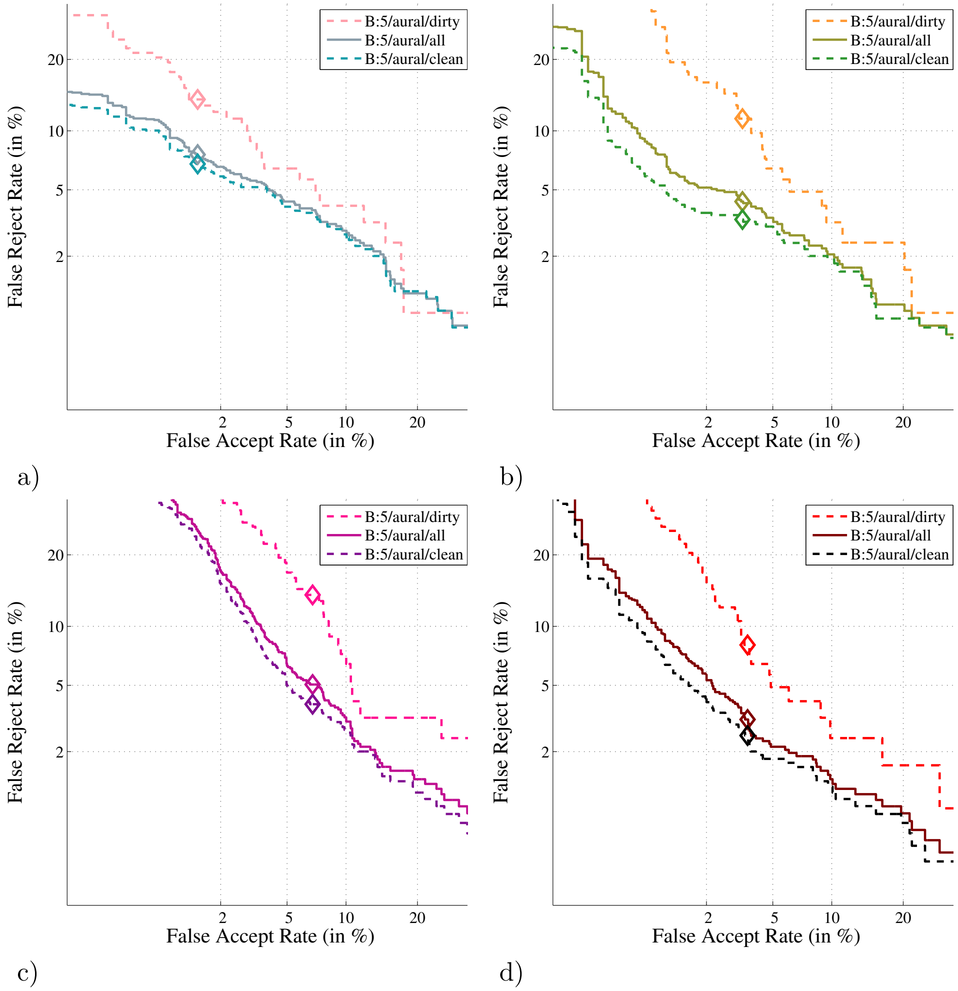


Figure D.8: Same comparisons as in the two previous figures but with corresponding B/clean and B/dirty test sets pooled, i.e. true-speaker parts of test sets contain a natural blend of speaking and recording errors. Diamonds indicate the FRR/FAR pair for a threshold determined as the EER threshold on the visually prompted 4-digit data in B/clean (with speaking and recording errors *excluded*).



a)

b)

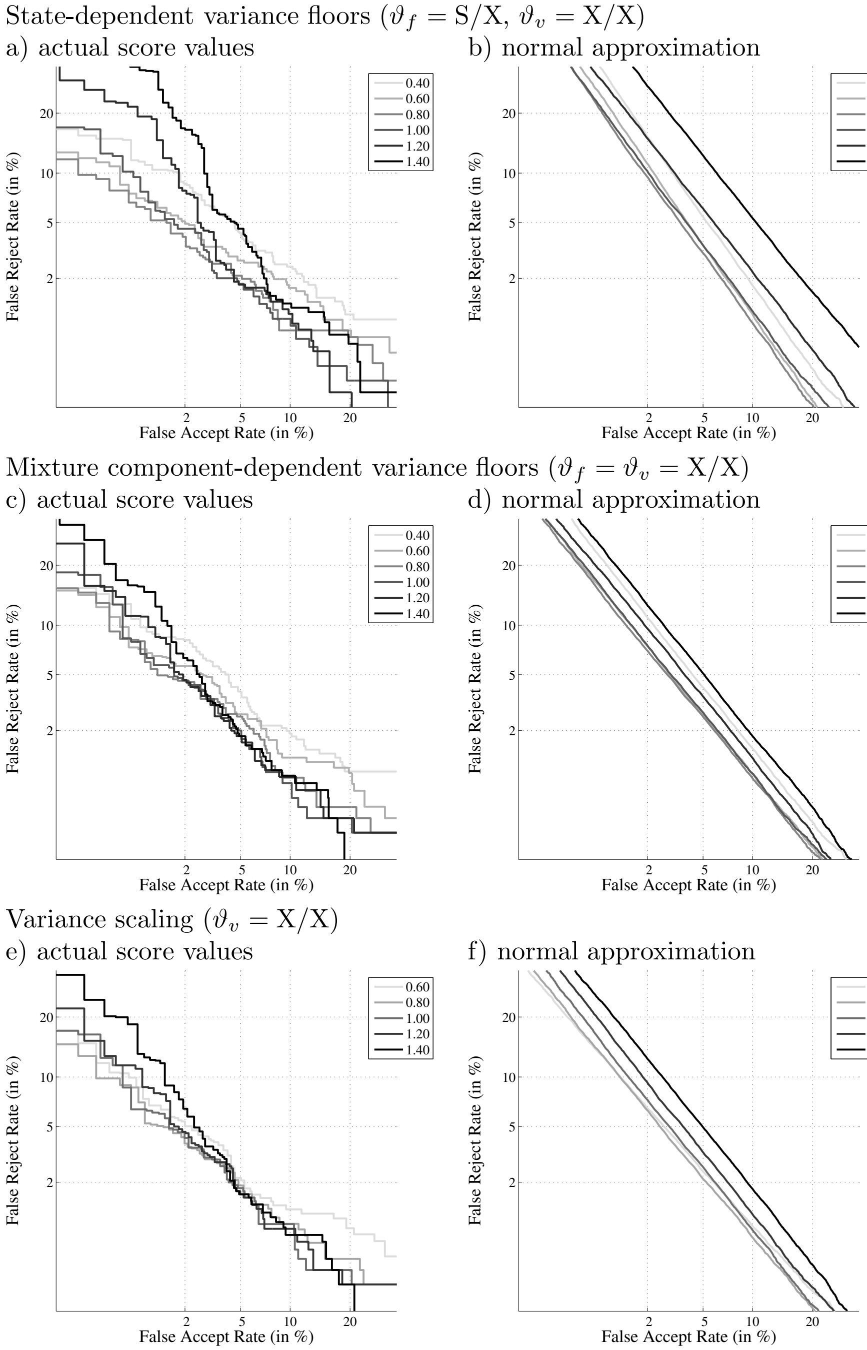
c)

d)

Figure D.9: DET curves for aurally prompted 5-digit sequences from the 'clean' and 'dirty' group of test sets in Experiment B in Chapter 8, and for the two groups pooled. The B/dirty test set is comprised of the aurally prompted 5-digits file in the 123 triples of files where at least one of the files in the triple has at least one speaking or recording error (SRE). The B/clean test set is comprised of the corresponding file in all other triples from Experiment B, i.e. those where none of the files in the triple contain an SRE. Panes show result for a) the HMM-based CAVE system used in (Lindberg and Melin, 1997), b) the HMM subsystem, c) the GMM subsystem and d) the combined HMM and GMM system. The three latter systems are all described in Chapter 3. Diamonds indicate the FRR/FAR pair for a threshold determined on as the EER threshold on the visually prompted *4-digit* data in B/clean (with speaking and recording errors *excluded*).

Appendix E

Variance estimation - extended results



State-dependent variance floors ($\vartheta_f = \text{S/X}$, $\vartheta_v = \text{X/X}$)

a) actual score values

b) normal approximation

Mixture component-dependent variance floors ($\vartheta_f = \vartheta_v = \text{X/X}$)

c) actual score values

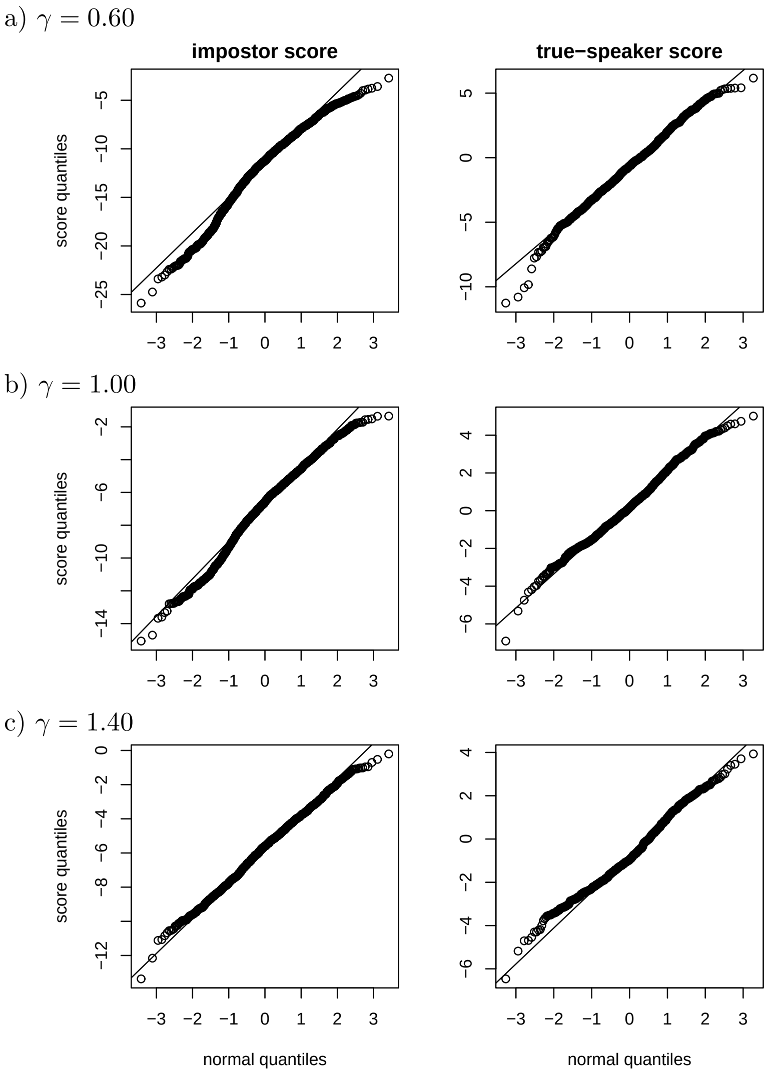
d) normal approximation

Variance scaling ($\vartheta_v = \text{X/X}$)

e) actual score values

f) normal approximation

Figure E.1: Gandalf development data: DET curves from variance flooring and variance scaling experiments in Chapter 9. Plots on the left (a,c,e) correspond to the "gand, dev" plots in Figures 9.1b, 9.1c and 9.1f, respectively. Plots on the right (b,d,f) show DET curves of synthetic score data generated from normal distributions estimated from curves on the left.



a) $\gamma = 0.60$

b) $\gamma = 1.00$

c) $\gamma = 1.40$

Figure E.2: Gandalf development data: normal quantile plots for *state-dependent variance floors* (ϑ_f = S/X, ϑ_v = X/X) and three scale factors. Data correspond to the "gand, dev" plot in Figure 9.1b (p. 182) and Figures E.1a,b.

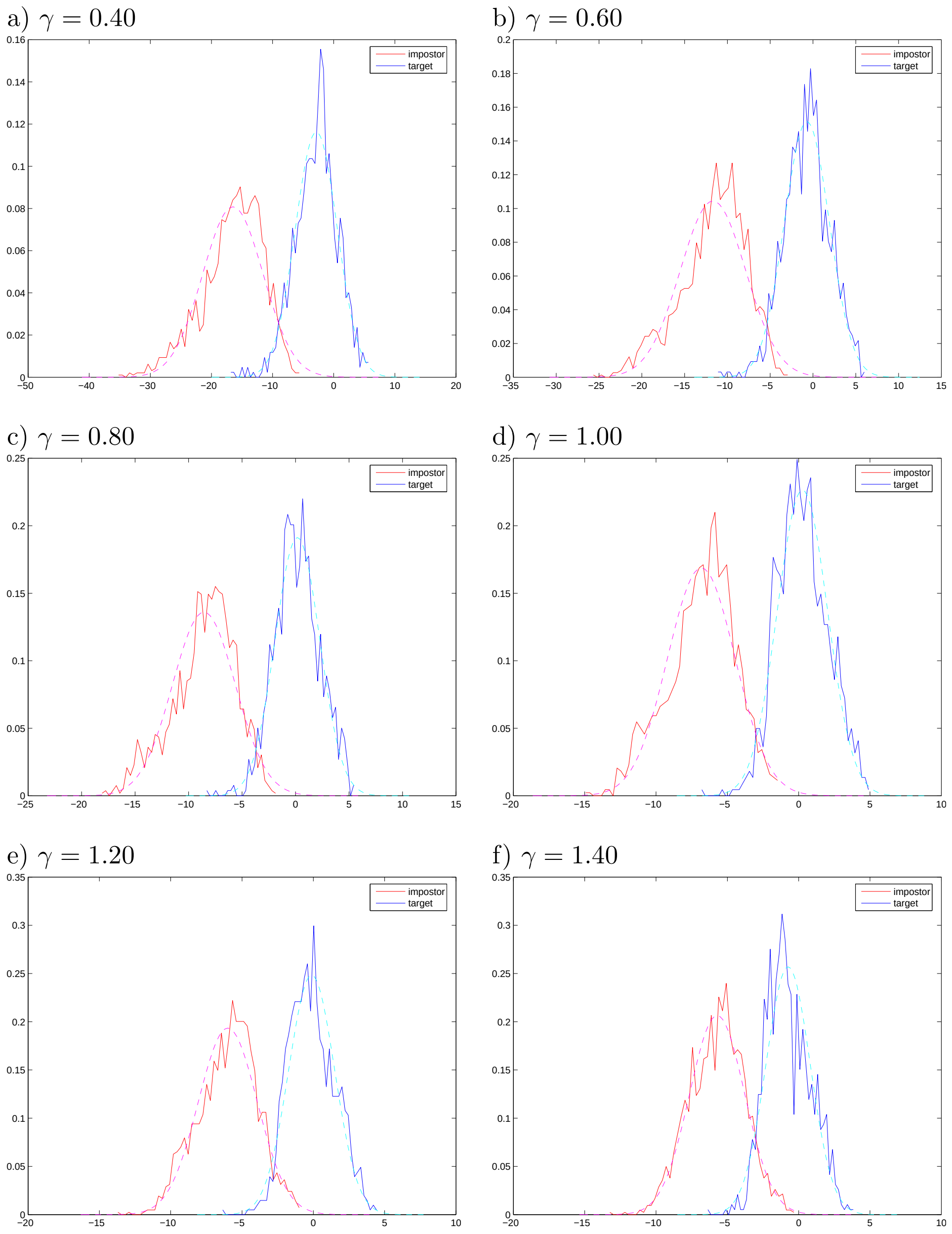
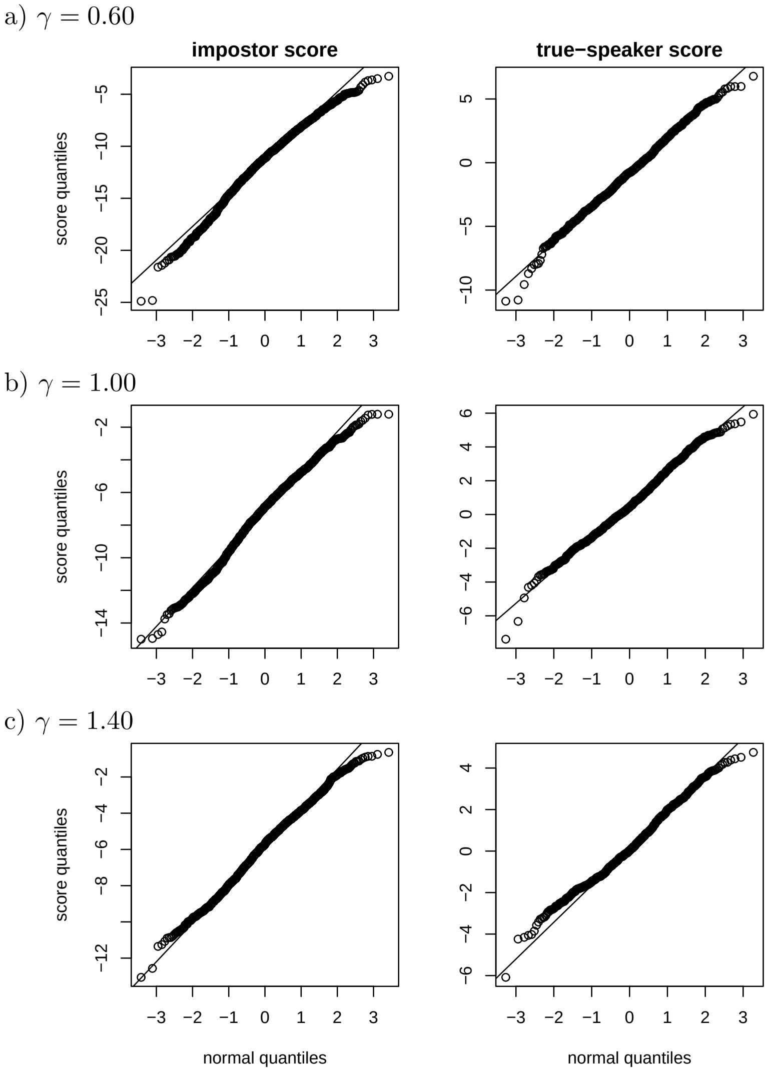


