Maximum-likelihood linear regression coefficients as features for speaker recognition

Marc Ferràs Font

To cite this version:

Marc Ferràs Font. Maximum-likelihood linear regression coefficients as features for speaker recognition. Linguistics. Université Paris Sud - Paris XI, 2009. English. NNT : . tel-00616673

Université Paris-Sud – Faculté des sciences d'Orsay
École Doctorale d'Informatique de Paris-Sud
Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur

Maximum-Likelihood Linear Regression Coefficients as Features for Speaker Recognition

Thèse défendue par **Marc Ferràs Font** pour le diplôme de docteur en sciences, spécialité informatique, le 10 juillet 2009 à Orsay

Contents

Introduction

Communication is possibly the most outstanding capability in a human being. As children, we quickly learn to interact with the world that surrounds us. We use our senses to feel what is around and we use our muscles, in any form, to modify what is around. From this interaction, sensorial and past experiences melt into tight associations, according to our needs, to what is pleasant or unpleasant, to what is allowed or not, and definitely creating a unique internal world... which is being projected out all the time, from the beginning. In and out, out and in, that is communication: the way we look, smile, move, dress or smell and, of course, the words we say too.

Undoubtedly, natural language is a powerful way to communicate, be it written or spoken. Words can talk about everyday life, feelings, abstract reasoning, or even the meaning of words themselves. By the way, this thesis is also words, hopefully a bit more than that. Spoken language is natural language enriched by real people. When we speak, we communicate our thoughts in the form of a linguistic message but we do communicate our state of mind as well, who we are at that particular moment, or an instance of who we are. Consciously or unconsciously, we choose our words according to how we feel, depending on the audience we speak to, the place we grew up in and, of course, the linguistic message itself. A large amount of factors, as many as defining any individual, can decide what our words will be and how they will be uttered.

The human vocal apparatus, composed of lungs, larynx and vocal tract, is specialized in converting what we want to say into sounds, the way we learned it. We know how words and phonemes sound like for languages and environments we have been exposed to. We know that an aggressive message will probably be in a loud and harsh voice, or that we are likely to speak in a soft and melodic voice to express tenderness and affection. This apparatus is essentially the same for all of us except for some differences in the physiology, as it is particularly visible in related pathologies. However, given that speech production mechanisms are highly unconscious, the speech signal itself can reveal information that is particular to the speaker, be it at the physiological or behavioral levels. Eventually, sound waves generated by the vocal apparatus are propagated away following the laws of physics and are hopefully received by other humans or, why not, machines.

We can glimpse that a large amount of information is available any time we speak. It seems that it is enough for us humans to recognize people by its voice, including our own. One can think about the possibility of a machine doing such task. Among the so-called speech technologies, automatic speaker recognition uses speech signals to identify the person that is speaking by using measurements taken from a recording of his/her voice. This technology is accessible, does not require any expensive equipment and it is not obtrusive, all of them being important factors for a wide-spread use. It can be used in cases where the person to be identified has no knowledge of it, whereas other biometrics such as retina scan or fingerprints are often not welcome by users. Related applications range from user authentication, typically in a rather text-dependent form, to multimedia indexing, surveillance and forensics. However, voice has one serious downside as it changes over time. This is intrinsic to human nature, and it leaves evidence that a specific voice is not unique to a person.

Both science and industry have shown increasing interest in the challenges posed by this technology. Current systems systematically use model vocal tract properties to derive speaker-relevant features. These are typically handicapped by several sources of variability and distortion. For instance, the linguistic message adds variability which might mask other speaker information. Inter-session variability, including channel mismatch or mood, is usually an important source of loss of performance. Language and dialect, as highlighted in international evaluation campaigns, significantly compromise system performance as well, not to forget the acoustic environment, i.e noise and reverberation, the old enemy to any speech application. Therefore, most of the current research effort in speaker recognition is directed to finding new efficient features.

Statistical modeling, either generative or discriminative, largely prevails in speaker recognition systems. Generative models make a smooth representation of what a speaker is in terms of observed features. In contrast, discriminant models make explicit use of additional data to separate what is to be characterized, i.e. a speaker, from what it is not, i.e. other speakers. Support Vector Machines (SVM) is one of such paradigms, one that has been vastly used for many binary pattern recognition tasks, including automatic speaker verification. SVM exhibit an interesting generalization behavior in high dimensional feature spaces, being able to deal even with degenerate data matrices[1].

Being feature-model coupling an important issue, rediscovery of SVM in the last decade gave birth to new ideas in many pattern-recognition disciplines, splashing speaker recognition too. Due to the variable-length nature of speech data as well as some efficiency constraints in the quadratic solvers required for SVM training, speaker recognition took some time to take off on using SVM. Currently, they have become widely spread in the community, rivalling with any other state-of-the-art approach to modeling. New features and strategies have allowed for a reformulation of the speaker recognition problem under different assumptions, i.e. Vapnik's statistical learning theory [Vapnik, 1998] and the structural risk minimization criterion.

Goals and Contributions

The goal of this thesis is to find new and efficient features for speaker recognition. We are mostly concerned with the use of the Maximum-Likelihood Linear Regression (MLLR) family of adaptation techniques as features in speaker recognition systems. MLLR transform coefficients are able to capture speaker cues after adaptation of a speaker-independent model using speech data. The resulting supervectors are high-dimensional and no underlying model guiding its generation is assumed a priori, becoming suitable for SVM for classification.

This thesis brings some contributions to the speaker recognition field by proposing new approaches to feature extraction and studying existing ones via experimentation on large corpora:

1. We propose a compact yet efficient system, CMLLR-SVM, which tackles the issues of transcript- and language-dependency of the standard MLLR-SVM approach by using single-class Constrained MLLR (CMLLR) adaptation transforms together with Speaker Adaptive Training (SAT) of a Universal Background Model (UBM).

[1] When less data samples than dimensions are available.

2. We propose several alternative representations of CMLLR transform coefficients based on the singular value and symmetric/skew-symmetric decompositions of transform matrices.

3. We develop a novel framework for feature-level inter-session variability compensation based on compensation of CMLLR transform supervectors via Nuisance Attribute Projection (NAP).

4. We perform a comprehensive experimental study of multi-class (C)MLLR-SVM systems along multiple axes including front-end, type of transform, type of model, model training and number of transforms.

5. We compare CMLLR and MLLR transform matrices based on an analysis of properties of their singular values.

6. We propose the use of lattice-based MLLR as a way to cope with erroneous transcripts in MLLR-SVM systems using phonemic acoustic models.

Outline

This manuscript is organized as follows:

Chapter 1 presents brief background knowledge about speaker recognition technology, starting from generic biometric principles going through inter- and intra-speaker sources of variability and describing the structure of speaker verification systems. We detail the common processing steps of speaker verification systems and we finish the chapter covering performance measures and current database resources.

Chapter 2 describes generative approaches to speaker verification. After a brief introduction to speech recognition technology, we discuss Gaussian mixture models and related training and adaptation techniques. We describe the GMM-UBM paradigm and we give an overview of current inter-session variability compensation techniques based on GMM.

Chapter 3 covers approaches using SVM as classifiers. After introducing SVM theory, we quickly get into SVM-based speaker verification describing the general structure then focusing on sequence and parameter-space kernels. The latter section discusses the MLLR-SVM paradigm which is the basis of the work developed in this thesis. We finish including an overview of inter-session variability compensation techniques for SVM-based systems.

Chapter 4 describes three acoustic speaker verification systems, PLP-GMM, PLP-SVM and GSV-SVM, used as baselines for system comparison and fusion purposes in further chapters. We also describe the experimental protocol used across the manuscript.

Chapter 5 moves around the CMLLR-SVM system. We start giving the required background about CMLLR and speaker adaptive training and how both can be used in speaker recognition systems. The base CMLLR-SVM system is described and evaluated and we tackle inter-session compensation of CMLLR supervectors also giving experimental results. We perform a brief analysis of CMLLR transforms that leads to alternative representations of CMLLR transforms for CMLLR-SVM systems. At the end of the chapter we present and evaluate a feature-level inter-session variability compensation framework based on CMLLR adaptation. This chapter covers points 1, 2 and 3 of the contributions in the previous section.

Chapter 6 presents a comprehensive experimental study of multi-class (C)MLLR-SVM systems. We first describe the experimental setup focusing on front-end and acoustic models based on a LVCSR system. We present experimental results for multi-class CMLLR-SVM and MLLR-SVM systems covering a wide range of system parameters. A side trip explores differences between CMLLR and MLLR transforms to support certain results from a more theoretical point of view. We explore systems using the combination of CMLLR and MLLR coefficients as features and we finish performing system fusion of baseline and (C)MLLR-SVM systems. This chapter covers contributions 4 and 5 of the previous section.

Chapter 7 explores lattice-based MLLR as a way of dealing with transcription errors in multi-class MLLR-SVM systems. A brief discussion about the impact of transcription errors on such systems is given first. Lattice-based MLLR is described next following a close parallel to standard MLLR. Lattice MLLR-SVM systems used in our experimentation are described and evaluated. We finish the chapter giving fusion results for these systems. This chapter covers contribution 6 of the previous section.

Part I

Automatic Speaker Recognition

Chapter 1

Speaker Recognition Background

Voice has unique advantages for person identification use while the wide range of variabilities affecting speech signals still poses significant technological challenges. After giving some foundations on biometrics and speaker recognition technology in Sections 1.1 and 1.2, Section 1.3 discusses factors contributing to and limiting speaker characterization based on speech signals. We focus on the text-independent Automatic Speaker Verification (ASV) task there on, presenting the general structure of such systems in Section 1.4 as well as giving a literature review of those processing steps common to ASV systems, performance measures and corpora for its experimental evaluation. Section 1.5 gives a brief summary and conclusions of the chapter.

1.1 Biometrics Overview

Speaker recognition technology uses speech as a biometric, which presents unique advantages over other biometrics. However, beyond the popular belief that a speaker can be characterized by a *voiceprint*, in analogy to the term *fingerprint*, as unique signature of a person's identity, using speech has serious drawbacks. In this section, we present the basic principles of biometrics and we further discuss the use of speech signals to identify people.

1.1.1 Principles of Biometrics

Biometrics concerns person identification by using its physical or behavioral traits. A biometric identifier, or just biometric, is the type of measurement used to achieve such identification. The advantage of using a biometric to identify a person is clear: whereas a security tag or a piece of identity can be lost or stolen, a biometric cannot. Whereas a password or a code can be forgotten, a biometric cannot. In other words, a biometric measures something that is intrinsic to a person, so that falsifying identity becomes harder. Biometrics can be broadly classified as:

- **Physical biometrics**, related to measurements of physical properties of our body. Their variation over time is slow. Examples of it are retina vessels or iris features, fingertip patterns, hand geometry or facial characteristics.
- **Behavioral biometrics**, measuring the behavior of the person in some way. Its varia-

tion over time is faster than in physical biometrics. Although requiring a significant effort, some variation patterns can be learned by gifted individuals, which might be seen as a threat for authentication systems. Examples of it are signature, mouse gestures, gait or voice.

Research in biometrics is driven by the fact that no biometric is perfect. Although very performing biometrics exist, these are often unpopular and they may not be widely accepted by users. This can be understood intuitively: the more performing a biometric is, the closer it is to unique traits of the person and, thus, the more menaced his/her intimacy may be. That means performance is not the only desired attribute in a biometric. For its evaluation, other dimensions such as uniqueness, acceptability, universality, permanence or collectability come into play. As no single biometric is suitable to all applications, a multimodal approach, i.e. the use of multiple biometrics together, is often adopted. By fusing multiple biometric technologies, e.g. several biometrics vs. several algorithms, the overall system can improve in any of the evaluation directions. Such diversity can compensate for noisy or missing data, incompatibility or aversion of certain subsets of the population to a certain technology or vulnerability of the system.

1.1.2 Speech as a Biometric

Voice, or more specifically speech, can be used for the purpose of person identification by taking measurements from a speech signal spoken by that person. Given that it is not possible to measure physical attributes from an acoustic waveform, speech is usually understood to be a behavioral biometric. The speech signal is thus assumed to convey significant information related to its production, which results from the neuro-motor commands given while the person speaks. For the most part, these occur at the unconscious level and result from a long-term learning process. However, behavior can be modified through specific re-education work such as learning a foreign language, which is decided consciously. In the same line, accent, sociocultural or idiosyncratic behavior can also be learnt. Nonetheless, speech can be somehow regarded as a physical biometric as well. It is common to model the speech signal to obtain parameters that can be interpreted in physical terms, namely cross-sectional areas or length of the vocal tract. Analysis of this type are possible assuming a common underlying model of human speech production.

From a biometric point of view, an outstanding quality of speech is acceptability. People use their voice to communicate with other people, everyday and everywhere, with almost no effort. Furthermore, even though it may still be intimidating for some, we are gradually getting used to being recorded and to talking to machines too. This fact helps data collection, which can be massively done using the already deployed telephonic network to record phone calls. Even explicit recording of speech data involves non-invasive equipment, e.g. microphones placed at a certain distance from the speaker.

Voice naturally changes over time, however. Passage from childhood to adulthood causes hormonal changes that have a significant effect on voice texture, especially in men, while it keeps on evolving at a slower pace over the entire lifetime. Vocal pathologies, either temporary or chronic, can result in serious alterations of its character as well. Many traits can be even faked by imitators to a certain extent. This variability partly explains its low performance compared to other biometrics such as retina or iris scan. In fact, science has not yet been able to show that the human vocal apparatus can generate two exactly identical utterances, so the unicity of a *voiceprint* as unique speech data identifying a person, is not generally accepted. Ironically, the average layman can become an impostor by using non-specialized equipment to record and exactly play back a message spoken by a person. Consequently, the use of voice as a biometric is likely to be structurally limited, but it certainly exploits unique information that other biometrics do not.

Many factors help discriminate voice character. Particularities in the vocal cords, i.e. size, shape, thickness, tension, and in the articulators, i.e. tongue, teeth, nose, lips, throat, they all have an impact on the speech signal, although not directly measurable. The way we articulate, i.e. how we use our lips, tongue, jaw, along with their dynamics, also help characterize a speaker from a more behavioral point of view. Stress and pitch, i.e. suprasegmental cues, extend over several sounds and/or words in a sentence and can bring extra information including semantic focus, speech rate or emotion. At a higher linguistic level, the structure of words in the sentence or the choice of words itself can be also useful, although a long time span is required to obtain reliable measures. In short, it seems very unlikely that all these features be the same for two different people.

Applications fall into two different categories depending on who benefits from speaker recognition systems [Doddington *et al.*, 2000]. In authentication applications, e.g. for facility or computer access control, someone claims to be an user of the system and its identity needs to be verified automatically. Claimant speakers are thus expected to collaborate with the system, for instance, by constraining the message to be identified to a user-specific password or whatever the system asks the user to say. These systems need to be high-performing while using little speech data, which often requires high-quality audio equipment as well as a controlled acoustic environment. In contrast, the beneficiary of a second type of applications is people other than the speaker of interest. Such applications tend to have much less constraints, e.g. they may involve low-quality recordings with unknown acoustic conditions, highly emotional and spontaneous speech, and they are often text-independent. Although more speech data is often available compared to authentication-like systems, performance is significatively lower. This modality can be used to automatically identify or track people in meetings, for instance. Intelligence and other government agencies are also interested in these applications and, although skepticism envelops the subject at the present time, it might also be used in forensics in the future.

1.2 Speaker Recognition Tasks

Speaker recognition applications specialize in several well-defined prototyped tasks that make system evaluation easier. These are defined by adding some constraints such as the assumed type of input speech data or a required decision type, while the overall goal is still to characterize a speaker from its speech signal to make decisions based on it. Three major tasks capture most of the research interest in speaker recognition: automatic speaker verification, identification and diarization. We make a quick description of them in the following sections.

1.2.1 Automatic Speaker Verification

Automatic Speaker Verification (ASV) [Bimbot *et al.*, 2004] consists in verifying the claimed identity of an individual. Based on speech data from a known speaker, a decision is made as to whether unknown speech segment belongs to that speaker or not, assuming only one speaker is present in any recording. This task is split into two phases: enrolment, creating a model for the known speaker and test, scoring unknown speech against the known speaker model.

ASV applications use a wide range of text constraints. Systems may require a password that must be known by the user, or a different message may be prompted at each trial, or just let the user speak freely. In any case, text-dependency adds strong constraints in the form of a priori knowledge, and this can be used to make more performing systems. Text-independent ASV involves much higher variability in the linguistic message, which indirectly implies that much more speech material is needed for equivalent performance to a text-dependent system. As a side effect, more data enables extraction of more robust features as well as collection of higher-level information, e.g. word-level or discourse-level cues, for richer speaker characterization.

This thesis focuses on text-independent ASV as the evaluation task. First, ASV is a conceptually simple task while it encompasses all major challenges in speaker characterization. Second, system performance remains stable as more and more users are enrolled, given that two speakers are always compared. Third, as a consequence of taking open decisions, systems require proper calibration. Fourth, although most of the current applications are text-dependent, text-independency adds complexity as well as richness to the task, which pushes research a step further from immediate needs in the industry. Fifth, it is the focus of periodic international evaluation campaigns that provide priceless data resources and a common evaluation protocol, eventually promoting exchange in the scientific community.

1.2.2 Automatic Speaker Identification

Automatic Speaker Identification (ASI) classifies a speech recording as belonging to one speaker in a set of speakers, assuming only one speaker per recording. The task is split into enrolment and test phases with analogous operation to ASV. ASI systems compare unknown speech against all the stored models, deciding on the better matching speaker. Either in a text-dependent or text-independent form, we find two major variants of ASI: closed-set and open-set. In closed-set ASI, all the speakers are known and the system decision is always one of the known users. However, in open-set ASI, unknown speech can be rejected as not belonging to any enrolled speaker. Therefore, this variant can be thought of as involving both closed-set ASI and ASV.

Performance of ASI systems decrease as the number of speakers grows, since the overall distance from speaker to speaker decreases too. Due to the rejection possibility in open-set ASI systems, proper calibration is essential. Closet-set ASI decision can be reduced to a competition among speakers models, i.e. a relative decision, which is less critical.

1.2.3 Automatic Speaker Diarization

Automatic Speaker Diarization (ASD) automatically finds when people speak in a recording. The task is usually split into two simpler tasks: speaker segmentation finds acoustic changes, hopefully speaker-turn changes, all along the recording. Speaker clustering finds similarities between these segments and groups them as belonging to the same speakers. The output is a variable number of unlabeled speakers with a list of time boundaries for each speaker and segment.

As opposed to ASV and ASI, this task involves multiple speakers in the same recording. In this sense, human interaction in the speech signal entails additional complications, e.g. the use of distinct microphones (channels) or the presence of speaker overlap. This task is highly unsupervised in nature where neither the number of speakers, the linguistic message or any speaker models are known a priori.

1.3 Speaker Characterization

Speaker recognition technology implicitly assumes that a speech signal carries valuable information for characterization of its speaker. However, we also know that the same signal is used for diverging purposes in speech technologies, be it speaker recognition, speech recognition, or any other task. It is thus of interest to understand how speaker recognition systems are affected by the speech signal variability. To gain insight into it, we have assumed two broad sources of variability, inter-speaker and intra-speaker variabilities, in the speech signal. Although this classification gives a rough idea of which a priori information is desired and which should be rejected in speaker recognition systems, it should be also clear that both behaviors are ever present in any source of variability to a certain extent. Sections 1.3.1 and 1.3.2 discuss a several sources that are considered to contribute to either inter- or intra-speaker variabilities.

1.3.1 Sources of Inter-Speaker Variability

Since nobody knows yet what a speaker exactly is, speaker recognition uses diverse measures to represent it, so-called features. As a collection of features is always a partial representation of a speaker, the goal is to find those features that represent it most accurately and efficiently. Features exhibiting high inter-speaker variability are desirable for speaker recognition, as they result in a large separation among speakers, becoming easier to discriminate. In the following, we present a list of sources that significantly contribute to inter-speaker variability:

- **Vocal tract**: The vocal tract is the set of cavities where the sound from the vocal cords or lungs resonate before radiation of actual speech sounds through the mouth. It consists of the larynx, and pharynx, oral and nasal cavities. Each of these has its own physical properties such as size, shape, and texture that are specific to the speaker. Moreover, active articulators, i.e. tongue and lips, in the vocal tract can move over and change its resonance characteristics, resulting in the whole range of sounds that can be generated by humans. In this sense, articulation habits, whatever the origin, help characterize a speaker. Currently, the vocal tract is probably the richest source of low-level speaker-specific information available.

- **Vocal cords**: The vocal cords are two mucous membranes placed across the larynx that are capable of oscillating. By mastering the muscular strain affecting the vocal folds together with the air flow expelled from the lungs, a roughly periodic sound can be generated as a result of fold openings and closures. For the most part, soft, melodic, creaky, breathy voice is generated at this level, so vocal cords are highly responsible for voice character. Furthermore, temporal patterns in the fundamental frequency, also known as pitch contours, convey significant emotional and semantic load that can be related to speaker identity as well. The acoustic signal in the vocal cords is not directly measurable and, in practice, it is estimated using the filter-source model[1] [Fant, 1970]. Besides pitch contours, information related to the vocal cords is marginally exploited in speaker recognition systems, probably because of the highly random nature[2] of the related acoustic signal.

[1] The source-filter model of speech production explains any speech signal as originating from a uncorrelated signal source (vocal cords) passing through a filter modeling a set of resonant cavities (vocal tract).

[2] Vocal cords receive air expelled directly from the lungs. A random, or chaotic at the most, acoustic signal lacking a clear structure is expectable at this point.

- **Lexical and syntactical patterns**: The usage patterns of words and phrases can be another source of speaker-specific information [Doddington, 2001]. These cues are focused on the linguistic message, aiming at characterizing the idiolectal traits of the speaker. In practice, though, a large amount of speech data is typically required for reliable collection of token statistics, which puts serious limitations to the usage of such high-level information.