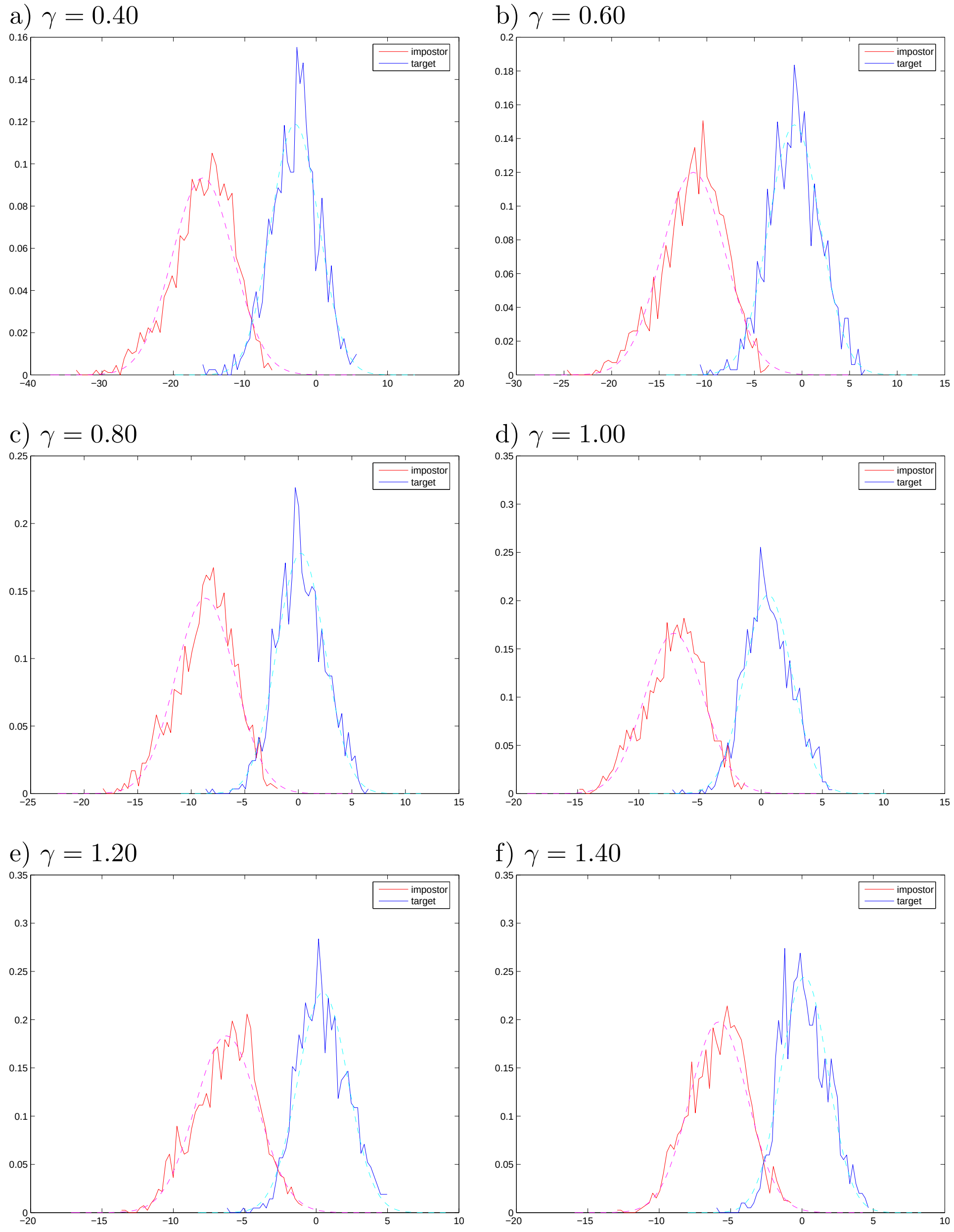
Figure E.3: Gandalf development data: Score histograms for *state-dependent variance floors* ($\vartheta_f = \mathrm{S/X}$, $\vartheta_v = \mathrm{X/X}$) and a range of scale factors. Data correspond to the "gand, dev" plot in Figure 9.1b (p. 182) and Figures E.1a,b. Dashed lines are normal distributions estimated from the actual score values.


a) $\gamma = 0.60$

b) $\gamma = 1.00$

c) $\gamma = 1.40$

Figure E.4: Gandalf development data: normal quantile plots for *mixture component-dependent variance floors* (ϑ_f = X/X, ϑ_v = X/X) and three scale factors. Data correspond to the “gand, dev” plot in Figure 9.1c (p. 182) and Figures E.1c,d.



a) $\gamma = 0.40$

b) $\gamma = 0.60$

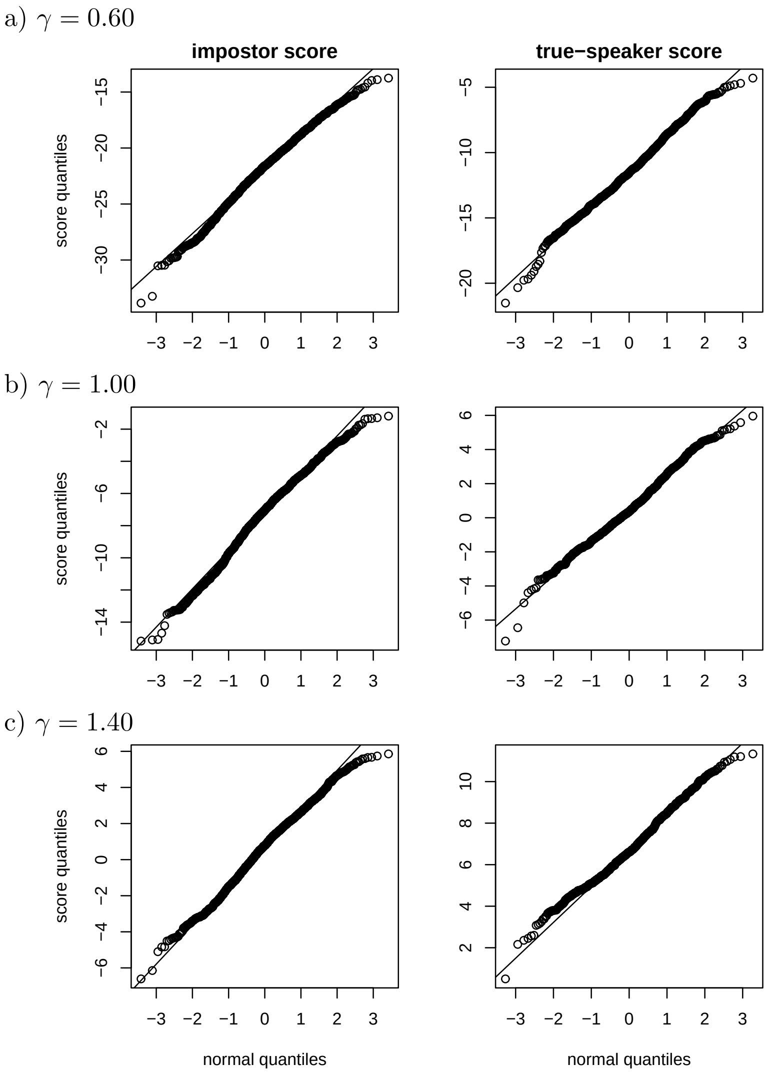
c) $\gamma = 0.80$

d) $\gamma = 1.00$

e) $\gamma = 1.20$

f) $\gamma = 1.40$

Figure E.5: Gandalf development data: Score histograms for *mixture component-dependent variance floors* ($\vartheta_f = \vartheta_v = \text{X/X}$) and a range of scale factors. Data correspond to the "gand, dev" plot in Figure 9.1c (p. 182) and Figures E.1c,d. Dashed lines are normal distributions estimated from the actual score values.

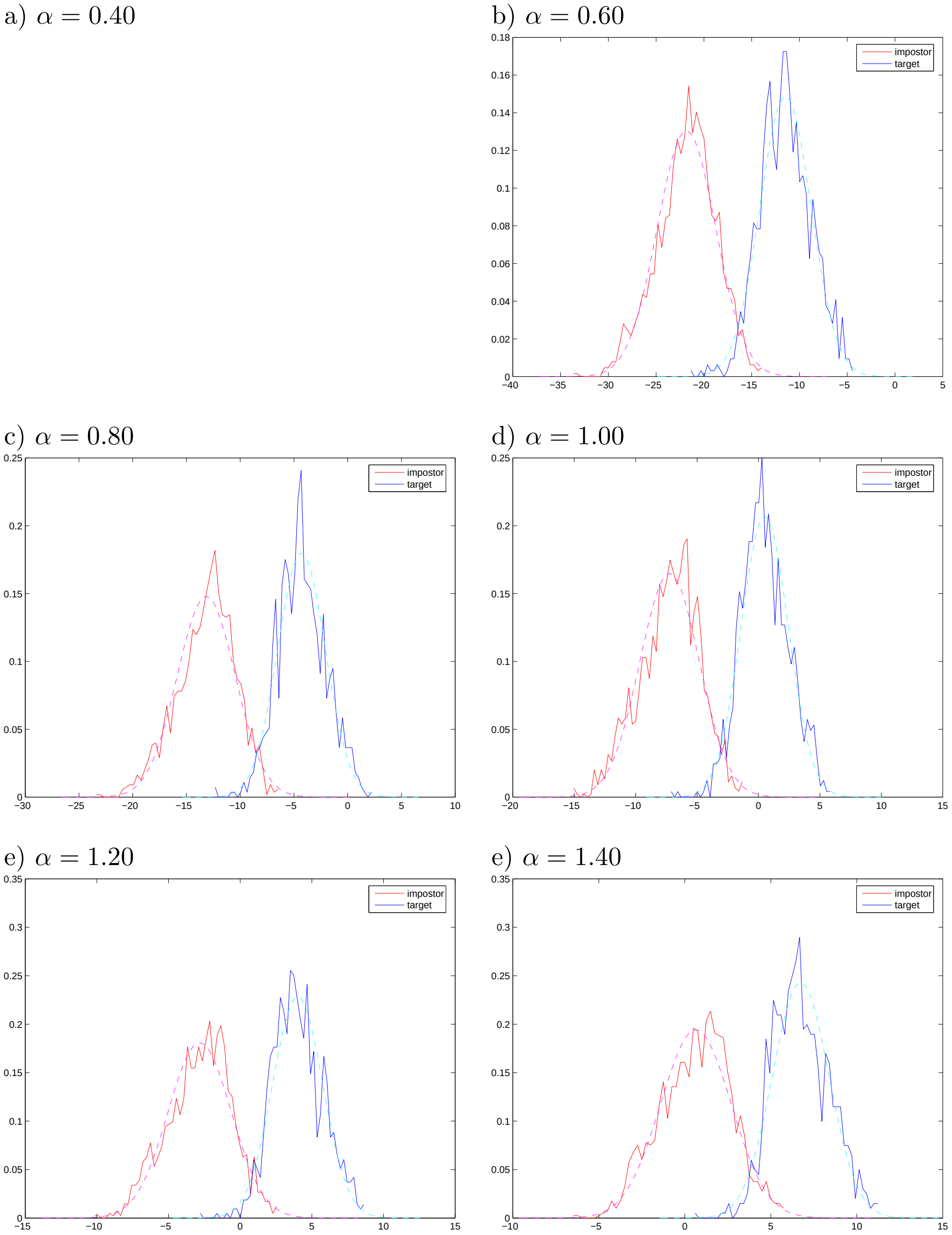


a) $\gamma = 0.60$

b) $\gamma = 1.00$

c) $\gamma = 1.40$

Figure E.6: Gandalf development data: normal quantile plots for *variance scaling* with untied variances ($\vartheta_v = \mathrm{X/X}$) and three scale factors. Data correspond to the "gand, dev" plot in Figure 9.1f (p. 182) and Figures E.1e,f.


a) $\alpha = 0.40$

b) $\alpha = 0.60$

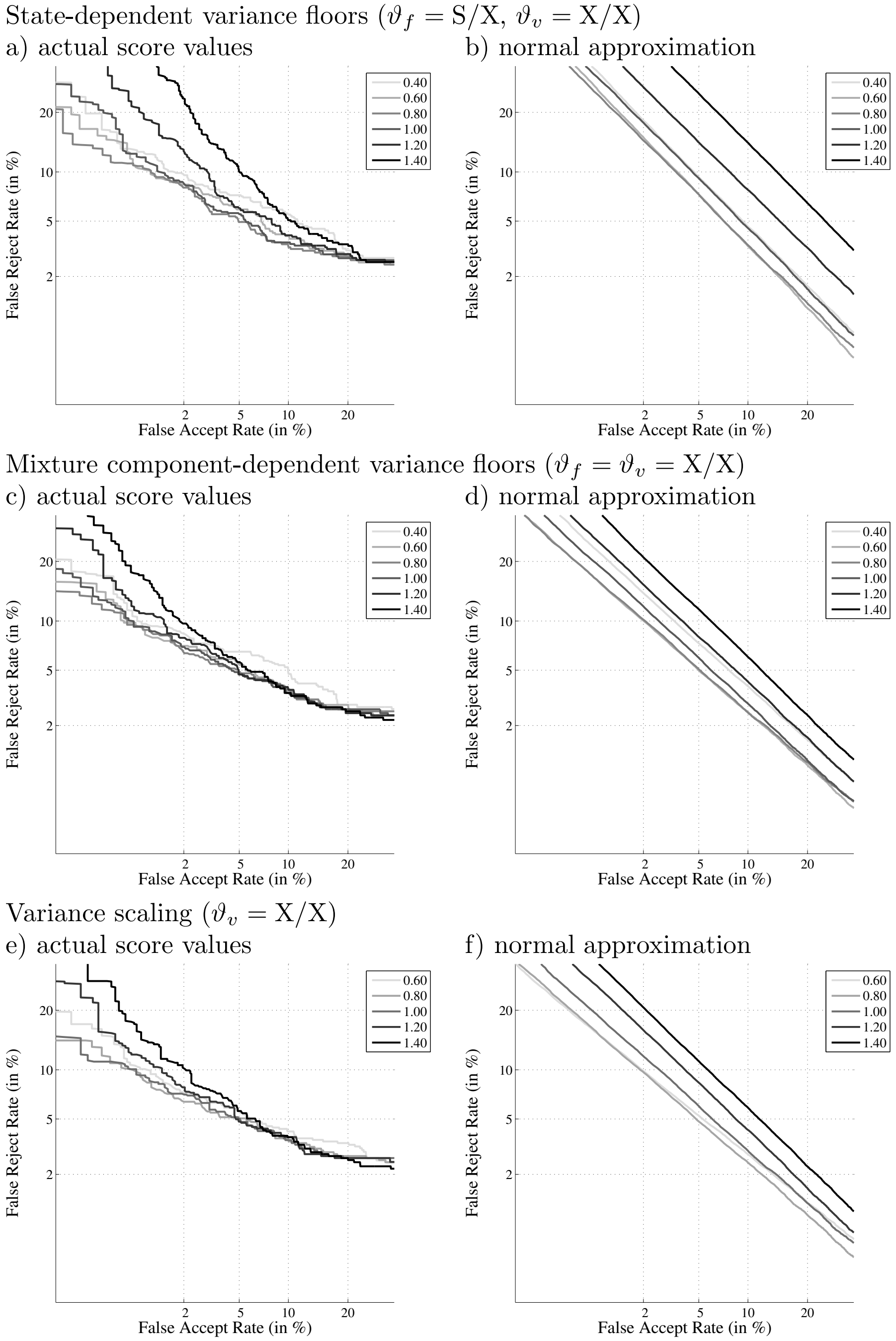
c) $\alpha = 0.80$

d) $\alpha = 1.00$

e) $\alpha = 1.20$

e) $\alpha = 1.40$

Figure E.7: Gandalf development data: Score histograms for *variance scaling* with untied variances ($\vartheta_v = \text{X/X}$) and a range of scale factors. Data correspond to the "gand, dev" plot in Figure 9.1f (p. 182) and Figures E.1e,f. Dashed lines are normal distributions estimated from the actual score values.