- **Dialect and language**: Communication inherently implies some kind of exchange with our surroundings, be it our country, our culture, or just our family or friends. Being exposed to an environment for a certain amount of time may eventually change our speaking habits at the articulation, prosodic, lexical and syntactical levels.

1.3.2 Sources of Intra-Speaker Variability

In section 1.3.1 we have listed some important sources of inter-speaker variability. These significantly contribute to characterize a speaker but at the same time involve variation for any given speaker. This intra-speaker variability, either speaker-dependent or speaker-independent, corrupts and distorts speaker-specific features. We name a few sources of intra-speaker variability in the next list:

- **Voice changes**: A disadvantage of using voice over other biometrics concerns its evolution over time. Speech samples from the same speaker can show a high acoustic mismatch along short or long time-scales. These changes are usually considered as adverse since we rarely have any knowledge about them. Variations in the voice currently considered as harmful for speaker recognition include

  – Emotions, resulting in short-term variations of the voice character.

  – Vocal pathologies, having an impact on the physical properties of the vocal apparatus properties and/or neuro-motor commands.

  – Aging, resulting in a slow evolution at the physical and behavioral level.

  However, if the underlying mecanism guiding change is understood, its effects can be compensated or even exploited positively to characterize a speaker. In short, change itself can be rather considered as a source of richness[3], whereas unpredictable or unmodeled change is not desired.

- **Environmental distortion**: Microphones are the most widely-used transductors used for recording speech. Due to the limited size of diafragms used, microphones always capture sound coming from directions other than that of the speaker. Therefore, recordings can contain sound other than the speech from the speaker of interest. Whether it is noise in the room, e.g. air conditioning or computers, the reverberant effects of a room, a car passing by or other people talking around, speaker recognition systems are handicapped by the acoustic environment. Close-talk microphones typically offer good-quality speech, whereas distant microphones provide low-energy speech, low direct-to-reverberant speech ratios and a significant amount of noise, all of these highly depending on the placement of the microphone with respect to the

[3] All sources of inter-speaker variability presented in section 1.3.1 imply an evolution of something over time, and thus, change as well. However, their behavior is understood up to a certain extent and this knowledge is used to characterize the speaker. A similar approach can be envisaged to exploit speaker-dependent intra-speaker variability

speaker. Furthermore, changes in the acoustic environment can happen quite unpredictably. Limiting acoustic condition variability is a way to prevent system performance drop by minimizing mismatch across recordings. This requires, though, a controlled recording protocol which is often not possible in some applications.

- **Channel distortion**: The mechanical and electrical properties of a microphone, as well as the analog circuitry accompanying it, result in distortion of the speech signal. If the speech is compressed in size and transmitted over a network, as it is the case of telephone speech, non-linear distortion and noise can further impair speech quality and even result in audible artifacts. Speaker recognition systems are highly sensitive to these phenomena, specially when channels in the enrolment and test phases differ. Proper modeling of these phenomena is hard even with specific knowledge of the hardware and, even so, the processes involved are often not reversible. However, this type of distortion tends to vary slowly over time[4], a fact often exploited in channel compensation techniques.

- **Linguistic content**: The linguistic message in the recording is another source of variability that, in principle, impairs speaker recognition systems. In text-dependent applications, e.g. those using a password, the low variability in the linguistic content eases the estimation of speaker-specific features by focusing on a specific region of the user's acoustic space[5], thus improving performance. However, as mentioned in section 1.3.1, properly modeling the linguistic structure can be used to characterize a speaker's idiolect, then becoming a source of inter-speaker variability.

1.4 Structure of ASV Systems

Despite the high diversity of approaches, ASV systems operate in a similar way. In the enrolment phase, relevant features are collected from a speech signal uttered by a known, or target, speaker to eventually generate a model for it. In the test phase, speech features taken from an unknown, or test, speaker sample are compared against the target model. Based on a similarity measure, a decision is made as to whether the test speech was uttered by the target speaker. Figure 1.1 illustrates these two phases in the form of a block diagram.

Three of the modules in Figure 1.1 are specially relevant: feature extraction, modeling and decision. Typically arranged in a cascaded structure, information becomes more and more compact as it flows from feature extraction to decision phases. This means that any information that has been rejected in feature extraction can not be recovered later. This means too that low-quality modeling or bad calibration of the decision module probably result in wrong decisions, no matter what our features are.

Although there are no a priori constraints between adjacent modules, e.g. feature extraction and modeling, a reasonable coupling of the two steps is important. Certain modeling paradigms welcome features with certain properties: we guess that a Gaussian model is a good choice for Gaussian-distributed features, while it is probably not for high-dimensional features in sparse spaces; we can use a discriminant model only if data other than that of the speaker of interest, i.e. impostor speaker data, are available.

Feature extraction is the first and probably most important module. If we could extract

[4] Unpredictable channel variations may occur in cellular telephone speech due to fading of radio channels, for instance.

[5] Text-dependent speaker recognition actually uses text-specific and speaker-specific features. The more we constrain the intra-speaker variability the easier the problem.

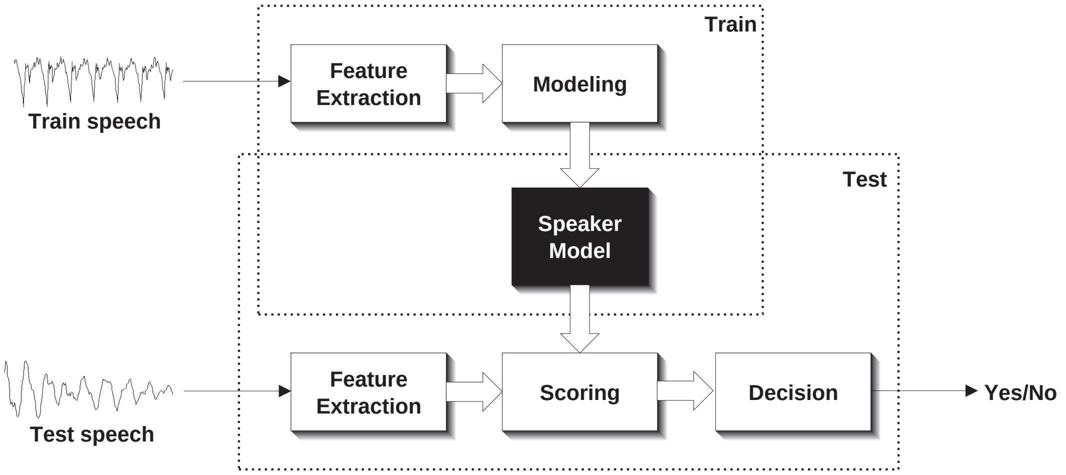

Figure 1.1 — Block diagram of a generic ASV system.

good enough features the rest of the modules would become irrevelant[6]. Obviously, that is never the case and only a partial characterization of a speaker is possible. Most features used in current ASV systems indirectly emerge from spectral-envelope representations of the vocal tract properties (See Section 1.4.1 for a discussion on spectral-envelope features).

Regarding modeling, statistical approaches abound in ASV. From text-dependent to text-independent, statistical modeling often brings improved robustness to unseen patterns or conditions. Current state-of-the-art systems use either a generative approach, discussed in Chapter 2, or a Support Vector Machine (SVM) approach, presented in Chapter 3. Nonetheless, non-statistical approaches, e.g Dynamic Time Warping (DTW) [O'Shaughnessy, 1986; Furui, 1981], are still used in text-dependent ASV or in linguistic-based diarization [Canseco-Rodriguez *et al.*, 2004].

As a binary classification problem, only two decisions can be made in ASV: either the test speaker is accepted as the target speaker or it is rejected. A natural formulation for this problem is a statistical hypothesis test, where we decide upon one of the hypotheses based on a threshold. This subject is discussed in-depth in Section 1.4.5. It is also common to normalize scores before calibration to make the decision more robust to speech variability. Score normalization is discussed in Section 1.4.4.

Current state-of-the-art speaker recognition systems result from the fusion of sub-systems using heterogeneous speaker-relevant information, e.g. several acoustic, prosodic and stylistic sub-systems. Systems are typically combined by fusing scores obtained by all sub-systems, although it can be alternatively carried out in other domains, e.g. at the feature-level. For SVM systems (see Chapter 3) kernel fusion has also been proposed [Dehak *et al.*, 2008]. Score-level system fusion is discussed in Section 1.4.6

1.4.1 Spectral-envelope Features

Beyond complexity and difficulty of handling, speech-related information is not directly observable in the speech signal. Changes in the vocal apparatus are eventually produced by muscles and these are subject to mechanical constraints, which forces a relatively slow

[6] In the limit, the target speaker is the only relevant feature.

variation of speech-related features. Actually, these vary slowly enough to be considered as stationary within short time spans, typically no more than 30ms. One feature vector per segment is thus obtained for each frame, capturing static characteristics of the vocal tract. Adjacent frames are overlapped to obtain a smoother evolution of feature vectors over time.

Different types of features that parameterize speech signals abound in the literature [Furui, 1981; Reynolds, 1994]. Most of them involve the so-called cepstrum transformation, which is a time-domain representation of the log-magnitude spectrum of the speech signal as

$$c_x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log|X(e^{j\omega})|e^{j\omega n}d\omega \tag{1.1}$$

where $|X(e^{j\omega})|$ is the magnitude frequency response of the speech signal $x(n)$ and $c_x(n)$ the corresponding cepstral coefficients. In this domain, any convolutive effect in $x(n)$ is transformed into additive due to properties of the logarithm function.

As accounted in the source-filter model [Fant, 1970], the production of the speech signal can be explained as the result of exciting a filter, which is related to the compound response of the vocal tract cavities, with a source signal produced by the vocal cords as

$$x(n) = e(n) \star h(n) \tag{1.2}$$

involving a convolution in the time domain, or the product

$$X(e^{j\omega}) = E(e^{j\omega})H(e^{j\omega}) \tag{1.3}$$

in the frequency domain. $S(e^{j\omega})$ and $H(e^{j\omega})$ are the frequency responses of the source signal $e(n)$ and filter impulse response $h(n)$, respectively. Including the decomposition in Equation 1.3 into Equation 1.1 yields

$$\begin{aligned} c_x(n) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \log|E(e^{j\omega})|e^{j\omega n}d\omega + \frac{1}{2\pi}\int_{-\pi}^{\pi} \log|H(e^{j\omega})|e^{j\omega n}d\omega && (1.4) \\ &= c_e(n) + c_h(n) && (1.5) \end{aligned}$$

where $c_x(n)$, $c_s(n)$ and $c_h(n)$ are the cepstral coefficients corresponding to the speech signal, source signal and filter. Smooth components in $|X(e^{j\omega})|$, i.e. the spectral envelope, are mapped onto the first coefficients of $c_x(n)$ while the source signal components are mapped onto the last coefficients.

Three methods for computing cepstral coefficients are systematically used in ASV applications:

- **Mel-Frequency Cepstral Coefficients** (MFCC) [Rabiner & Juang, 1993] derive from the cepstrum computation in Equation 1.1 except for several divergences. First, the power spectrum $|X(e^{j\omega})|^2$ is used instead of the magnitude spectrum. Second, a Mel-warped filterbank is used to mimic the non-linear frequency effects of the human auditory system. This rouhly involves integration of the power spectrum over overlapping windows. Finally, a Discrete Cosine Transform (DCT) in computed instead of an Inverse Discrete Fourier Transform (IDFT), which results in uncorrelated real coefficients while improving compactness of the resulting cepstrum.

- **Linear Prediction Cepstral Coefficients** (LPCC) [Atal & Hanauer, 1971] are based on Linear Prediction (LP) to estimate both $e(n)$ and $h(n)$ of the source-filter model. LP uses a Finite Impulse Response (FIR) filter as a predictor for the speech signal $x(n)$

$$e(n) = x(n) - \sum_{p=1}^{P} a(p)x(n-p) \tag{1.6}$$

where $e(n)$ is the prediction error and $a(n)$ the predictor coefficients. $a(n)$ are typically optimized to minimize the Mean Square Error (MSE) of $e(n)$ over a short window of speech by means of the Levinson-Durbin algorithm [Levinson, 1947]. Equation 1.6 can be re-arranged as

$$x(n) = \sum_{p=1}^{P} a(p)x(n-p) + e(n) \tag{1.7}$$

which explains production of the speech signal $x(n)$ in terms of an input signal $e(n)$ and an all-pole filter with recurrent coefficients $a(n)$. Transforming Equation 1.7 into the spectral domain

$$\begin{aligned} X(e^{j\omega}) &= \frac{E(e^{j\omega})}{\sum_{p=1}^{P} a(p)e^{j(\omega-p)}} && (1.8)\\ &= \frac{E(e^{j\omega})}{A(e^{j\omega})} && (1.9)\\ &= E(e^{j\omega})H(e^{j\omega}) && (1.10)\end{aligned}$$

which corresponds to Equation 1.3 if we let $H(e^{j\omega}) = 1/A(e^{j\omega})$. The prediciton filter $A(e^{j\omega})$ and the spectral envelope $H(e^{j\omega})$ are inverse transfer functions while both are determined by $a(n)$. The cepstral coefficients related to the spectral envelope can be directly computed by means of a recursion on the predictor coefficients $a(n)$ [Rabiner & Juang, 1993].

- **Perceptual Linear Prediction** (PLP) [Hermasky, 1990] integrates several psycho-acoustic concepts, i.e. critical-band spectral analysis, equal loudness curve and intensity power law, into LPCC. These enhancements are applied on the power spectrum, which is later transformed back to an autocorrelation function. The optimal $a(n)$ are found using the Levinson-Durbin algorithm on the autocorrelation coefficients and converted into cepstral coefficients using the same recursion as in LPCC. Including perceptual criteria into LPCC has been reported to considerably improve speaker recognition performance using several methods [Vuuren, 1996].

Cepstral coefficients are highly uncorrelated. This property is desirable since it provides a set of non-redundant, i.e. efficient, coefficients in variance terms. Besides, the derived covariance matrices are close to diagonal, which can be used to decrease model complexity.

Up to second or even third order derivatives of cepstral coefficients, i.e. Δ, $\Delta\Delta$ and $\Delta\Delta\Delta$ coefficients, are often taken to include longer acoustic context. By doing so, feature covariance matrices become block-diagonal suggesting adaptation with more complex models. However, using delta features with models with diagonal covariance matrices generally results in improved system performance in practice.

Channel distortion and mismatch is a major concern in ASV applications. The convolutive nature of linear distortion, e.g. due to transduction or channel transmission, has an additive effect on the cepstrum, in a similar way to the source-filter model of speech production. Taking advantage of this property together with the fact that, for the most part, linear distortion is rather stationary, relatively efficient techniques for channel compensation can be conceived in the cepstral domain. In this direction, Cepstral Mean Substraction (CMS) [Atal, 1971; Furui, 1981] considers the channel to be constant and removes part of its effects by substracting the mean of the cepstral coefficients computed over a long window or even the whole speech segment. This is a first-order data centering computing statistics along the temporal axis, i.e assuming ergodicity. The technique is directly extended to second order statistics by normalizing variance of the coefficients to unity, i.e. Cepstral Mean and Variance Normalization (CMVN) [Jain & Hermansky, 2001]. Variance normalization is targeted to improving robustness against additive noise, since a strong negative correlation has been found between additive noise and ceptral coefficient variance.

A popular alternative for improving robustness against channel distortion and noise is de-emphasizing non-speech components. In this line, the Relative Spectral Transform (RASTA) [Hermansky & Morgan, 1994] class of techniques reinforce those components of the amplitude or log-amplitude spectrums corresponding to speech modulation rates while attenuating slower and faster rates. Spectrum dynamics is modified by means of a band-pass filter with band-pass frequencies 1Hz-10Hz which is integrated into a front-end using PLP features. In the log-amplitude domain, RASTA removes convolutional distortion while additive noise is suppressed in the amplitude spectrum variant. J-RASTA is targeted to both phenomena simultaneously, using an alternative spectral transform that behaves as linear for low amplitudes and logarithmic for large amplitudes.

Channel distortion and noise change the statistical distribution of cepstral features. Although the preceding methods indirectly compensate for this phenomenon, none of them aims at directly changing the feature distribution. Feature Warping (FW) [Pelecanos & Sridharan, 2001] is a feature normalization technique that conforms the short-term feature distribution of cepstral features to a target distribution, namely a zero-mean unit-variance Gaussian distribution. The procedure is performed on a frame-by-frame basis on windows of several seconds of duration. Such shaping of the feature distribution aims at compensating both short-term linear channel and noise effects. Mapping onto a Gaussian distribution also improves feature-model coupling for Gaussian mixture based systems.

A major limitation of all of these techniques is the temporal extent of the performed compensation. Since the cepstrum is computed on a frame-by-frame basis, compensation of linear distortion is not possible beyond frame boundaries, e.g. for long impulse responses in cases of strong reverberation and echo. These are usually tackled at the signal level using speech enhancement techniques [Huang *et al.*, 2007] despite complexity and computational burden.

1.4.2 Prosodic Features

We have seen in Section 1.3.1 that temporal patterns of energy and fundamental frequency of the speech signal as well as speech rate and duration and frequency of pauses carry prosodic information characterizing a speaker. However, state-of-the-art systems based on such cues [Atal, 1976; Adami *et al.*, 2003] obtain low performance when compared to those using spectral-envelope features. It is common to include prosodic information into acoustic systems to further improve performance.

Energy is typically obtained by performing some kind of smoothing on the square of the speech signal $x(n)^2$, which is proportional to its instantaneous energy. The smoothing stage is essential since the air expelled by the lungs, slowly varying compared to the sample rate, is at the origin of vocal cord vibration. Exponential or square window FIR filters are often used for this purpose.

Detecting the fundamental frequency of a signal is considerably more tricky due to factors such as non-stationarity or the presence of noise and reverberation. Autocorrelation and normalized cross-correlation time-domain methods [Huang *et al.*, 2001] are amongst the most popular for speech signals. Within a sliding short segment of speech signal, the time lag exhibiting maximal correlation is estimated, thus expecting to predict maximal similarity as well. In the spectral domain, cepstral analysis, the harmonic product spectrum [Noll, 1969] and spectral comb analysis [Martin, 1982] are also used amongst others. In any case, given the spurious nature of instantaneous pitch estimates, a post-filtering stage, typically based on dynamic programming, is further applied.

Speech rate, syllables and pauses can be roughly detected at the signal level from energy or pitch contours. If we are looking for more precise estimates, speech recognition can give a comprehensive solution to detecting speech, non-speech, syllable or phone boundaries from which a variety of prosodic features can be derived.

1.4.3 Speech Activity Detection

The task of separating speech regions from non-speech regions in the signal is called Speech Activity Detection (SAD) or, more generally, Voice Activity Detection (VAD). Only speech is supposed to carry relevant speaker information in the speech signal, so using non-speech regions to compute speaker-related features can result in serious performance degradation due to the noise introduced. Therefore, SAD is an preliminary but essential task in any ASV application.

The main difficulty in SAD is robustness to the virtually unlimited acoustic conditions it must face as neither speech nor background noise can be easily modeled a priori. A wide range of approaches are present in the literature most of them being mostly based on energy [Junqua *et al.*, 1994], pitch or periodicity [Tucker, 1992; Kristjansson *et al.*, 2005] and entropy [Renevey & Drygajlo, 2001] measures as well as distances derived from linear prediction coefficients [Rabiner & Sambur, 1977] and cepstral coefficients [Haigh & Mason, 1993].

Speech enhancement algorithms [Ephraim & Malah, 1984; Ephraim & Malah, 1985; Cohen, 2002] often require SAD for successful operation. They rely on estimation of noise statistics, e.g. noise spectra, which are updated on-line during non-speech regions. These often include advanced SNR estimators into the algorithm [Sohn *et al.*, 1999] which are good indicators of vocal activity. The resulting SAD are robust under stationary noise considerations.

The most common SAD techniques found in current ASV systems include:

- **Energy-based** methods [Lamel *et al.*, 1981; Li *et al.*, 2002] basically use features derived from instantaneous energy estimation of the speech signal. Speech is detected when energy goes beyond a certain threshold, and viceversa. Additional constraints, e.g. multiple thresholds or minimum segment duration, can be set to minimize false alarms and to make sure that word on-sets and off-sets are detected as speech. This approach has the disadvantage of being highly dependent on the acoustic conditions..

- **Voicing features** are often used as an indicator of speech activity in ASV applications. This is a rather conservative approach since it drops unvoiced frames, assuming they are too noisy to reliably bring speaker information. Voicing level is typically determined by pitch estimation methods [Gerhard, 2003; Huang *et al.*, 2001] or using periodicity-related measures such as maximum autocorrelation lag to energy ratio or spectral flatness. Energy and voicing measures are highly correlated in speech signals.

- **Model-based** approaches [Basu, 2003; Sohn *et al.*, 1999] use acoustic models of speech and non-speech classes. It is common to use two Gaussian mixture models, one per class, trained on cepstral and energy features. Frames are assigned to one of the classes based on a likelihood score. Minimum duration constraints can be included using more complex hidden Markov models. Other approaches using linear discriminant functions such as those based on Linear Discriminant Analysis (LDA) have also been explored [Padrell *et al.*, 2005].

- **Forced-alignment** is a particular model-based approach that explicitly models the linguistic content in the speech signal to determine speech and non-speech segment boundaries. Using an orthographic transcription of the speech data to be segmented, features are aligned against the phonemic acoustic models of an ASR system. This is probably the most reliable of the presented methods, but it is language dependent as it relies on the transcription. Alternatively, alignments can be found by phone decoding or using ASR hypotheses.

1.4.4 Score Normalization

It is common to normalize scores to improve calibration robustness against target and test speech variability. Both inter-speaker and intra-speaker variabilities have an effect on the score distribution depending on the speaker, channel or language conditions. If a fixed threshold is used in the decision phase regardless of these factors, decision can not be optimal anymore.

For the most part, score normalization methods focus on inter-speaker variance normalization of impostor (or cohort) speaker score distributions. Given that impostor speech data is much more abundant, collection of impostor speaker statistics is easier and more reliable compared to true speaker statistics.