State-dependent variance floors ($\vartheta_f = \mathrm{S/X}$, $\vartheta_v = \mathrm{X/X}$)

a) actual score values

b) normal approximation

Mixture component-dependent variance floors ($\vartheta_f = \vartheta_v = \mathrm{X/X}$)

c) actual score values

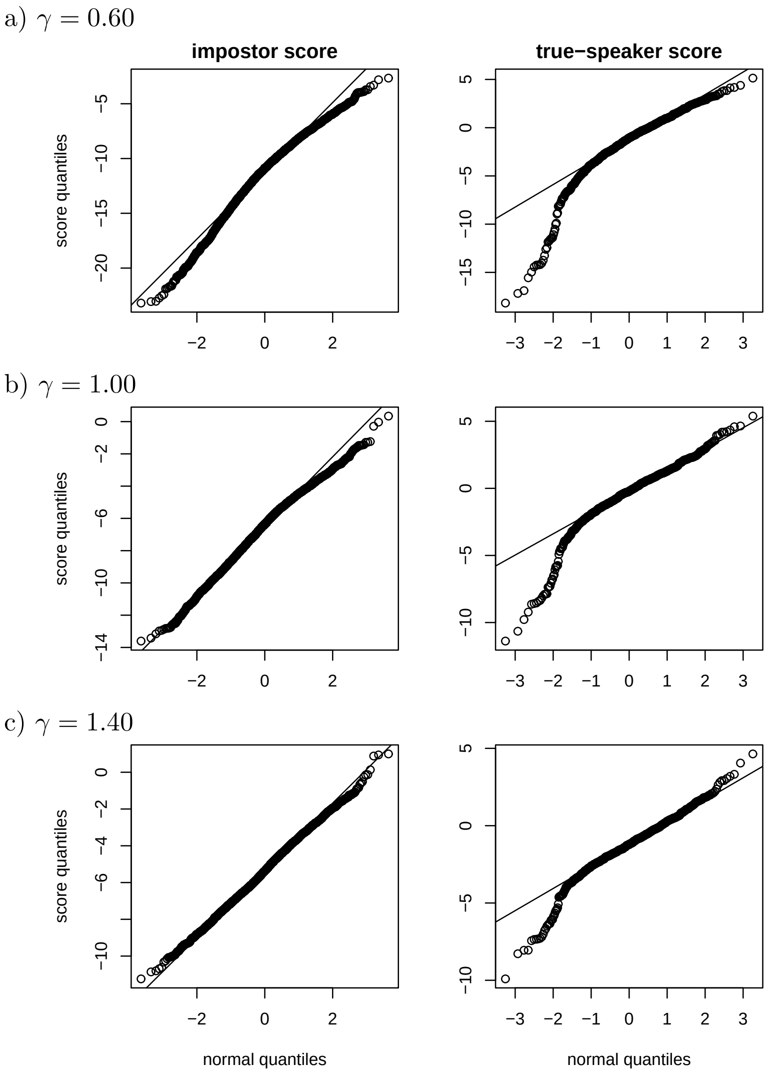
d) normal approximation

Variance scaling ($\vartheta_v = \mathrm{X/X}$)

e) actual score values

f) normal approximation

Figure E.8: Gandalf evaluation data: DET curves from variance flooring and variance scaling experiments in Chapter 9. Plots on the left (a,c,e) correspond to the "gand, eva" plots in Figures 9.1b, 9.1c and 9.1f, respectively. Plots on the right (b,d,f) show DET curves of synthetic score data generated from normal distributions estimated from curves on the left.


a) $\gamma = 0.60$

b) $\gamma = 1.00$

c) $\gamma = 1.40$

Figure E.9: Gandalf evaluation data: normal quantile plots for *state-dependent variance floors* (ϑ_f = S/X, ϑ_v = X/X) and three scale factors. Data correspond to the "gand, eva" plot in Figure 9.1b (p. 182) and Figures E.8a,b.

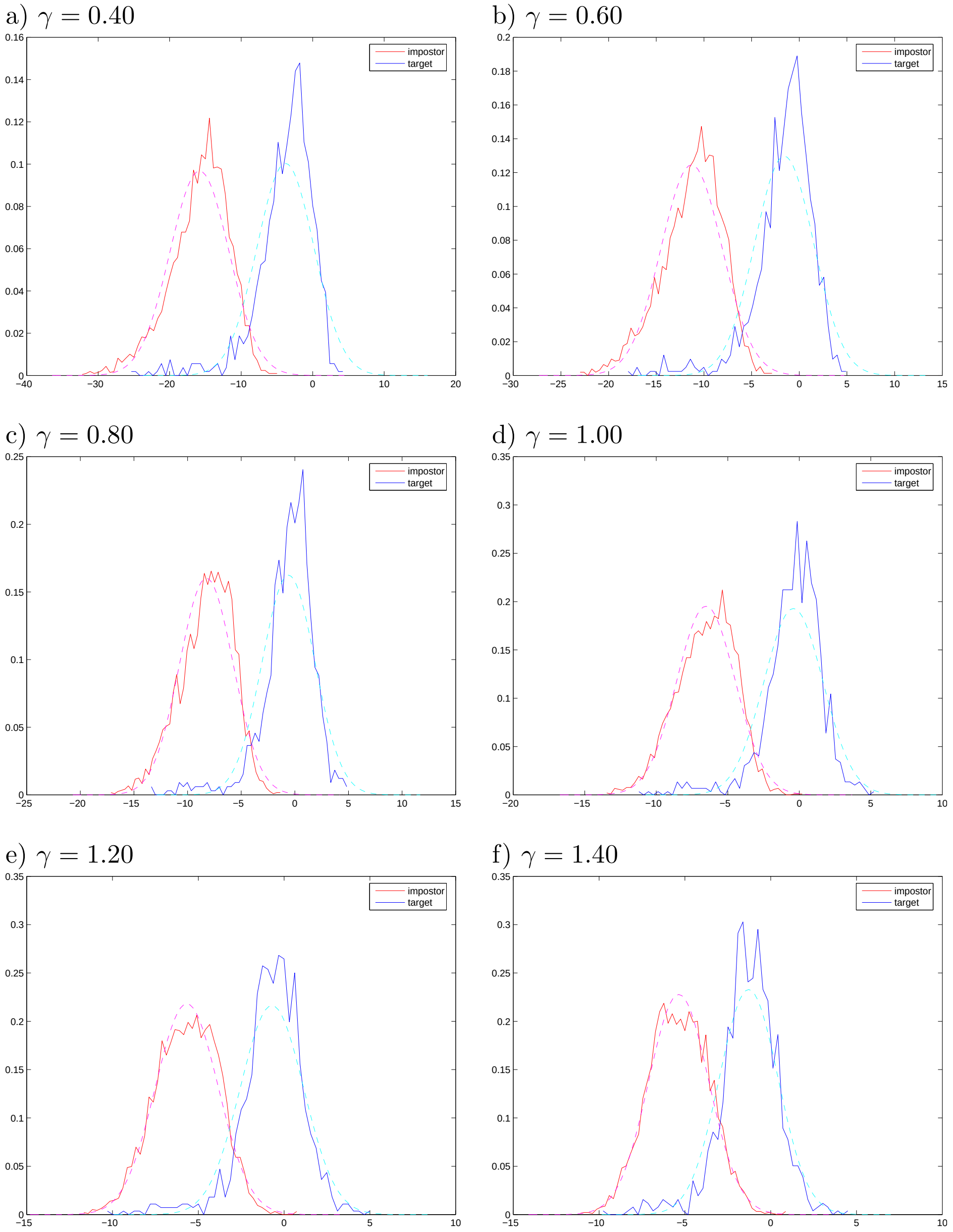
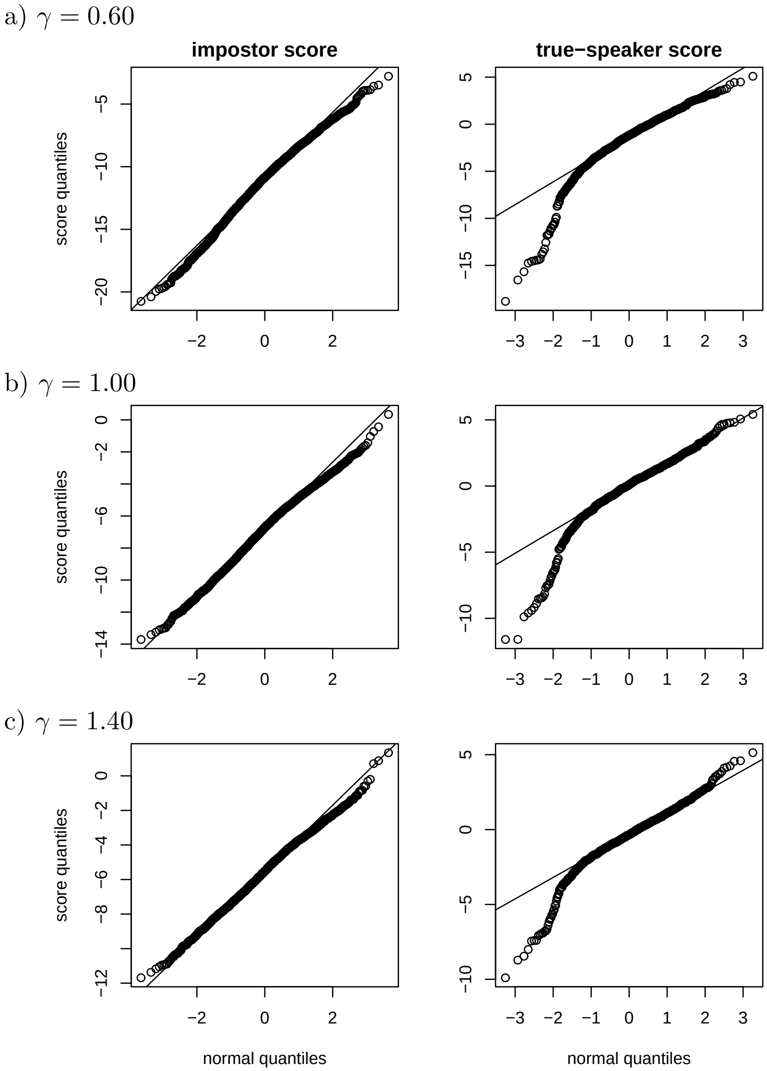


Figure E.10: Gandalf evaluation data: Score histograms for *state-dependent variance floors* (ϑ_f = S/X, ϑ_v = X/X) and a range of scale factors. Data correspond to the "gand, eva" plot in Figure 9.1b (p. 182) and Figures E.8a,b. Dashed lines are normal distributions estimated from the actual score values.

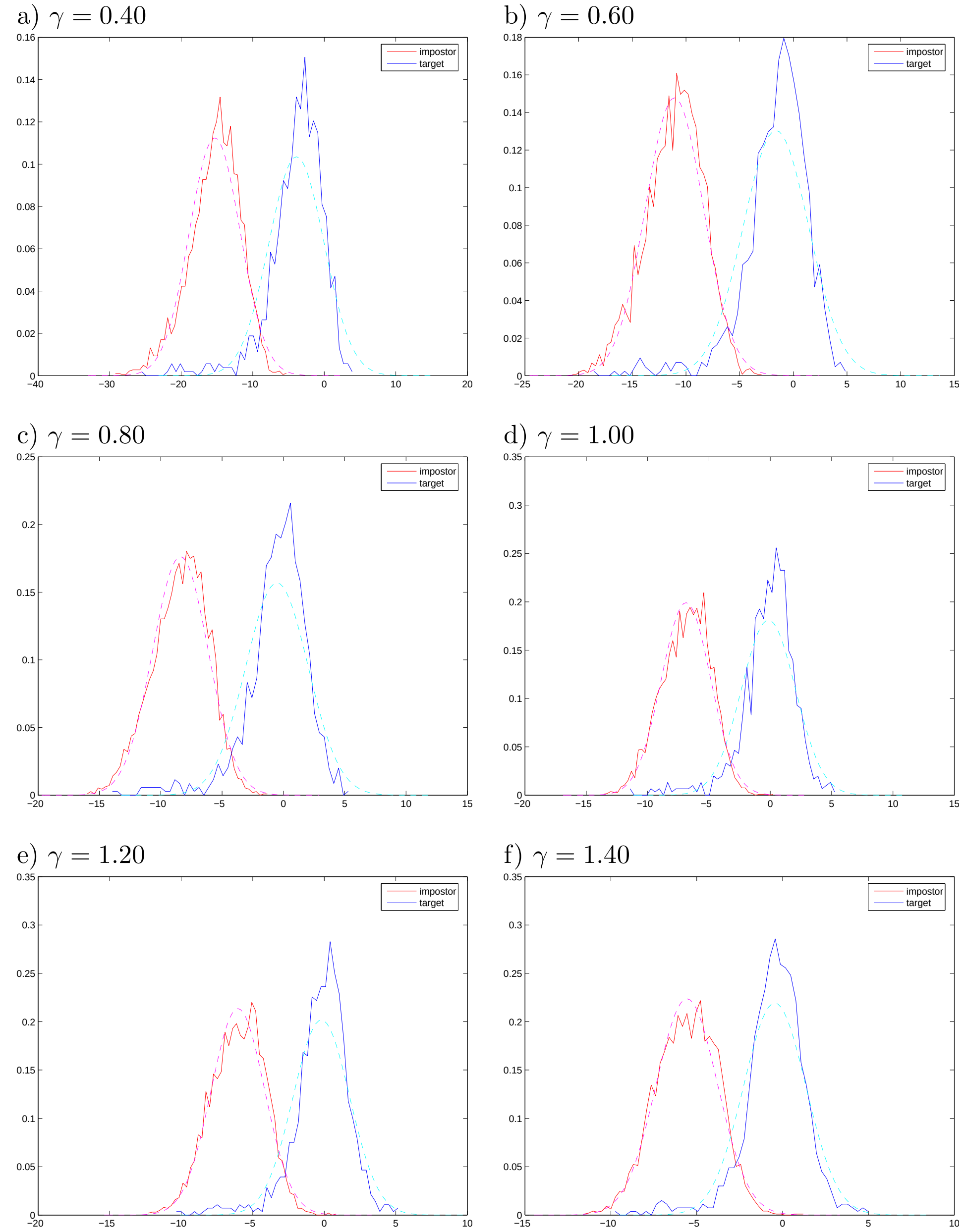


a) $\gamma = 0.60$

b) $\gamma = 1.00$

c) $\gamma = 1.40$

Figure E.11: Gandalf evaluation data: normal quantile plots for *mixture component-dependent variance floors* (ϑ_f = X/X, ϑ_v = X/X) and three scale factors. Data correspond to the "gand, eva" plot in Figure 9.1c (p. 182) and Figures E.8c,d.