Scores are typically normalized centering and scaling scores based on mean and variance estimates as

$$\hat{s}(\mathbf{X}) = \frac{s(\mathbf{X}) - \mu}{\sigma} \tag{1.11}$$

where $\hat{s}(\mathbf{X})$ and $s(\mathbf{X})$ are normalized and non-normalized scores respectively. μ and σ are the mean and variance of the impostor speaker score distribution estimated using a cohort speaker set, chosen to represent the variability we are interested in compensating. This approach to normalization assumes that the cohort distribution is Gaussian which is often a good approximation. Several variants are used to estimate μ and σ parameters resulting in different methods:

- **Z-norm** [Li & Porter, 1988; Reynolds, 1997] takes μ and σ as the mean and variance of cohort speaker speech segments $\mathbf{X_{coh}}$ scored against the target speaker model S

$$\begin{aligned} \mu_Z &= E[s(\mathbf{X_{coh}})] && (1.12) \\ \sigma_Z &= \sqrt{\mathrm{Var}[s(\mathbf{X_{coh}})]} = \sqrt{E[(s(\mathbf{X_{coh}}) - \mu_Z)^2]} && (1.13) \end{aligned}$$

  resulting in scores calibrated for the speaker S. Although, scores are more stable after normalization, they are still dependent on speaker S which suggests a different decision threshold for each speaker. If a fixed threshold is used, sub-optimal decisions are made.

- **H-norm** [Reynolds, 1997] is a channel-dependent version of Z-norm. μ and σ are estimated as the mean and variance of cohort speaker speech segments $\mathbf{X_{coh}}$ scored against the target speaker model, with cohort speech data being recorded in the same channel conditions as the claimed speaker speech. As a previous step, channel identification must be performed to label all speech segments. H-norm can be also understood as a score-level channel compensation technique, since it maps a channel-dependent score distribution onto a channel-independent distribution.

- **T-norm** [Auckenthaler *et al.*, 2000] estimates μ and σ as the mean and variance of the test speech segment $\mathbf{X_t}$ scored against a set of cohort models

$$\begin{aligned} \mu_T &= E[s_{ch}(\mathbf{X_t})] && (1.14) \\ \sigma_T &= \sqrt{\mathrm{Var}[s_{ch}(\mathbf{X_t})]} = \sqrt{E[(s_{ch}(\mathbf{X_t}) - \mu_T)^2]} && (1.15) \end{aligned}$$

  thus depending on the test segment. For good normalization, cohort and target speakers are required to be similar in terms of acoustic conditions. Z-norm and T-norm methods are sometimes cascaded to further stabilize score distributions which leads to ZT-norm and TZ-norm variants.

1.4.5 Decision

The test phase of an ASV system scores speech data from an unknown speaker against a target model. To make a decision as to whether the speech sample belongs to the target speaker or not, the score is compared to a threshold. If the score is higher than the threshold the test speech is accepted and otherwise it is rejected. Therefore, a statistical hypothesis test with hypotheses

H_0 : The claimant speaker is an impostor

H_1 : The claimant speaker is the true speaker

is specially suited to such problem.

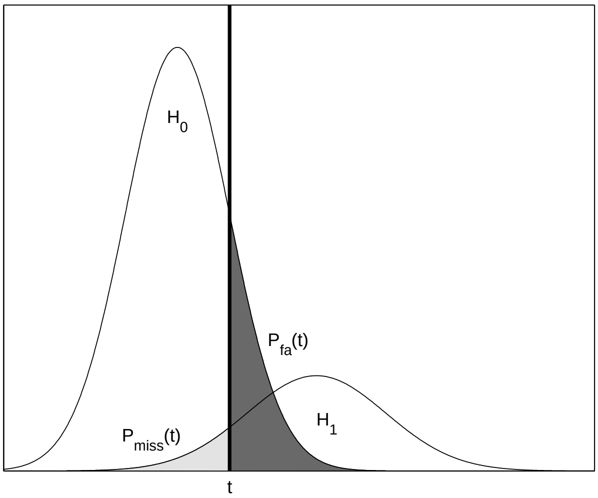

Figure 1.2 — False alarm (P_{fa}) and miss (P_{miss}) probabilities in terms of the decision threshold t.

Letting $\mathbf{X}$ be the ensemble of feature vectors extracted from the test speech segment[7], $s(\mathbf{X})$ the corresponding score[8] using whatever model and t the chosen threshold, the decision is made according to

$$D = \begin{cases} H_1 & \text{if } s(\mathbf{X}) \geq t \\ H_0 & \text{if } s(\mathbf{X}) < t \end{cases} \tag{1.16}$$

Any detection system is subject to errors. Given the two possible hypothesis H_0 and H_1 only four possible outcomes are possible, two of them involving decision errors. Deciding H_1 when H_0 was true is called a false alarm (FA) error, while the opposite is called a false reject (FR) or miss error. They can be defined in probabilistic terms as

$$P_{fa}(t) = p(accept|H_0) = \int_t^{\infty} p(s|H_0)ds \tag{1.17}$$

$$P_{miss}(t) = p(reject|H_1) = \int_{-\infty}^{t} p(s|H_1)ds \tag{1.18}$$

where $p(s|H_i)$ is the conditional pdf for hypothesis H_i. Figure 1.2 illustrates the probability mass involved in the development of $P_{fa}(t)$ and $P_{miss}(t)$.

Associating costs C_{fa} and C_{miss} to the actions taken when false alarms and miss errors occur, we can define the expected Decision Cost Function (DCF) [Brummer, 2004; Duda *et al.*, 2001] in terms of Equations 1.17 and 1.18 as

$$\hat{C}(t) = C_{fa}P_{fa}(t)P(H_0) + C_{miss}P_{miss}(t)P(H_1) \tag{1.19}$$

[7] Note that $\mathbf{X}$ is a sequence of spectral-envelope features for Gaussian models while it can be a single high dimensional vector representing the speaker in Support Vector Machine models.

[8] Gaussian models use log-likelihood scores averaged over the whole speech segment.

It is common to numerically find t that minimizes Equation 1.19 on a training corpus. Calibration is thus fixed for the operating point defined by the choice of C_{fa}, C_{miss} and a priori probabilities $P(H_0)$ and $P(H_1)$. Going further with the analytical optimization of Equation 1.19, we set the derivative of $\hat{C}(t)$ to zero

$$\frac{d\hat{C}(t)}{dt} = -C_{fa}p(t|H_0)P(H_0) + C_{miss}p(t|H_1)P(H_1) = 0 \tag{1.20}$$

yielding

$$R(t) = \frac{p(t|H_1)}{p(t|H_0)} = \frac{P(H_0)C_{fa}}{P(H_1)C_{miss}} = T \tag{1.21}$$

from which the optimal threshold T is obtained in the likelihood-ratio domain in terms of costs and priors too. However, the optimal threshold in the score domain is found by the inverse function of $R(t)$, i.e. $R^{-1}(T)$, which is not known unless we know the conditional pdfs involved. However, since $R^{-1}(T)$ and minimization of Equation 1.19 are equivalent, the former can be approximated by running the expected cost minimization procedure for many values of T since

$$\hat{R}^{-1}(T) \approx R^{-1}(T) = \operatorname*{argmin}_t \hat{C}(t) \tag{1.22}$$

It is thus equivalent to decide in the score domain based on threshold t and score $s(\mathbf{X})$ than to decide in the log-likelihood score domain using threshold T and score $R(s(\mathbf{X}))$. Then, the decision rule in (1.16) can be alternatively rewritten as

$$D = \begin{cases} H_1 & \text{if } R(s(\mathbf{X})) \geq T \\ H_0 & \text{if } R(s(\mathbf{X})) < T \end{cases} \tag{1.23}$$

which lets the user choose the operating point of the system based on costs and a priori knowledge.

1.4.6 System Fusion

ASV system performance can be significantly improved by combining systems exploiting heterogeneous types of information. Although system fusion can be performed at the data, feature, modeling or score levels, we focus in the latter in this Section. Scores are scalar values that integrate the output of different levels of processing before making actual decisions.

Given the scores s_i, $1 \leq i \leq C$, output by C individual systems for one target-test trial, a fused score s_c is obtained as an arbitrary function of the scores, i.e. $s_c = f(s_1, \ldots, s_C)$. The most straightforward form of fusion is score averaging

$$s_c = \sum_{i=1}^{C} s_i \tag{1.24}$$

which assumes s_i is equally distributed from system to system. This can be achieved by score normalization techniques such as T-norm or Z-norm, or by centering and scaling global score statistics otherwise. Although there is no guarantee of improvement, averaging filters out noise in the scores across systems. Note that combination of a very performing system with several low-performing systems can eventually decrease performance, as s_c is dominated by non-reliable scores.

An extension of the previous model is weighted averaging, using a weight w_i for each score as

$$s_c = \sum_{i=1}^{C} w_i s_i \tag{1.25}$$

where w_i roughly accounts for system reliability if $w_i >= 0$. Equation 1.25 can be alternatively seen as a discriminant function of the scores. Applying a threshold on the output score is thus equivalent to splitting the score space using a hyperplane[9]. The weights w_i can be obtained automatically by optimization of a certain criterion over a score database. If true scores are available along with the training data, the model of Equation 1.25 can be trained in a supervised way using back-propagation or Support Vector Machine regression. Non-linear models are easily obtained by applying an activation function, e.g sigmoid, to the weighted average model. An interesting activation function is the logistic function

$$s_c = \frac{1}{1 + \mathrm{e}^{-g(\mathbf{s})}} \tag{1.26}$$

with

$$g(\mathbf{s}) = \sum_{i=1}^{C} w_i s_i + w_0 = \mathbf{w}^T \mathbf{s} + w_0 \tag{1.27}$$

which form a multi-variate logistic regression model for estimation of the posterior probability $P(H_1|s_c)$ [Pigeon *et al.*, 2000]. The optimal weights w_i can be found by maximizing the likelihood of s_c over all labeled scores in a database, assumed to be independent and identically distributed. Logistic regression results in calibrated log-likelihood ratios $\hat{s}_c$

$$\hat{s}_c \approx \log \frac{P(s_c|H_1)}{P(s_c|H_0)} \tag{1.28}$$

for which a single theoretical threshold can be used for optimal classification performance.

More sophisticated approaches focus on score decorrelation [Ferrer *et al.*, 2008b] aimed at improving cooperation among systems or using auxiliary information to guide logistic-regression-based fusion [Ferrer *et al.*, 2008a].

[9] The hyperplane lies in the origin if no offset is used.

1.4.7 Error Measures

Section 1.4.5 defines two types of errors produced by an ASV system, false alarms and miss (or false reject) errors. For a specified threshold t, probabilities P_{fa} and P_{miss} indirectly define the operating point of the system, which is application dependent. A system with a high cost for false alarms, C_{fa}, will set a large threshold to decrease P_{fa} at the price of increasing P_{miss}. Conversely, a system conceived to detect the slightest of the nuances sets a very low threshold, thus increasing P_{miss}. Therefore, a direct measure of system performance must account for both types of errors.

Several measures can be used to globally quantify system performance. The expected detection cost presented in Equation 1.19 measures the so-called Bayesian risk. This cost function includes prior information as well as a quantification of the risk, by means of costs C_{fa} and C_{miss}, taken when making wrong decisions. The Equal Error Rate (EER) is the privileged operating point for which the false alarm rate equals the miss rate, i.e. $P_{fa} = P_{miss}$. Therefore, we obtain a good estimate of system performance as long as P_{fa} and P_{miss} in the final application are relatively balanced. Another alternative measure is the Half Total Error Rate (HTER) defined as the arithmetic mean of false alarm and miss rates, $1/2P_{fa} + 1/2P_{miss}$. The C_{llr} [Brummer & du Preez, 2006] measure has been recently proposed for application-independent evaluation as

$$C_{llr} = \frac{1}{2\log 2}\left(\frac{1}{N_{tt}}\sum_{s(\mathbf{X})\in S_{tt}}(1+\frac{1}{s(\mathbf{X})}) + \frac{1}{N_{nt}}\sum_{s(\mathbf{X})\in S_{tt}}(1+s(\mathbf{X}))\right) \tag{1.29}$$

where $s(\mathbf{X})$ is a score in the form of a likelihood ratio, output by the system, and N_{tt} and N_{nt} are the number of target trials and non-target trials respectively. C_{llr} measures the effective amount of information that the system delivers to the user.

Unless we have a priori knowledge of the application requirements, it is not reliable to evaluate a system based on measures working on a single operating point. So-called Receiver Operating Characteristics (ROC) [Swets, 1964; O'Shaughnessy, 1995] and Detection Error Trade-off (DET) [Martin *et al.*, 1997] curves are threshold-independent representations of system performance over all operating points. ROC curves plot true positive classification rates versus false positive rates. DET curves plot miss rates versus false positive (false alarm) rates on logarithmic scales. Figures 1.3(a) and 1.3(b) show ROC and DET curves for a sample detection problem. We assumed true and impostor scores to be distributed according to two unit-variance Gaussian distributions with distances between Gaussian means of 0, 0.5, 1, 1.5, and 2.

ROC curves are monotonically increasing curves since true positives are correlated with false positives. The worst classification performance (d=0, 100% overlapped Gaussians) leads to a straight line equalling true and false positive rates, meaning all samples were misclassified. Curves increase their curvature as system performance increase. In the limit, the curve becomes a right angle curve, i.e. we obtain 100% true positive rate at any operating point. The area under the ROC curve lies in the range [0.5,1] and it corresponds to system accuracy, i.e. from 50% to 100% for a binary decision problem.

Using logarithmic scales in DET curves spreads out curves which allows observation of slight differences among systems. As illustrated in Figure 1.3(b), curves are decreasing and close to linear, with straight lines corresponding to Gaussian score distributions. Curves move towards the lower-left part of the graph as system performance increases. It is thus common to show only the lower-left quadrant of the graph.

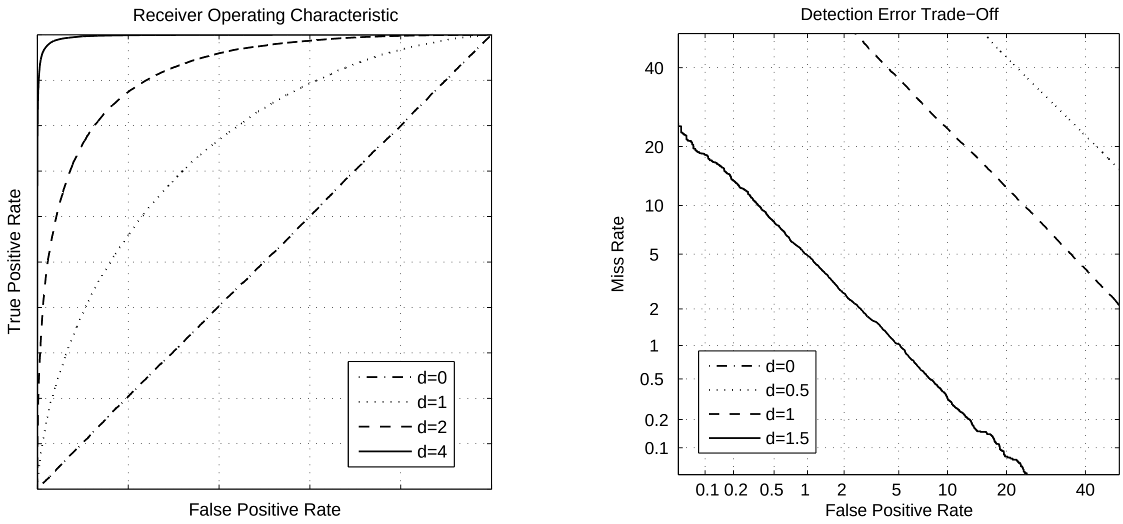

Figure 1.3 — Sample ROC (left, a) and DET (right, b) curves for a binary detection system. Distance between true and impostor score Gaussian means are 0, 1, 2 and 4 respectively.

1.4.8 Corpora

Experimental evaluation of ASV systems requires a speech corpus as well as an evaluation protocol. ASV-specific corpora basically provide speech data for a large amount of speakers from which access trials can be synthesized by taking speech data from pairs of speakers. In case system require so, speech data for impostor speakers can also be found in these corpora. The evaluation protocol proposes one or more performance measures as well as rules and considerations for evaluation.

Several speech corpora are currently available for ASV system performance assessment. These typically include several hundreds of speakers, several sessions per speaker as well as an orthographic transcription of the linguistic content of each session. We list three of them in the following:

- **Polyvar** is a database involving a total of 160h of telephone speech in the French (Swiss dialect) language and over 140 speakers. Multiple sessions are provided for half of the speakers. It consists of heterogeneous types of items including some spontaneous as well as read speech, digits, spellings and dates.

- **TIMIT** is a database primarily conceived for evaluation of speech recognition systems, although given the large amount of speakers it is used for ASV system evaluation as well. It consists of 10 sentences of read speech for a total of 630 speakers covering 8 major dialect regions of the United States. Inter-session variability is limited to a single session per speaker using a wide-band headset.

- **YOHO** is a database involving 138 speakers with 4 enrolment and 10 verification sessions each. Sessions involve 24 spoken phrases for the former and 4 for the latter.

- **Switchboard I**, **Switchboard II** and **Switchboard Cellular** corpora are a data collection effort made by the Linguistic Data Consortium (LDC) towards multi-site system evaluation. Switchboard I involves 2400 two-side topic-oriented conversations of land-line telephone speech for 500 speakers with multiple conversations per speaker. Switchboard II also involves hundreds of speaker and focuses on the use of multiple telephone handsets for proper evaluation of system under channel-mismatched conditions. Switchboard Cellular targets cellular handsets additionally.

- **Mixer** [Cieri *et al.*, 2004; Cieri *et al.*, 2006; Cieri *et al.*, 2007] corpora expand Switchboard series by including multi-language speaker data, namely bilingual English speakers, which allows evaluation of the effect of language mismatch on speaker recognition systems. Certain conversations are simultaneously recorded using eight or more microphones covering close-talk, near-field and far-field microphones. The Mixer 5 corpus contains further channel variation by including six structured interviews of about half an hour for hundreds of speakers.

Please refer to [Martin, 2009] for a detailed discussion on corpora for speaker recognition.

1.5 Summary and Conclusion

This chapter has presented an overview of speaker recognition technology. We have stressed those sources of variability present in the speech signal leading to successful characterization of speakers as well as those remaining serious challenges to current speaker recognition systems. The general structure of ASV systems,which is used by the approaches presented in later chapters, has also been introduced. We have introduced spectral envelope features, thoroughly used in current acoustic ASV systems, as well as speech activity detection, score normalization and decision, other important processing steps not further discussed in the remaining of this manuscript. We have finally enumerated the most widely used performance measures and corpora.

Chapter 2

Generative Approaches

Generative models aim at capturing the statistical distribution of feature vectors, spectral envelope features in acoustic ASV systems. They can be alternatively seen as mechanisms guiding generation of those features, hence the word *generative*. Borrowed from and solidified in the speech recognition field before being used in speaker recognition, two of such models, i.e. Hidden Markov models (HMM) and Gaussian mixture models (GMM), have become ubiquitous in speaker recognition technology during the last decade. Since its introduction, the GMM-UBM paradigm [Reynolds, 2000] has remained the state-of-the-art in text-independent ASV and it has been later been extended giving birth to more sophisticated adaptation techniques dealing with inter-session variability.

This chapter describes HMM, GMM as well as related adaptation techniques that become the basis of later chapters. Sections 2.1 and 2.1.1 give brief introductions to HMM and Continuous Speech Recognition (CSR) respectively, the latter mainly focusing on acoustic modeling. GMM are presented in Section 2.2 as a special case of HMM as well as several adaptation techniques based on them, emphasizing those concerning inter-session variability compensation which is specially relevant to speaker recognition. Section 2.4 describes the GMM-UBM paradigm, an obliged reference in state-of-the-art ASV systems. We finally discuss the use of GMM for intra-speaker variability compensation in Section 2.5. Section 2.6 gives a brief summary and conclusions of the chapter.

2.1 Hidden Markov Models

Hidden Markov Models (HMM) [Rabiner & Juang, 1986; Rabiner, 1989] are finite state machines with random observations and transitions. They are used in many speech applications to model sequence dynamics. Stationary events in a sequence are represented as states, e.g. a part of a word, a phone or a sub-phone unit, and each of these emit features based on an observation probability distribution. Non-stationarity is captured by moving from one state to another according to the state-transition probabilities, so the more states we use the more precise the temporal representation is. An HMM can be characterized by:

- The number of states, N, in the model.
- The transition probabilities a_{ij}, defined as the probability of moving from state S_i to state S_j, that is,

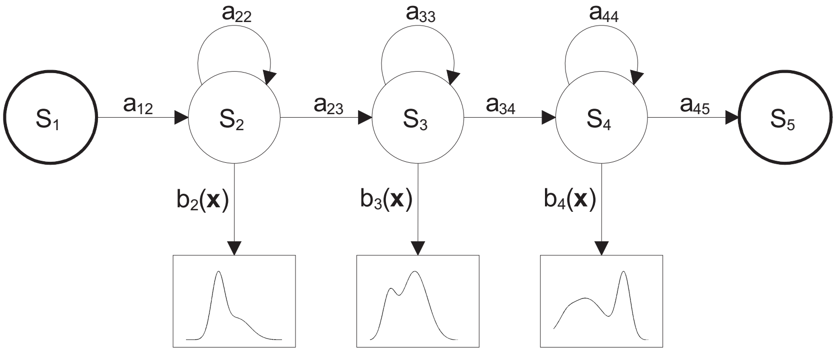

Figure 2.1 — A left-to-right sample HMM diagram.

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i) \quad 1 \leq i, j \leq N \tag{2.1}$$

where q_{t+1} and q_t are states at times $t+1$ and t. Therefore, Markov models are memoryless since going to state q_{t+1} depends on the current state q_t only.