a) $\gamma = 0.40$

b) $\gamma = 0.60$

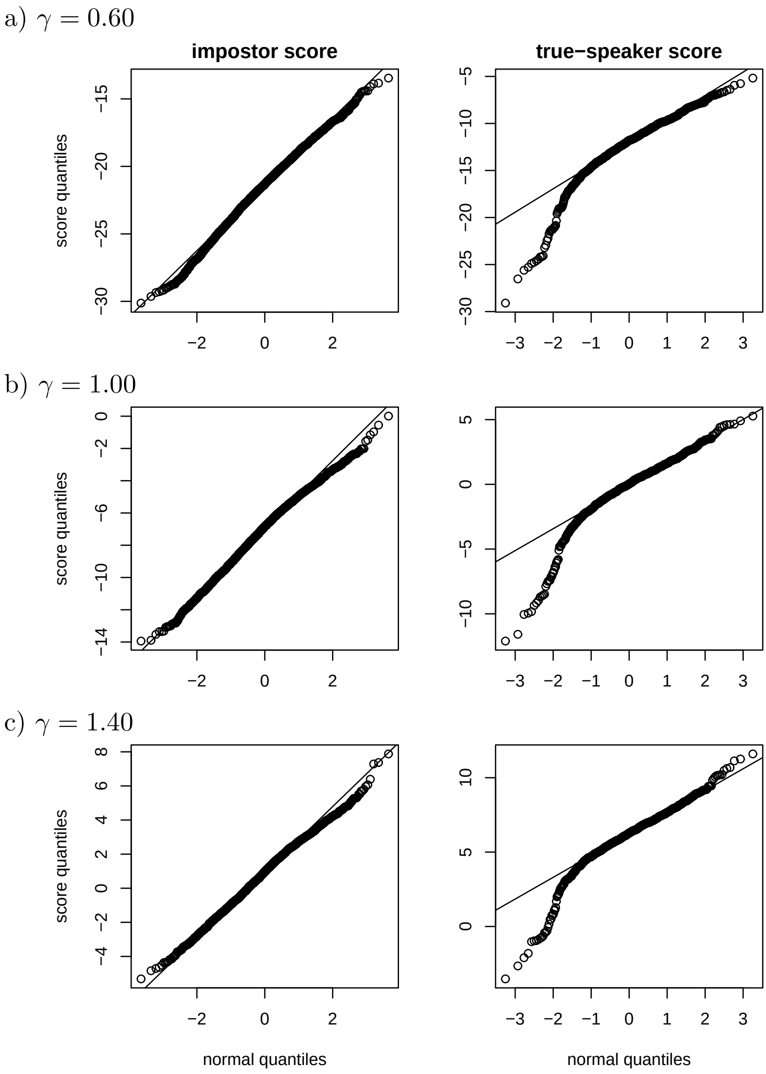
c) $\gamma = 0.80$

d) $\gamma = 1.00$

e) $\gamma = 1.20$

f) $\gamma = 1.40$

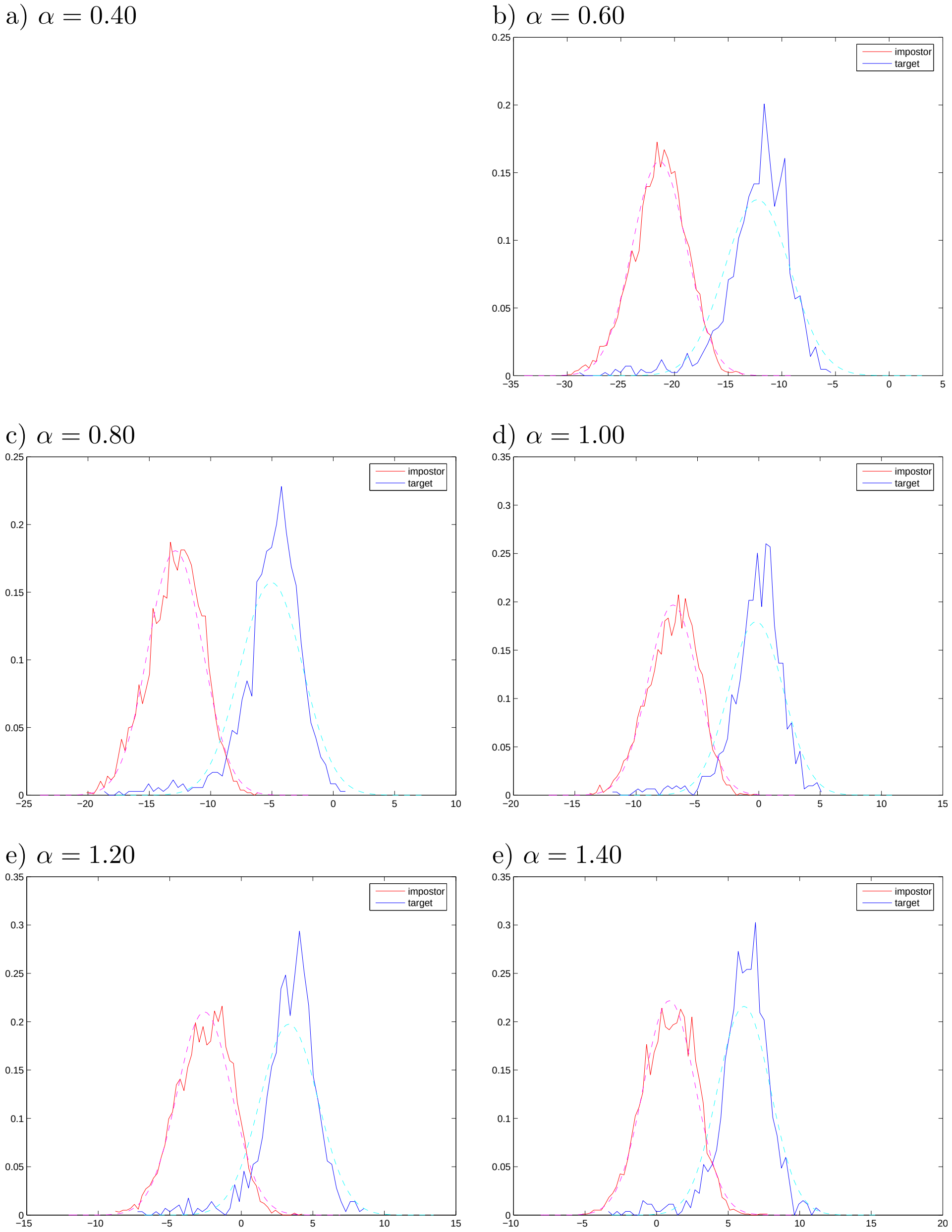
Figure E.12: Gandalf evaluation data: Score histograms for *mixture component-dependent variance floors* ($\vartheta_f = \vartheta_v =$ X/X) and a range of scale factors. Data correspond to the "gand, eva" plot in Figure 9.1c (p. 182) and Figures E.8c,d. Dashed lines are normal distributions estimated from the actual score values.


a) $\gamma = 0.60$

b) $\gamma = 1.00$

c) $\gamma = 1.40$

Figure E.13: Gandalf evaluation data: normal quantile plots for *variance scaling* with untied variances ($\vartheta_v = \mathrm{X/X}$) and three scale factors. Data correspond to the "gand, eva" plot in Figure 9.1f (p. 182) and Figures E.8e,f.



a) $\alpha = 0.40$

b) $\alpha = 0.60$

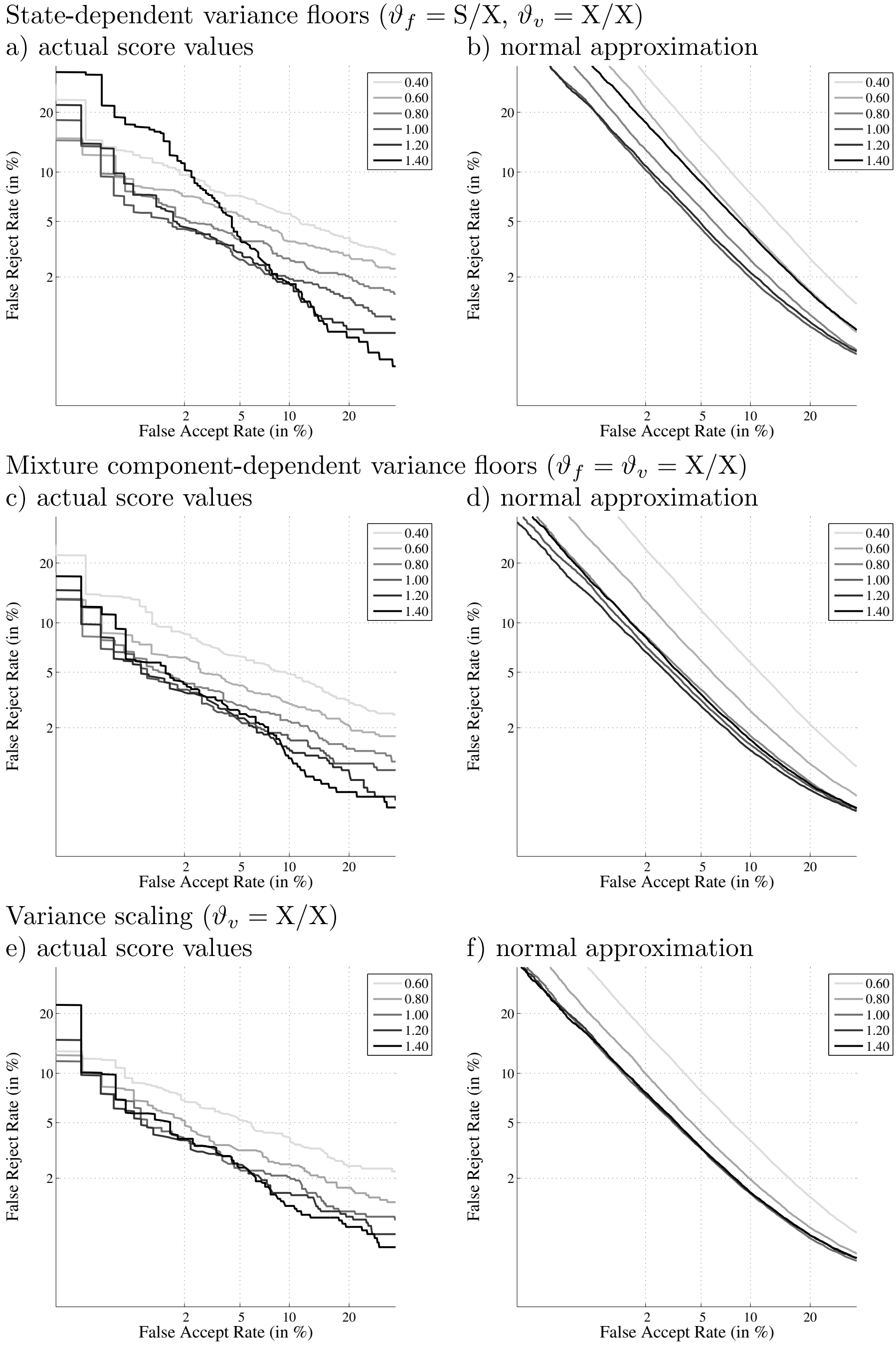
c) $\alpha = 0.80$

d) $\alpha = 1.00$

e) $\alpha = 1.20$

e) $\alpha = 1.40$

Figure E.14: Gandalf evaluation data: Score histograms for *variance scaling* with untied variances (ϑ_v = X/X) and a range of scale factors. Data correspond to the “gand, eva” plot in Figure 9.1f (p. 182) and Figures E.8e,f. Dashed lines are normal distributions estimated from the actual score values.


State-dependent variance floors ($\vartheta_f = \text{S/X}$, $\vartheta_v = \text{X/X}$)

a) actual score values

b) normal approximation

Mixture component-dependent variance floors ($\vartheta_f = \vartheta_v = \text{X/X}$)

c) actual score values

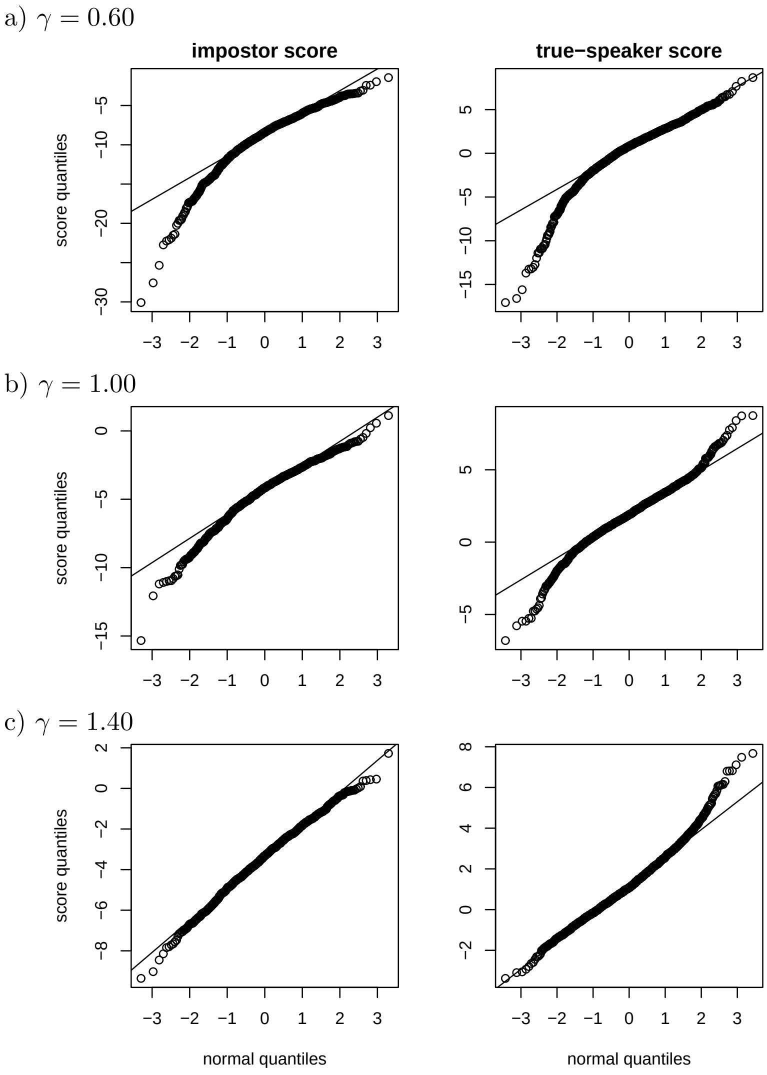
d) normal approximation

Variance scaling ($\vartheta_v = \text{X/X}$)

e) actual score values

f) normal approximation

Figure E.15: SESP data: DET curves from variance flooring and variance scaling experiments in Chapter 9. Plots on the left (a,c,e) correspond to the "sesp" plots in Figures 9.1b, 9.1c and 9.1f, respectively. Plots on the right (b,d,f) show DET curves of synthetic score data generated from normal distributions estimated from curves on the left.



a) $\gamma = 0.60$

b) $\gamma = 1.00$

c) $\gamma = 1.40$

Figure E.16: SESP data: normal quantile plots for *state-dependent variance floors* (ϑ_f = S/X, ϑ_v = X/X) and three scale factors. Data correspond to the "sesp" plot in Figure 9.1b (p. 182) and Figures E.15a,b.

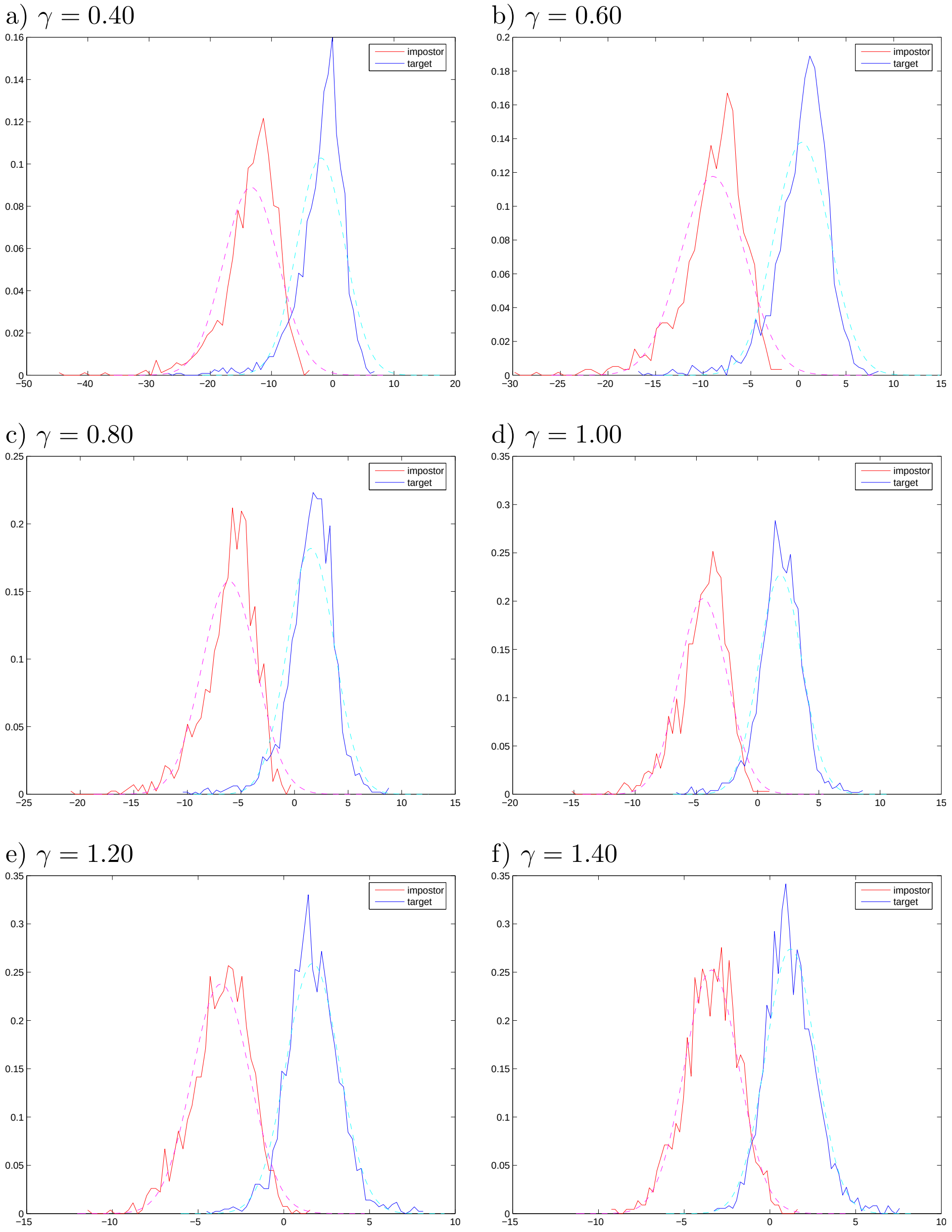
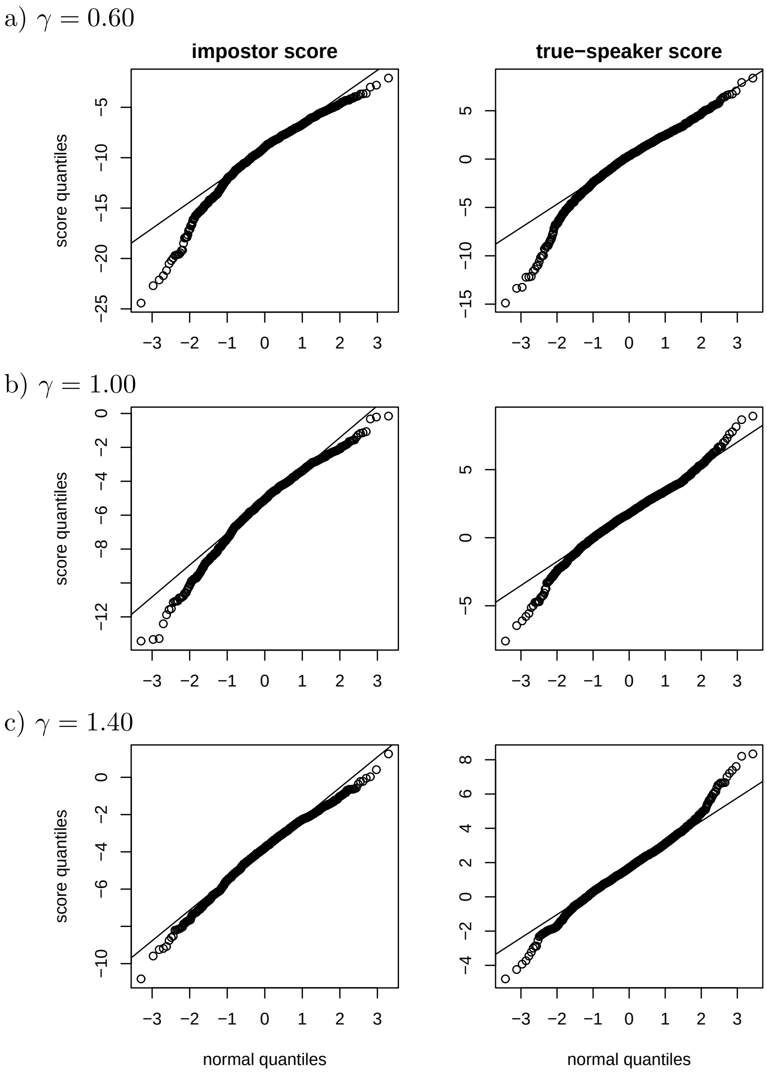


Figure E.17: SESP data: Score histograms for *state-dependent variance floors* (ϑ_f = S/X, ϑ_v = X/X) and a range of scale factors. Data correspond to the "sesp" plot in Figure 9.1b (p. 182) and Figures E.15a,b. Dashed lines are normal distributions estimated from the actual score values.



a) $\gamma = 0.60$

b) $\gamma = 1.00$

c) $\gamma = 1.40$

Figure E.18: SESP data: normal quantile plots for *mixture component-dependent variance floors* (ϑ_f = X/X, ϑ_v = X/X) and three scale factors. Data correspond to the “sesp” plot in Figure 9.1c (p. 182) and Figures E.15c,d.

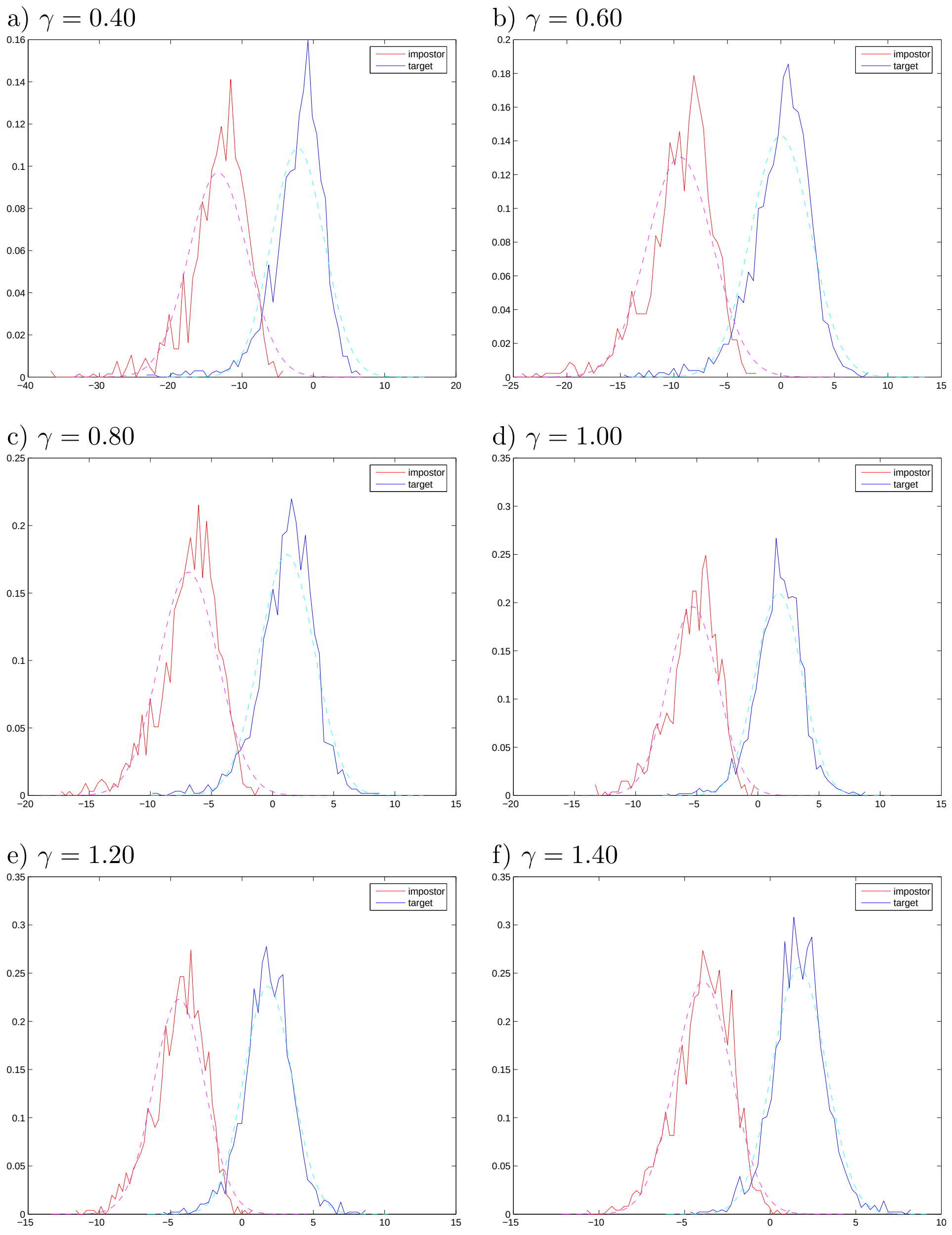
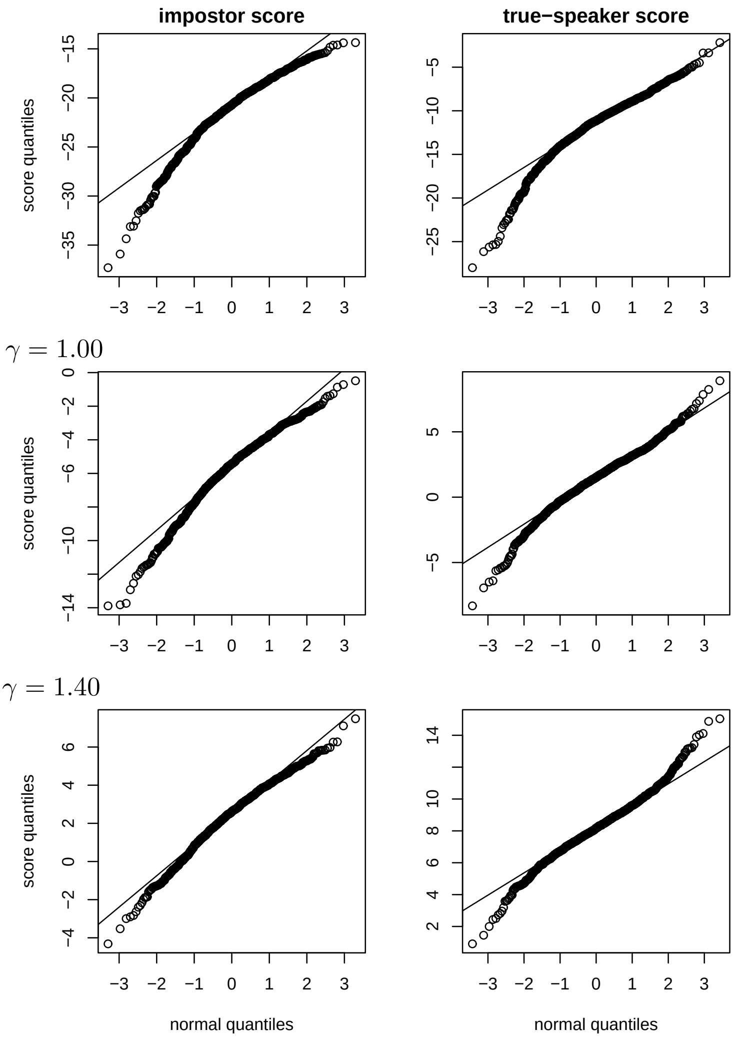


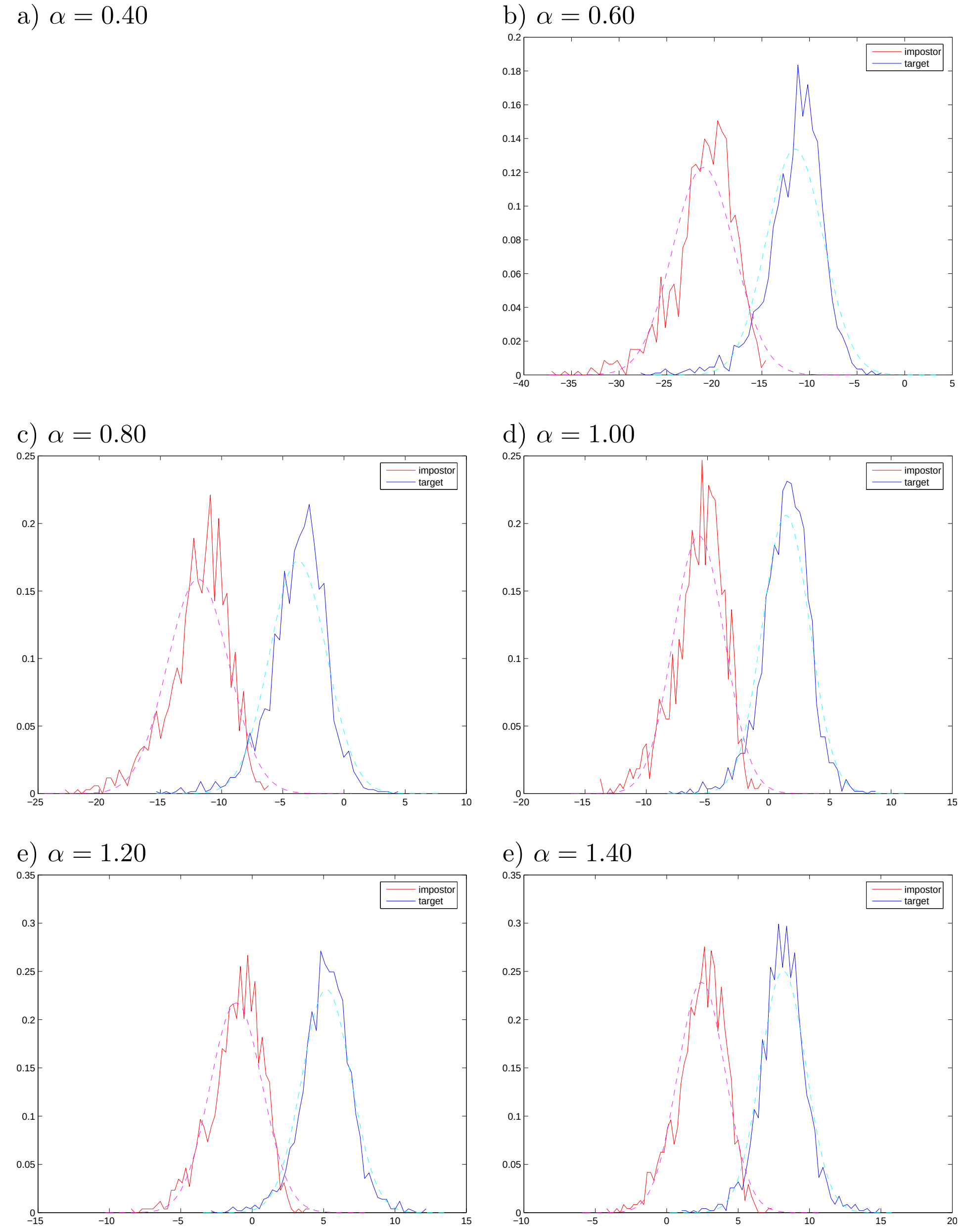
Figure E.19: SESP data: Score histograms for *mixture component-dependent variance floors* ($\vartheta_f = \vartheta_v =$ X/X) and a range of scale factors. Data correspond to the “sesp” plot in Figure 9.1c (p. 182) and Figures E.15c,d. Dashed lines are normal distributions estimated from the actual score values.