- The emission (or observation) probability distribution

$$b_i(\mathbf{x}) = p(\mathbf{x} | q_t = S_i) \quad 1 \leq i \leq N \tag{2.2}$$

which is dependent on the current state S_i only, thus assuming independence of the observations.

- The initial state distribution

$$\pi_i = p(q_1 = S_i) \quad 1 \leq i \leq N \tag{2.3}$$

We use $\mathbf{\Theta} = (\mathbf{A}, \mathbf{B}, \mathbf{\Pi})$ to compactly refer to a generic HMM, where $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{\Pi}$ globally represent the transition probabilities a_{ij}, emission probabilities $b_i(\mathbf{x})$ and initial probabilities π_i. In the sample HMM shown in Figure 2.1 we can identify some of the mentioned elements, namely transition and emission probabilities. The first and last states can be non-emitting states, i.e. they do not generate any feature vectors. Starting from $t = 1$, each time step generates a transition according to probabilities a_{ij} of the current state S_i and a feature vector is emitted according to the emission probability distribution $b_j(\mathbf{x})$ of the target state S_j.

HMM are typically used in a way different to that just described. Feature vectors taken from a speech signal are actually consumed by the HMM, causing hidden state transitions at each time step t. Observations are thus considered as inputs to the model instead of outputs generated by the model. Based on this conception of an HMM, it can be interesting to

1. compute $P(\mathbf{X}|\mathbf{\Theta})$, the probability of observing the sequence $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ given the model $\mathbf{\Theta}$. Since an exponential number of paths $\mathbf{Q} = (q_1, \ldots, q_T)$ are involved using direct formulation as

$$P(\mathbf{X}|\mathbf{\Theta}) = \sum_{\forall \mathbf{Q}} P(\mathbf{X}|\mathbf{Q}, \mathbf{\Theta}) P(\mathbf{Q}, \mathbf{\Theta}) \tag{2.4}$$

a recursive procedure called the forward-backward algorithm [Baum & Ergon, 1967] is used instead.

2. **decode**, or find the optimal sequence of states $\mathbf{Q} = (q_1, \ldots, q_T)$ given an observation sequence $\mathbf{X}$ and the model $\mathbf{\Theta}$. Optimality is usually sought by maximization of the posteriori probability $P(\mathbf{Q}|\mathbf{X}, \mathbf{\Theta})$, thus

$$\hat{\mathbf{Q}} = \underset{\mathbf{Q}}{\operatorname{argmax}} \, P(\mathbf{Q}|\mathbf{X}, \mathbf{\Theta}) \tag{2.5}$$

The Viterbi algorithm [Viterbi, 1967], which is based on dynamic programming, is used to find such optimal path efficiently.

3. **train**, or estimate the parameters $\mathbf{\Theta} = (\mathbf{A}, \mathbf{B}, \mathbf{\Pi})$ that are more likely to generate the observations in the training data, i.e. those maximizing the likelihood function $P(\mathbf{X}|\mathbf{\Theta})$. The Expectation-Maximization (EM) algorithm [Dempster *et al.*, 1977; Bilmes, 1997], assuring convergence to a local maximum of $P(\mathbf{X}|\mathbf{\Theta})$, is typically used for this purpose.

HMM is ever present in the acoustic models of speech recognition systems. In word-isolated recognition, one model per word is often used, determining the most likely model given the feature vector sequence for a test word in the recognition phase. Word-connected recognition uses linked isolated word models in a similar way. Word models of continuous speech recognition systems are split into smaller units each of which are modeled using HMM. Section 2.1.1 presents a quick overview of Continuous Speech Recognition.

2.1.1 Overview of Continuous Speech Recognition

Continuous speech recognition (CSR) systems are targeted at transcribing words which can be connected with no pauses among them. These systems use utterances as whole speech units instead of isolated words, as utterances are easier to segment in a continous speech flow compared to words. Given the sequence of feature vectors $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ corresponding to an utterance, the CSR problem is often formulated in statistical terms as

$$\hat{W} = \underset{W}{\operatorname{argmax}} \, P(W|\mathbf{X}) = \underset{W}{\operatorname{argmax}} \, \frac{P(\mathbf{X}|W)P(W)}{P(\mathbf{X})} = \underset{W}{\operatorname{argmax}} \, P(\mathbf{X}|W)P(W) \tag{2.6}$$

that is, finding the most likely word sequence W given feature vectors $\mathbf{X}$. On the right-hand side of Equation 2.6, we find that $P(W|\mathbf{X})$ can be factored into two terms, $P(\mathbf{X}|W)$ and $P(W)$. The former is the so-called acoustic model which determines the probability of observing $\mathbf{X}$ given a word sequence W. The latter is the language model which includes a priori knowledge about the sequence W.

With the exception of connectionist approaches [Bourlard & Morgan, 1994], HMM are ubiquitously used to train the acoustic model, i.e. estimate $P(\mathbf{X}|W)$. Given that vocabulary size of CSR systems is on the tens of thousands of words, it is unfeasible to estimate

$P(\mathbf{X}|W)$ based on several repetitions of each possible utterance. Utterances are actually decomposed into words and these into sub-word units, for which abundant speech can be available. Each sub-word unit is modeled using a HMM and word models are obtained by chaining sub-word HMM. The most common sub-word units are tri-phones [Lee *et al.*, 1990], which model left and right phone contexts for improved robustness against phone variability due to co-articulation. Other units such as phones, biphones or demi-phones [Marino *et al.*, 2000] have also been investigated. The HMM of Figure 2.1 illustrates a tri-phone model with its input and output non-emitting states, which help link tri-phone models to make up word models.

Acoustic models are trained in a supervised way by using a set of utterances labeled with the corresponding orthographic transcriptions, i.e. what was said in the utterance or, formally, the true word sequence W. W is decomposed into words and these into its phonetic transcriptions using a pronunciation dictionary, or lexicon[1]. The phoneme string is then decomposed into tri-phone states, resulting in a tri-phone level HMM of W. The corresponding utterance is then used to train the parameters of this word-sequence model.

CSR systems typically use parametric distributions such as weighted Gaussian mixtures as observation probability distributions, e.g. one Gaussian mixture per state in so-called continuous HMM. Given the large amount of acoustic models in a system, the balance between available training data and number of parameters to be estimated can be compromised. Semi-continuous HMM attempt to reduce the number of parameters by using a single Gaussian mixture for all the states. An intermediate approach is tying similar states, e.g. tri-phones with the same central phones, which share the same observation distribution. A tree clustering approach [Hwang & Huang, 1993] based on phonetic knowledge is often used in this case. Optimal state-tying based on distance measures derived from the Gaussian mixture distributions is also used in genonic HMM [Digalakis *et al.*, 1996].

Figure 2.2 shows a diagram of a CSR system in recognition mode. The basic modules of a CSR are feature extraction and decoding, the former computing spectral-envelope features and the latter converting the sequence of features into a sequence of words. The most likely word sequence $\hat{W}$ is searched by exploring the word sequence space. Even with the smallest vocabularies[2] sweeping the whole sequence space becomes prohibitive. It is clear, however, that most of these sequences are very unlikely since we expect language to be structured[3] and not just random words. A tree structure conceptually represents all the possible choices of words as they are being recognized. Different search strategies, such as depth-first, breadth-first, best or beam searches [Huang *et al.*, 2001], can be used to find an approximation of the optimal sequence of words while pruning non promising paths in the search tree. The language model plays a crucial role here by constraining search possibilities so that optimization is feasible and reasonable in terms of space and time resources.

Statistical language modeling approaches found in the literature include tree-based models [Bahl *et al.*, 1989], trellis models [Waegner & Young, 1992] and history models [Black *et al.*, 1992]. However, N-gram language models are by far the most widely spread due to its simplicity and effectiveness. N-gram models estimate the probability of a given word ω_k in terms of the preceding words $\omega_{k-1}\ldots\omega_1$ as

$$P(\omega_k|\omega_{k-1}\ldots\omega_1) = P(\omega_k|\omega_{k-1}\ldots\omega_{k-N+1}) \tag{2.7}$$

[1] The pronunciation dictionary is a table with one or more phonetic transcriptions for every word in the vocabulary.

[2] For an average 5 words per utterance on a 100-word vocabulary, $10^1 0$ word sequences are possible.

[3] Although spontaneous speech often contains hesitations and disfluencies that break language syntax structure.

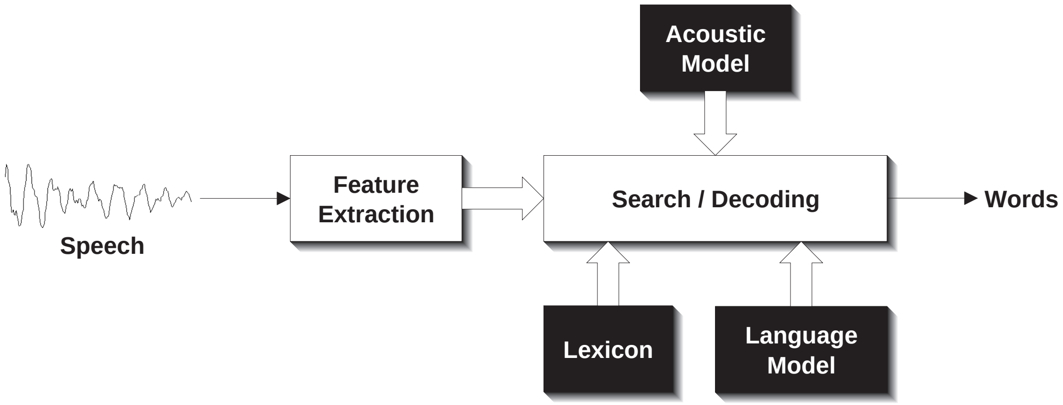

Figure 2.2 — Block diagram of a CSR system.

that is, using the previous N words only. Since words lie in a discrete space these probabilities can be computed as frequency counts on a text corpus. Although N-grams do not directly model the grammatical structure of language, they focus on the local dependence of words, indirectly involving syntax and also semantics. The effectiveness of such a simple model is dependent on the language. Languages that are strict in terms of word order are well-suited to N-grams while languages that allow rather arbitrary phrase or word reordering might require further modeling. Bi-grams, tri-grams and four-grams, with $N = 2$, 3 or 4 respectively, are common choices in CSR or Large Vocabulary CSR (LVCSR) systems. To alleviate the sparsity of high order N-grams ($N = 3$ or 4), techniques such as back off [Ney *et al.*, 1994] or discounting [Katz, 1987] are typically applied.

We refer the reader to the reviews [Young, 1996; Jelinek, 1976] as well as chapter 5, pages 149-189, of [Chou & Juang, 2003] for further insight on CSR and LVCSR.