a) $\gamma = 0.60$

b) $\gamma = 1.00$

c) $\gamma = 1.40$

Figure E.20: SESP data: normal quantile plots for *variance scaling* with untied variances ($\vartheta_v = \text{X/X}$) and three scale factors. Data correspond to the "sesp" plot in Figure 9.1f (p. 182) and Figures E.15e,f.



a) $\alpha = 0.40$

b) $\alpha = 0.60$

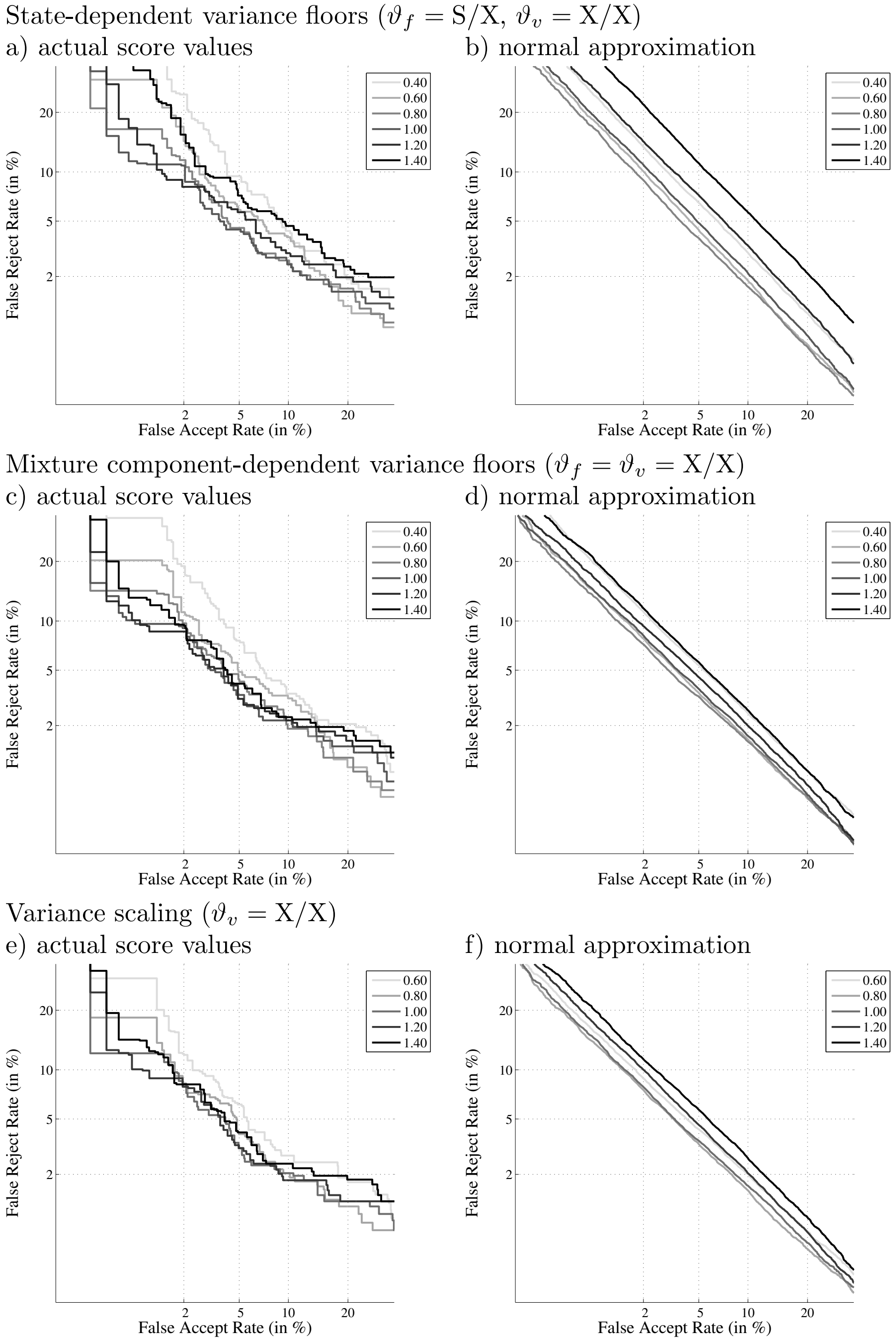
c) $\alpha = 0.80$

d) $\alpha = 1.00$

e) $\alpha = 1.20$

e) $\alpha = 1.40$

Figure E.21: SESP data: Score histograms for *variance scaling* with untied variances ($\vartheta_v = \text{X/X}$) and a range of scale factors. Data correspond to the “sesp” plot in Figure 9.1f (p. 182) and Figures E.15e,f. Dashed lines are normal distributions estimated from the actual score values.



State-dependent variance floors ($\vartheta_f = \text{S/X}$, $\vartheta_v = \text{X/X}$)

a) actual score values

b) normal approximation

Mixture component-dependent variance floors ($\vartheta_f = \vartheta_v = \text{X/X}$)

c) actual score values

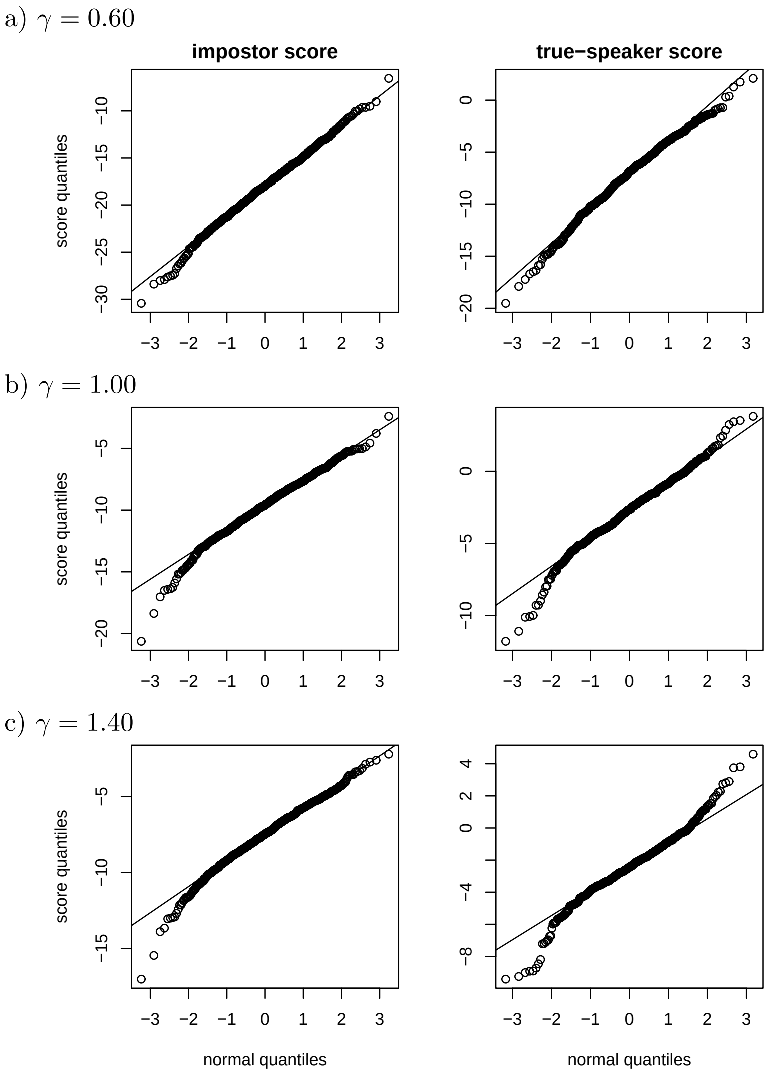
d) normal approximation

Variance scaling ($\vartheta_v = \text{X/X}$)

e) actual score values

f) normal approximation

Figure E.22: Polycost data: DET curves from variance flooring and variance scaling experiments in Chapter 9. Plots on the left (a,c,e) correspond to the "poly" plots in Figures 9.1b, 9.1c and 9.1f, respectively. Plots on the right (b,d,f) show DET curves of synthetic score data generated from normal distributions estimated from curves on the left.

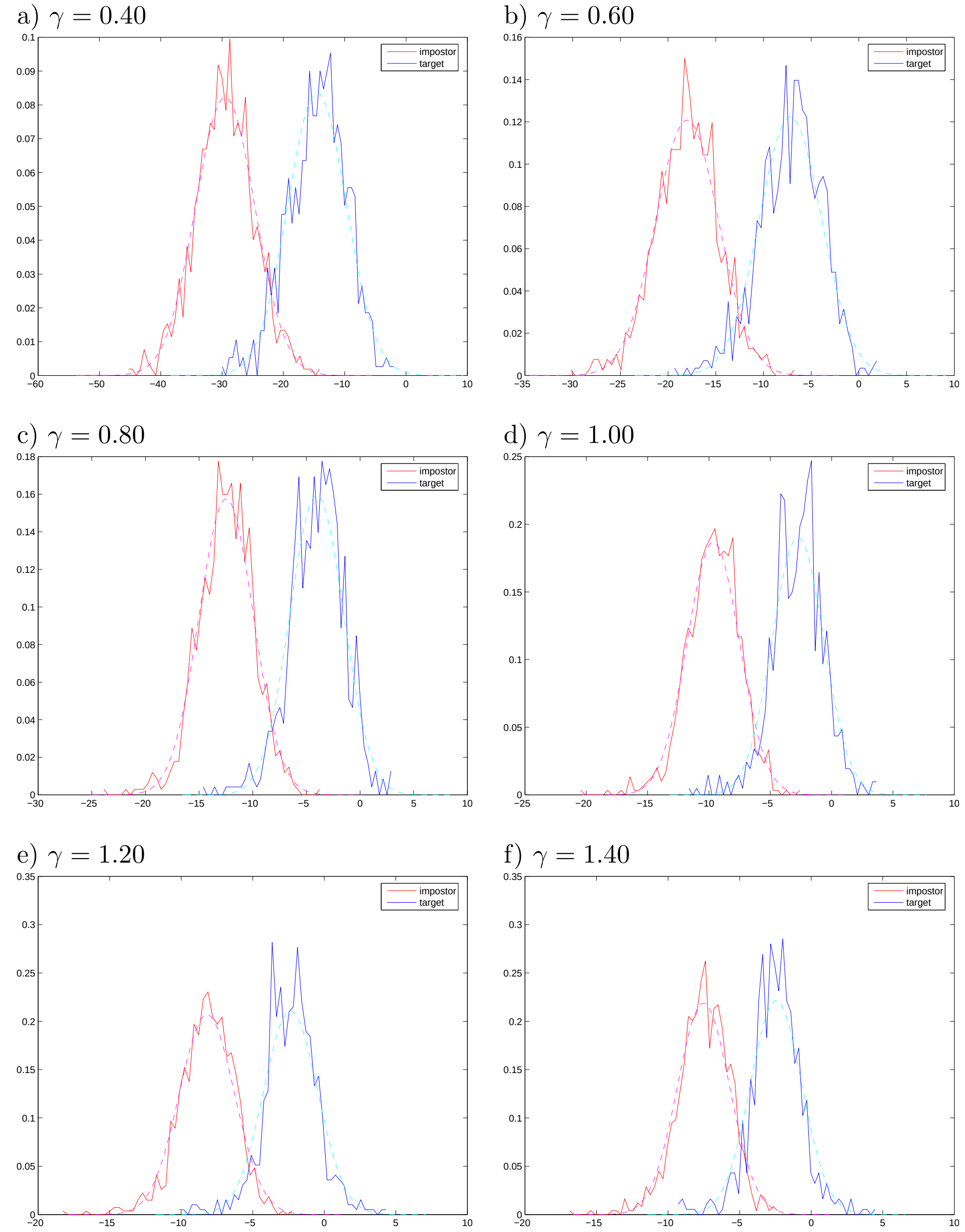


a) $\gamma = 0.60$

b) $\gamma = 1.00$

c) $\gamma = 1.40$

Figure E.23: Polycost data: normal quantile plots for *state-dependent variance floors* (ϑ_f = S/X, ϑ_v = X/X) and three scale factors. Data correspond to the "poly" plot in Figure 9.1b (p. 182) and Figures E.22a,b.



a) $\gamma = 0.40$

b) $\gamma = 0.60$

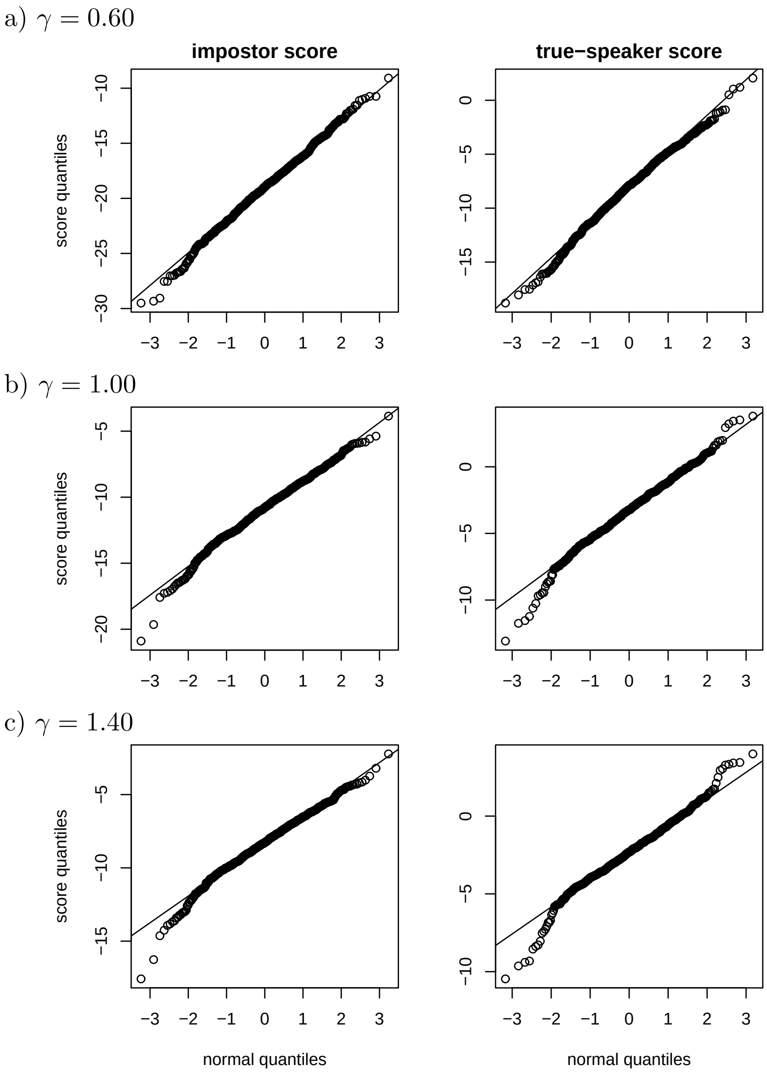
c) $\gamma = 0.80$

d) $\gamma = 1.00$

e) $\gamma = 1.20$

f) $\gamma = 1.40$

Figure E.24: Polycost data: Score histograms for *state-dependent variance floors* ($\vartheta_f = \mathrm{S/X}$, $\vartheta_v = \mathrm{X/X}$) and a range of scale factors. Data correspond to the "poly" plot in Figure 9.1b (p. 182) and Figures E.22a,b. Dashed lines are normal distributions estimated from the actual score values.


a) $\gamma = 0.60$

b) $\gamma = 1.00$

c) $\gamma = 1.40$

Figure E.25: Polycost data: normal quantile plots for *mixture component-dependent variance floors* (ϑ_f = X/X, ϑ_v = X/X) and three scale factors. Data correspond to the "poly" plot in Figure 9.1c (p. 182) and Figures E.22c,d.

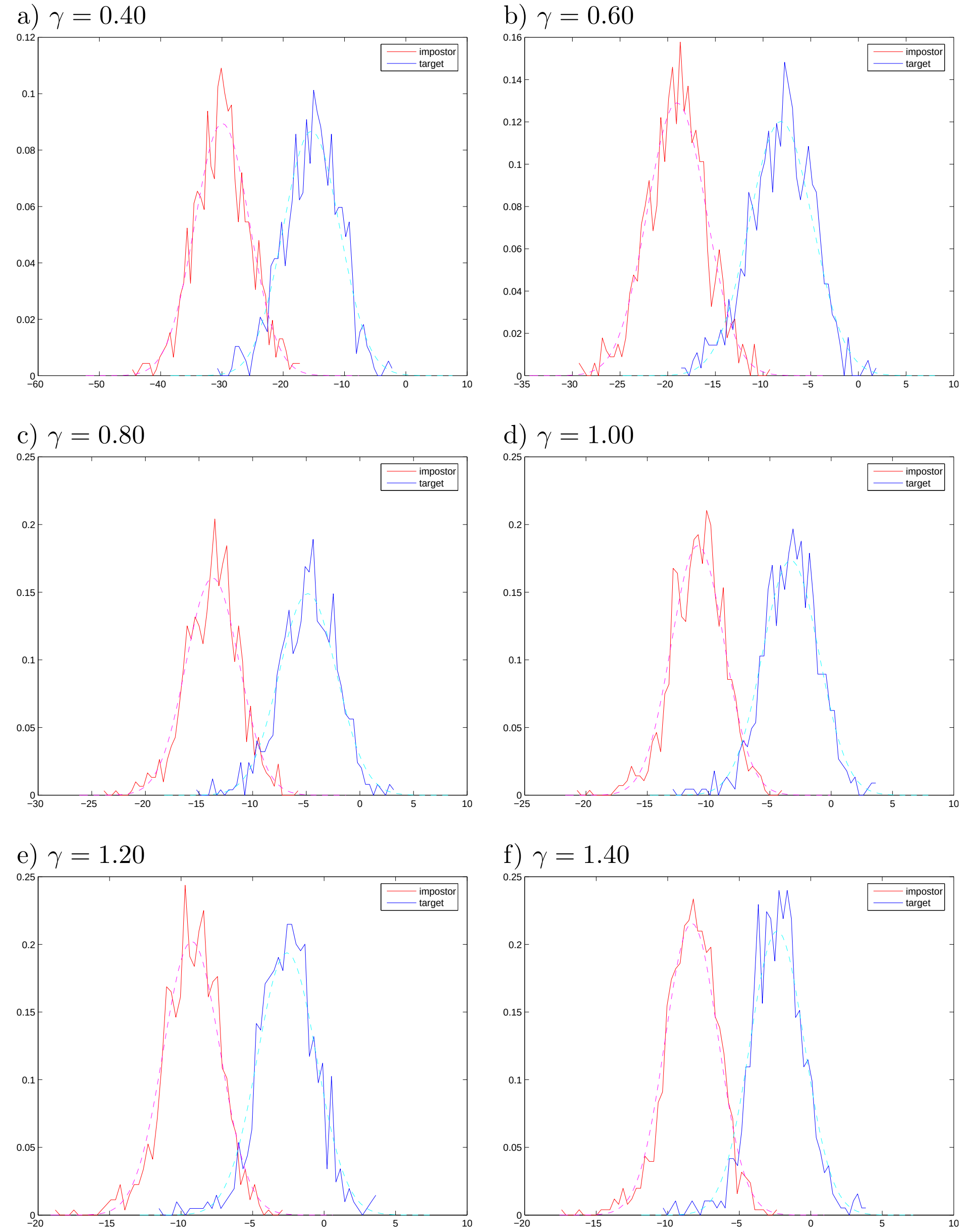
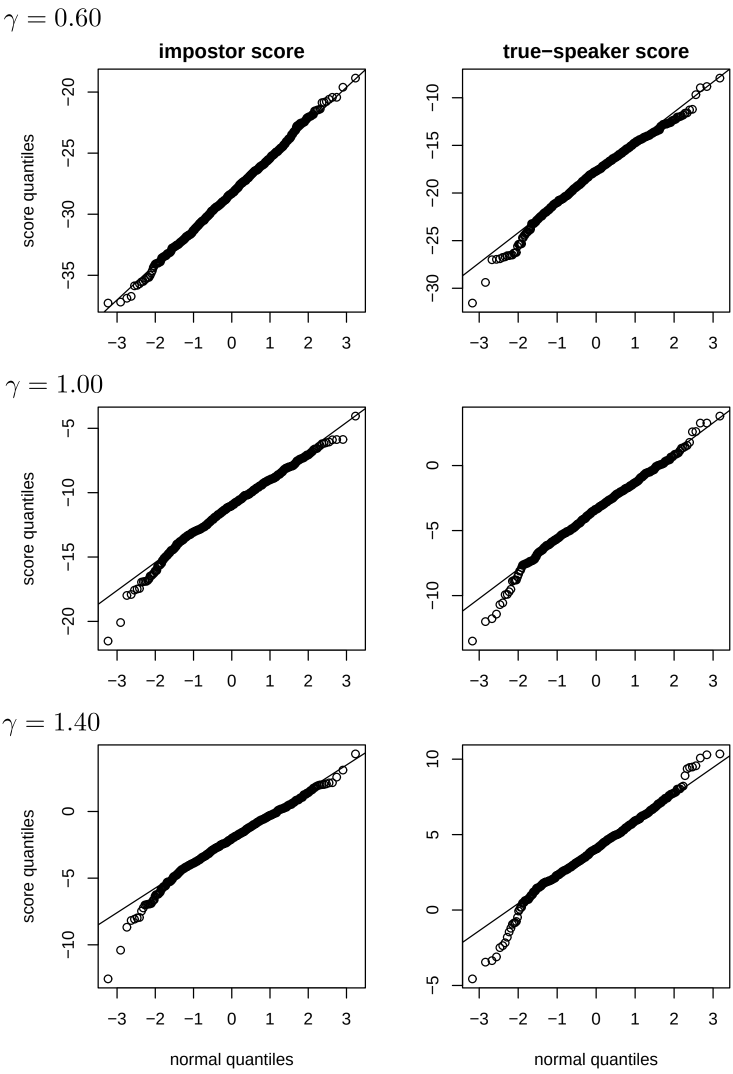


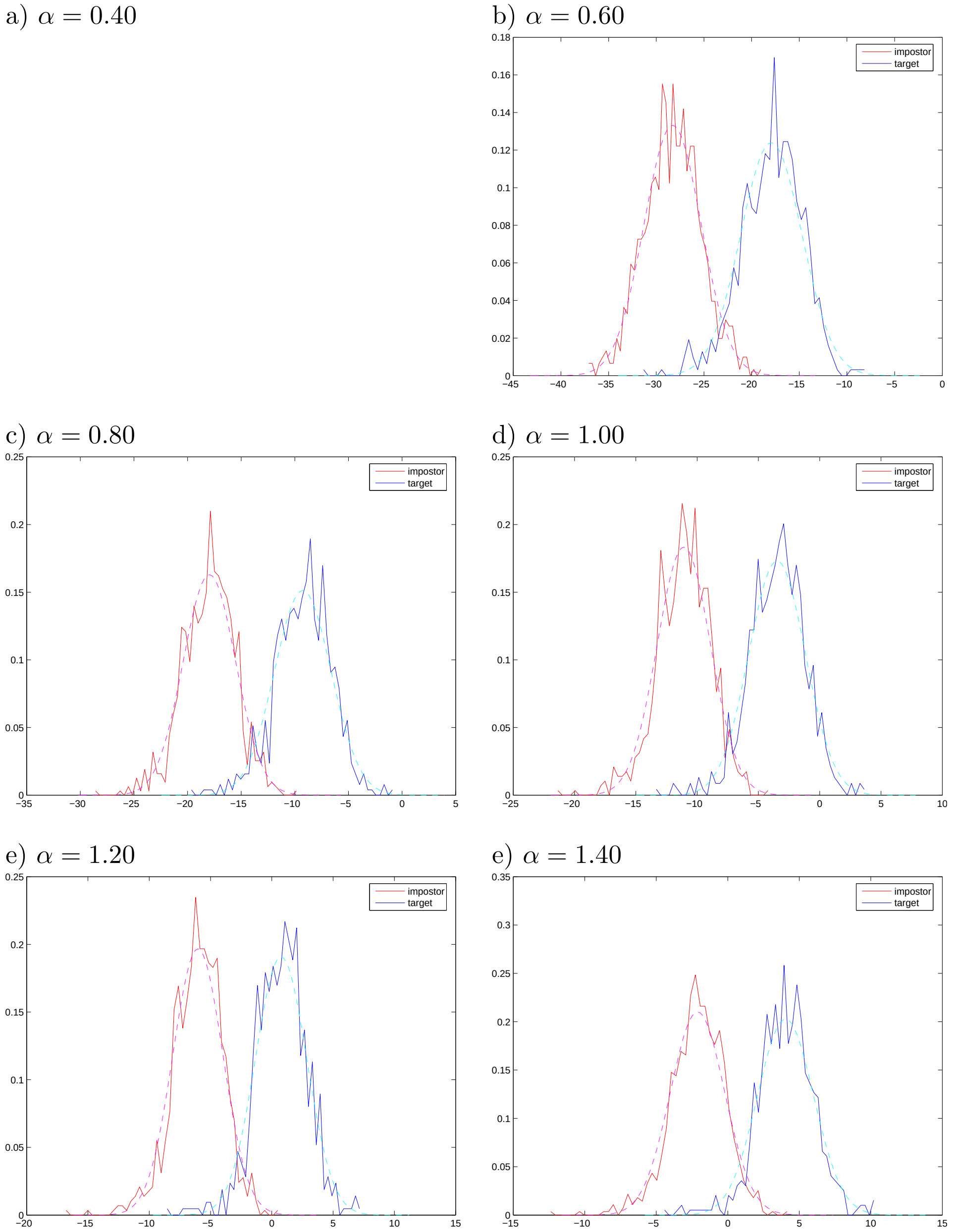
Figure E.26: Polycost data: Score histograms for *mixture component-dependent variance floors* ($\vartheta_f = \vartheta_v =$ X/X) and a range of scale factors. Data correspond to the “poly” plot in Figure 9.1c (p. 182) and Figures E.22c,d. Dashed lines are normal distributions estimated from the actual score values.


a) $\gamma = 0.60$

b) $\gamma = 1.00$

c) $\gamma = 1.40$

Figure E.27: Polycost data: normal quantile plots for *variance scaling* with untied variances (ϑ_v = X/X) and three scale factors. Data correspond to the "poly" plot in Figure 9.1f (p. 182) and Figures E.22e,f.



a) $\alpha = 0.40$

b) $\alpha = 0.60$

c) $\alpha = 0.80$

d) $\alpha = 1.00$

e) $\alpha = 1.20$

e) $\alpha = 1.40$

Figure E.28: Polycost data: Score histograms for *variance scaling* with untied variances (ϑ_v = X/X) and a range of scale factors. Data correspond to the "poly" plot in Figure 9.1f (p. 182) and Figures E.22e,f. Dashed lines are normal distributions estimated from the actual score values.

Appendix F

PER experiments - extended results

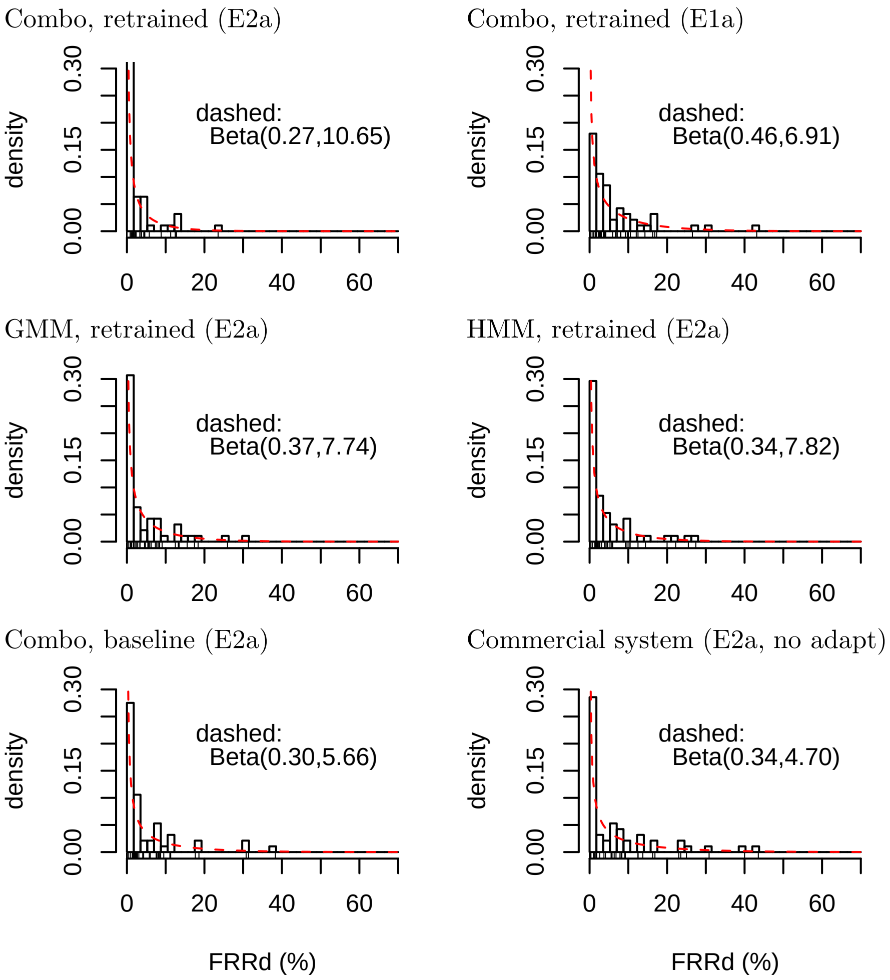


Figure F.1: Distribution of FRRd (at target-independent EERd-threshold) over targets in PER test set T2b_G8.

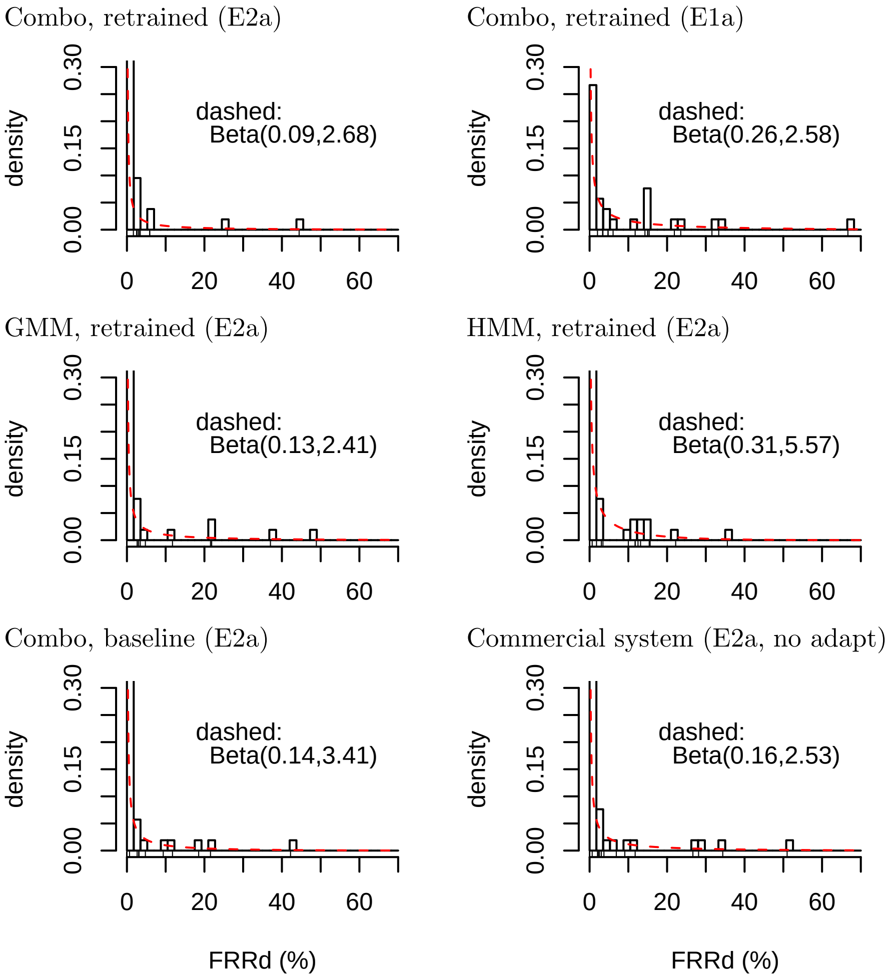


Figure F.2: Distribution of FRRd (at target-independent EERd-threshold) over targets in PER test set T2b_LO.

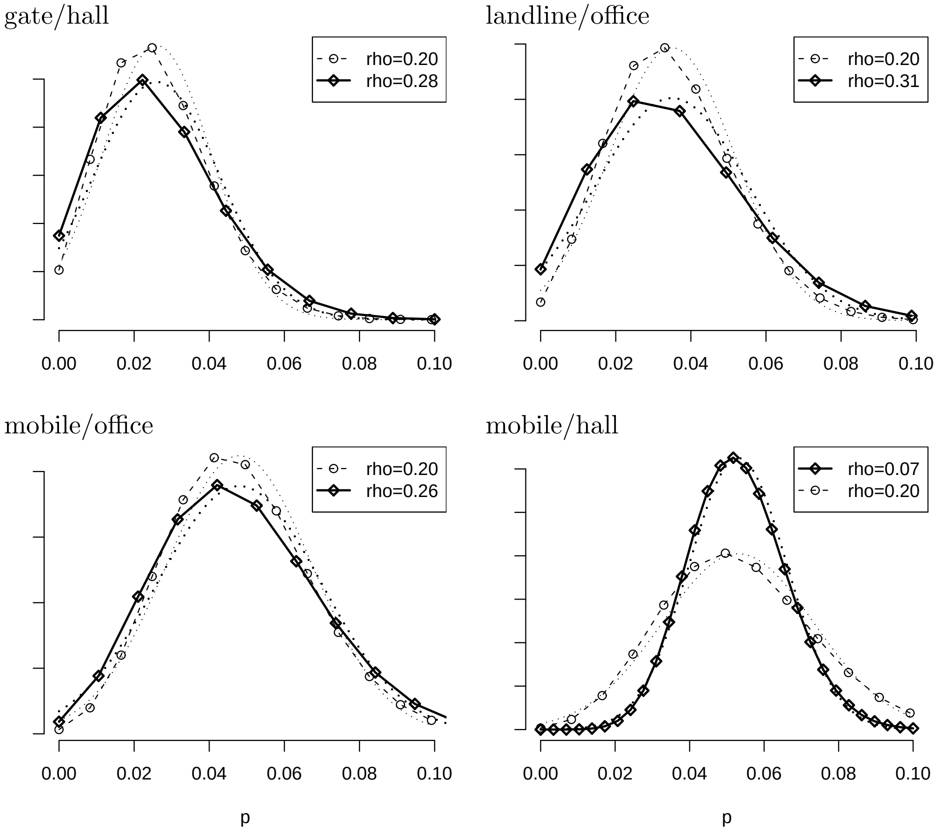


Figure F.3: Binomial distributions used to compute confidence intervals for the *retrained research system*, test sets T2b_Q:c and full enrollment (E2a_c). ρ (rho) for solid lines with diamonds are computed with Method 1, while the distributions for $\rho = 0.20$ (dashed lines with circles) correspond to Method 2 with an *a posteriori* choice of ρ. The normal approximation to each binomial is shown as a dotted line.

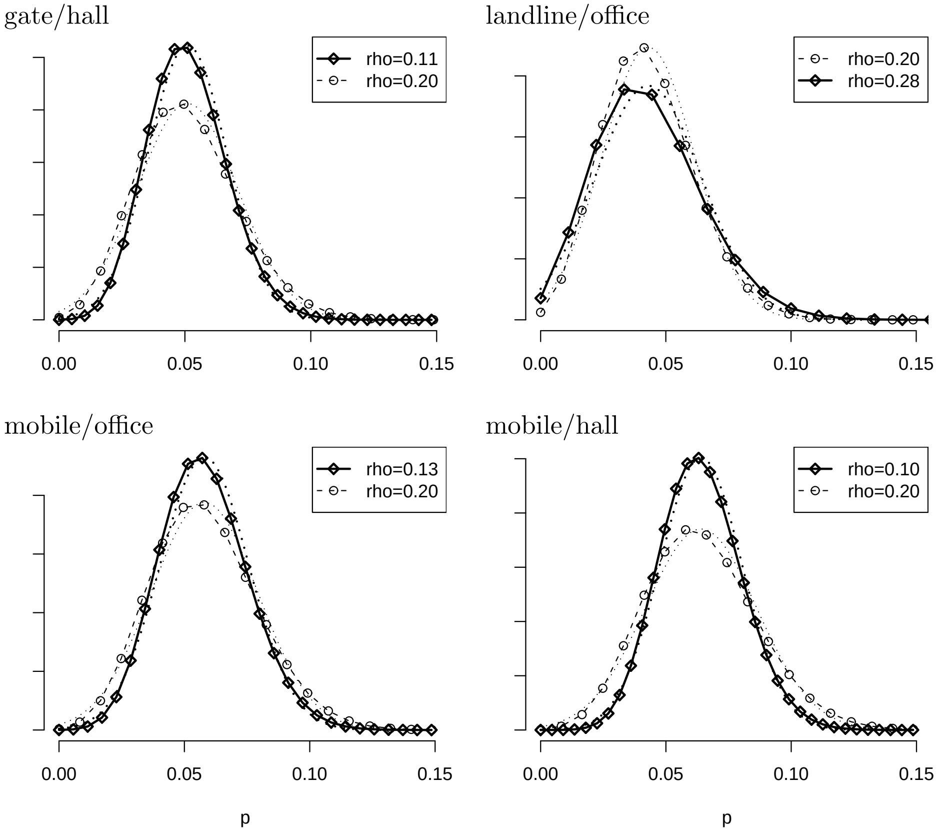


Figure F.4: Binomial distributions used to compute confidence intervals for the *baseline research system*, test sets T2b_Q:c and full enrollment (E2a_c). ρ (rho) for solid lines with diamonds are computed with Method 1, while the distributions for $\rho = 0.20$ (dashed lines with circles) correspond to Method 2 with an *a posteriori* choice of ρ. The normal approximation to each binomial is shown as a dotted line.

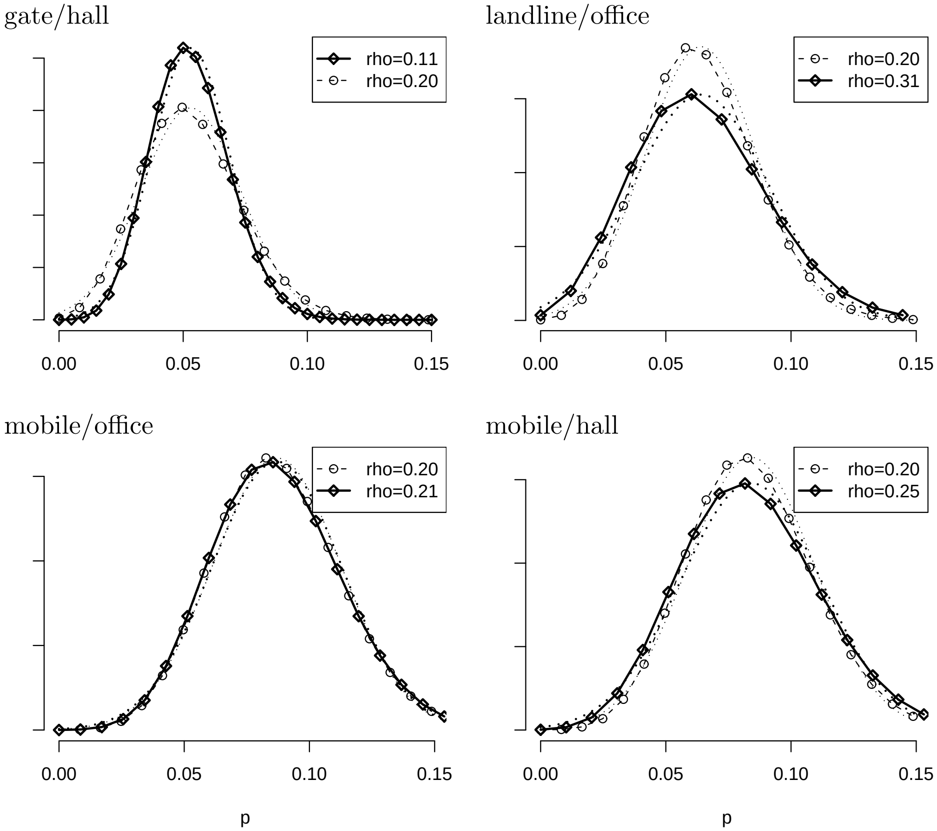


Figure F.5: Binomial distributions used to compute confidence intervals for the *commercial system*, test sets T2b_Q:*c* and full enrollment (E2a_*c*). ρ (rho) for solid lines with diamonds are computed with Method 1, while the distributions for $\rho = 0.20$ (dashed lines with circles) correspond to Method 2 with an *a posteriori* choice of ρ. The normal approximation to each binomial is shown as a dotted line.

Appendix G

ATLAS log file DTD

```
<!-- ############################################################

  File: atlas-log.dtd

  Document Type Definition for log files generated by Atlas.

  Created for use with JAXB (version 1.0ea).
  Note: JAXB 1.0ea does not support NOTATION, ENTITY, ENTITIES and
  enumeration NOTATION types.

  Missing from this DTD:
   - dialog component specific things
   - ...

  General attribute descriptions:
   - startDateTime specifies a calendar date and time
   - startTime specifies a time in milliseconds since the start of
     the session where an element belongs.
   - startTime and duration for say, play, gesture and listen elements
     refer to the media signal associated with the element.

################################################################## -->

<!-- dataDir should be base directory for session-dependent subdirectories -->
<!ELEMENT application ( history, propertyList, resources ) >
<!ATTLIST application
          name CDATA #REQUIRED
          startDateTime CDATA #REQUIRED
  duration CDATA #IMPLIED
  dataDir CDATA #IMPLIED >

<!ELEMENT history ( session | signal )* >

<!-- dataDir should specify a session directory as a path relative to
    a base directory specified in the application element. It may also
    specify an absolute path, especially if no base directory is specified
     for the application. -->
<!ELEMENT session ( subHistory, propertyList ) >
<!ATTLIST session
          startDateTime CDATA #REQUIRED
  duration CDATA #IMPLIED
  dataDir CDATA #IMPLIED
  localId CDATA #REQUIRED
  remoteId CDATA #REQUIRED >

<!ELEMENT signal ( description, action? ) >
<!ATTLIST signal
          type ( incoming | hangup | other ) #REQUIRED
          startDateTime CDATA #REQUIRED
          startTime CDATA #REQUIRED
  duration CDATA #IMPLIED >

<!ELEMENT description ( #PCDATA ) >

<!ELEMENT action ( #PCDATA ) >

<!-- propertyList may contain both given input values and resulting output
     values -->
<!-- purpose should provide a short intuitive description for what
    the component is supposed to do, for example "login", "confirmation" or
     "getDate". -->
<!-- type should be a unique identifier for the component type,
     such as its qualified classname -->
<!ELEMENT dialogComponent ( subHistory, propertyList ) >
<!ATTLIST dialogComponent
          purpose CDATA #REQUIRED
          type CDATA #REQUIRED
          startTime CDATA #REQUIRED
  duration CDATA #IMPLIED >

<!ELEMENT subHistory ( ( dialogComponent | turn | signal )* ) >

<!ELEMENT turn ( say | play | gesture | listen | signal )+ >
```

```
<!ATTLIST turn
          startTime CDATA #REQUIRED
  duration CDATA #IMPLIED >

<!-- startTime indicates when the audio signal started playing in the
     media device(s) -->
<!-- duration indicates the length of the played audio signal -->
<!ELEMENT say ( message, audioFile?, errorDescription? ) >
<!ATTLIST say
          startTime CDATA #IMPLIED
  duration CDATA #IMPLIED >

<!-- startTime indicates when the audio signal started playing in the
     media device(s) -->
<!-- duration indicates the length of the played audio signal -->
<!ELEMENT play ( message?, audioFile, errorDescription? ) >
<!ATTLIST play
          startTime CDATA #IMPLIED
  duration CDATA #IMPLIED >

<!-- startTime indicates when the gesture started showing in the media
     device(s) -->
<!-- duration indicates the length of the animation -->
<!ELEMENT gesture ( errorDescription? ) >
<!ATTLIST gesture
          name CDATA #REQUIRED
          startTime CDATA #IMPLIED
  duration CDATA #IMPLIED >

<!ELEMENT message EMPTY >
<!ATTLIST message
          text CDATA #REQUIRED
  language CDATA #REQUIRED >

<!ELEMENT expectedMessage EMPTY >
<!ATTLIST expectedMessage
          text CDATA #REQUIRED
  language CDATA #IMPLIED
  description CDATA #IMPLIED >

<!-- expectedMessage is a description of what was expected from
     the user a priori -->
<!-- audioFile refers to the output of a speech detector if one is used,
     otherwise to the recorded data -->
```

```
<!-- id is an id number unique among listen operations in a single
     application run -->
<!-- startTime indicates when the audio signal started recording in
     the media device -->
<!-- duration indicates the length of the recorded audio signal -->
<!ELEMENT listen ( expectedMessage?, result, errorDescription?,
                   processors, audioFile?, speechDetector? ) >
<!ATTLIST listen
          id CDATA #REQUIRED
          startTime CDATA #IMPLIED
  duration CDATA #IMPLIED >

<!-- audioFile refers to input data to speech detector (the recorded data) -->
<!ELEMENT speechDetector ( sdUtterance, audioFile?, propertyList ) >
<!ATTLIST speechDetector
          resourceName CDATA #REQUIRED >

<!ELEMENT propertyList ( property )* >

<!ELEMENT property ( #PCDATA ) >
<!ATTLIST property
          name CDATA #REQUIRED >

<!-- omitted startTime and duration means no utterance was detected -->
<!ELEMENT sdUtterance EMPTY >
<!ATTLIST sdUtterance
          startTime CDATA #IMPLIED
  duration CDATA #IMPLIED >

<!ELEMENT audioFile EMPTY >
<!ATTLIST audioFile
          fileName CDATA #REQUIRED
          fileFormat ( raw | wav | other ) #REQUIRED
  sampleRate CDATA #REQUIRED
  sampleFormat ( lin16 | lin8 | alw | ulw | other ) #REQUIRED
  byteOrder ( low | high | none ) #REQUIRED
  length CDATA #IMPLIED >

<!ELEMENT processors ( processor )* >

<!-- locale, if defined, must be a string like "sv_SE", "en_GB".
     Undefined means the processor is locale independent. -->
<!-- propertyList is meant to list properties specific to the particular
     application of the processor. More static properties should be
     listed as resource properties. -->
<!ELEMENT processor ( result, errorDescription?, propertyList ) >
<!ATTLIST processor
          resourceName CDATA #REQUIRED
  resourceType ( RECOGNIZER | VERIFIER | OTHER ) #REQUIRED
  locale CDATA #IMPLIED >

<!ELEMENT result ( speakerHypothesis*, textHypothesis*, labelFile? ) >
<!ATTLIST result
          processingTime CDATA #REQUIRED >

<!ELEMENT speakerHypothesis ( speakerId, personName? ) >
<!ATTLIST speakerHypothesis
          score CDATA #REQUIRED
  decision ( reject | accept | none ) #IMPLIED >

<!ELEMENT speakerId ( #PCDATA ) >

<!ELEMENT personName ( #PCDATA ) >

<!ELEMENT textHypothesis ( message ) >
<!ATTLIST textHypothesis
          score CDATA #REQUIRED >

<!ELEMENT labelFile EMPTY >
<!ATTLIST labelFile
          fileName CDATA #REQUIRED
  format ( htk ) #REQUIRED >

<!-- startTime is the time when the error occured (relative to the start
     of the session in which it occured) -->
<!ELEMENT errorDescription ( errorDescription? ) >
<!ATTLIST errorDescription
          startTime CDATA #REQUIRED
          message CDATA #IMPLIED
          exceptionClassName CDATA #IMPLIED >

<!ELEMENT resources ( resource )* >

<!-- locale, if defined, must be a string like "sv_SE", "en_GB".
     Undefined means the resource works in any locale. -->
<!ELEMENT resource ( propertyList ) >
<!ATTLIST resource
          name CDATA #REQUIRED
  type CDATA #REQUIRED
  locale CDATA #IMPLIED
  numGets CDATA #IMPLIED >

<!-- end of DTD -->
```