2.2 Gaussian Mixture Models

Gaussian Mixture Models (GMM) are probability distributions consisting of a linear combination of Gaussian probability density functions. Considering $\mathbf{x}$ as a multivariate random variable such a distribution can be written as

$$p(\mathbf{x}|\mathbf{\Theta}) = \sum_{i=1}^{N} \lambda_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \mathbf{\Sigma}_i) \tag{2.8}$$

where $\mathbf{\Theta}$ is the vector of parameters of the model, i.e. the Gaussian means $\boldsymbol{\mu}_i$, covariance matrices $\mathbf{\Sigma}_i$ and weights[4] λ_i for Gaussian i such that

$$\sum_{i=1}^{N} \lambda_i = 1 \qquad \text{and} \qquad \lambda_i \geq 0 \quad \forall i$$

and $\mathcal{N}(.)$ is the D-dimensional Gaussian probability density function defined as

[4] Sometimes called mixing probabilities.

$$\mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})} \tag{2.9}$$

GMM can be seen as one-state HMM with a Gaussian mixture as observation probability distribution and, thus, they can be trained under the maximum likelihood (ML) criterion using the EM algorithm [Dempster *et al.*, 1977; Bilmes, 1997]. Using a hidden variable representing Gaussian i, the expectation (E) step reduces to the computation of the alignment probability $p(i|\mathbf{x}_t)$, i.e. probability of assigning feature vector $\mathbf{x}_t$ to Gaussian i. Using the Bayes formula and assuming diagonal covariance matrices

$$p(i|\mathbf{x}_t) = \frac{\lambda_i \mathcal{N}(\mathbf{x};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)}{\sum_{j=1}^{N} \lambda_j \mathcal{N}(\mathbf{x};\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)} \tag{2.10}$$

The maximization (M) step computes weight, mean and variance sufficient statistics as

$$\lambda_i = \frac{1}{N}\sum_{t=1}^{T} p(i|\mathbf{x}_t) \tag{2.11}$$

$$\boldsymbol{\mu}_i = \frac{\sum_{t=1}^{T} p(i|\mathbf{x}_t)\mathbf{x}_t}{\sum_{t=1}^{T} p(i|\mathbf{x}_t)} \tag{2.12}$$

$$\boldsymbol{\Sigma}_i = \frac{\sum_{t=1}^{T} p(i|\mathbf{x}_t)(\mathbf{x}_t-\boldsymbol{\mu}_i)(\mathbf{x}_t-\boldsymbol{\mu}_i)^T}{\sum_{t=1}^{T} p(i|\mathbf{x}_t)} \tag{2.13}$$

Expectation and maximization steps are applied iteratively until convergence to a local maximum of the likelihood function, typically a fixed number of iterations. For initialization, it is common practice to use random Gaussian or automatically clustered mean vectors. As for covariance matrices and weights, the identity matrix and a uniform distribution can be used.

The likelihood of a speech segment $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ given a model, i.e. $p(\mathbf{X}|\boldsymbol{\Theta})$, is usually computed in the logarithmic domain to avoid numerical stability problems. Assuming statistical independence of the observations $\mathbf{x}_t$

$$\log p(\mathbf{X}|\boldsymbol{\Theta}) = \frac{1}{T}\sum_{t=1}^{T}\log\sum_{j=1}^{N}\lambda_j\mathcal{N}(\mathbf{x}_t;\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j) \tag{2.14}$$

GMM scoring of long segments can be slow for models using many Gaussians. Faster scoring strategies include downsampling and frame selection [Woszczyna, 1998]. Alternatively, Gaussian selection [Bocchieri, 1993; Woszczyna, 1998] only scores a subset of expected top-scoring Gaussians that are expected to dominate the final score. Other optimizations are proposed in [Chan *et al.*, 2004].

ASV systems use GMM for many purposes, involving spectral-envelope feature modeling for the most part. Some of them are Universal Background models[5] (UBM), speaker-dependent models, feature mapping models for channel compensation, factor analysis models for inter-session variability compensation, MLLR adaptation transforms, and others.

[5] Sometimes also called world model.

2.3 Gaussian Mixture Adaptation

Maximum likelihood estimation as presented in Section 2.2 is a good approach to training GMM when a reasonable amount of speech data are available. As any statistical estimation procedure, the requirements in the amount of training data grow exponentially with respect to the number of parameters, which limits the size of the model used in practice. However, it is possible to relieve the amount of training data required in those situations where some a priori knowledge about the training data is available. In ASV systems, for instance, it is common to train a speaker-independent GMM using speech data from many speakers and adapt its parameters to a particular speaker for which little data are available. The following sections describe three major techniques targeted at this problem, namely Maximum a Posteriori, Eigenvoice and Maximum-Likelihood Linear Regression adaptation, that focus on adaptation of Gaussian mixture observation distributions.

2.3.1 MAP Adaptation

Maximum Likelihood Estimation (MLE), i.e. maximizing $p(\mathbf{X}|\mathbf{\Theta})$, can be extended by including an a priori parameter distribution $p(\mathbf{\Theta})$ leading to optimization based on the Maximum A Posteriori (MAP) criterion. Using the Bayes formula and discarding the constant term $p(\mathbf{X})$, the MAP criterion can be written as

$$\hat{\mathbf{\Theta}} = \underset{\mathbf{\Theta}}{\operatorname{argmax}}\, p(\mathbf{\Theta}|\mathbf{X}) = \underset{\mathbf{\Theta}}{\operatorname{argmax}}\, p(\mathbf{X}|\mathbf{\Theta})p(\mathbf{\Theta}) \tag{2.15}$$

In the case of Gaussian mixtures, the prior distribution $p(\mathbf{\Theta})$ is chosen as the product of a Dirichlet density, which models the Gaussian weights λ_i, and a normal density, modeling mean and variance parameters. Given that mean vectors are placed at the most likely points of each Gaussian component, an efficient way of changing the overall statistical distribution is via mean adaptation. Covariance and weight adaptation are not generally used for speaker recognition. The EM algorithm [Dempster *et al.*, 1977] is used again to derive the mean vector re-estimation formulas [Gauvain & Lee, 1994; Reynolds, 2000]. For adaptation data $\mathbf{X}$

$$\hat{\boldsymbol{\mu}}_i = \alpha_i E_i\{\mathbf{X}\} + (1-\alpha_i)\boldsymbol{\mu}_i \tag{2.16}$$

where $\hat{\boldsymbol{\mu}}_i$ and $\boldsymbol{\mu}_i$ are the adapted and non-adapted mean vectors for Gaussian i, $E_i\{\mathbf{X}\}$ the corresponding expected mean for the adaptation data and

$$\alpha_i = \frac{n_i}{n_i + \tau_i} \tag{2.17}$$

which balances the relevance given to old and new mean estimates. τ_i is the so-called relevance factor, expressed in frames, and n_i is the average number of frames assigned to gaussian i estimated as

$$n_i = \sum_{t=1}^{T} p(i|\mathbf{x}_t) \tag{2.18}$$

When $\tau_i = n_i$, $\alpha_i = 0.5$ thus weighting old and new parameters equally. The larger the τ_i compared to n_i the slower the adaptation. Note, however, that n_i depends on the amount of adaptation data. $E_i\{\mathbf{x}\}$ is the expected feature vector for Gaussian i, estimated as

$$E_i\{\mathbf{x}\} = \frac{1}{n_i}\sum_{t=1}^{T} p(i|\mathbf{x}_t)\mathbf{x}_t \tag{2.19}$$

When a small amount of data is used for adaptation, only a partial observation of the acoustic space is possible. For those regions covered by the adaptation data, α_i is high, given that $p(i|\mathbf{x}_t)$ and n_i are too. The corresponding Gaussians are effectively adapted since data are assumed to be reliable. Conversely, those regions for which no adaptation data is observed are not adapted, trusting the a priori model. Therefore, the more adaptation data is available to a Gaussian the more confidence we give to this data. As more and more data are available, all Gaussians are adapted, asymptotically converging to speaker-dependent ML estimates.

Several variations of MAP adaptation are found in the literature. Maximum Mutual Information MAP (MMI-MAP) [Longworth & Gales, 2006; Povey *et al.*, 2003] switches to a discriminative criterion where the likelihood ratio of the speaker model versus a set of competing speaker models is maximized. Other techniques aim at improving adaptation in scenarios where little amount of data is available. Later described in Section 2.3.2, Eigenvoice MAP [Kenny *et al.*, 2005] models adaptation of the Gaussian mean supervectors, obtained by arranging mean vectors of the GMM into vector form, as the linear combination of a set of speakers which constrain the adaptation space. Since only the linear combination coefficients are estimated, these techniques use a very small amount of adaptation data.

A formulation in terms of Gaussian mean supervectors can be used to alternatively formalize standard MAP adaptation [Kenny *et al.*, 2004] as

$$\hat{\mathbf{m}} = \mathbf{m} + \mathbf{D}\mathbf{z} \tag{2.20}$$

where $\hat{\mathbf{m}}$ and $\mathbf{m}$ are adapted and speaker-independent mean supervectors, $\mathbf{z}$ is a hidden random vector depending on the speaker with an a priori normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\mathbf{D}$ is a diagonal matrix, thus adapting each of the Gaussian mean supervector coefficients independently. Given the a priori model of Equation 2.20, MAP estimates are found by maximizing a modified likelihood function involving the prior distribution of $\mathbf{z}$ as

$$P(\mathbf{X}|\mathbf{m}) = \int P(\mathbf{X}|\mathbf{m}, \mathbf{z})\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})d\mathbf{X} \tag{2.21}$$

by optimization of Equation 2.21 with respect to $\mathbf{z}$ while keeping $\mathbf{D}$ fixed, and viceversa [Kenny *et al.*, 2005]. This EM scheme is then iterated several times as usual in the EM algorithm. Assuming a full matrix $\mathbf{F}$ instead of the diagonal matrix $\mathbf{D}$ leads to the Extended MAP (EMAP) [Zavaliagkos *et al.*, 1995; Jon *et al.*, 2001] variant. This full matrix ties adaptation for all of the Gaussians based on the correlation among them given by $\mathbf{F}\mathbf{F}^T$, which results in adaptation even for those Gaussian that are not observed in the adaptation data.

2.3.2 Eigenvoice Adaptation

In scenarios where little speech data are available, standard MAP adaptation can only adapt few Gaussians in a GMM. In an attempt to cope with this limitation, Eigenvoice adaptation [Kuhn *et al.*, 1998] constrains adaptation in a low-dimension subspace, thus reducing the number of parameters. Assuming N Gaussians and D-dimensional features, a formulation in Gaussian mean supervector terms is given by

$$\hat{\mathbf{m}} = \mathbf{m} + \mathbf{V}\mathbf{y} = \mathbf{m} + \sum_{i=1}^{M} y_i \mathbf{v}_i \tag{2.22}$$

where $\hat{\mathbf{m}}$ and $\mathbf{m}$ are the adapted and non-adapted Gaussian mean supervectors, $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_M)^T$ is the adaptation subspace basis and $\mathbf{y} = (y_1, \ldots, y_M)$ the corresponding set of linear combination coefficients. Adapted speaker supervectors are distributed with mean $\mathbf{m}$ and covariance matrix $\mathbf{V}\mathbf{V}^T$.

Assuming $\mathbf{V}$ is known, adaptation involves estimating the speaker factors $\mathbf{y}$ only. [Kuhn *et al.*, 1998] uses an EM procedure to estimate $\mathbf{y}$ in the maximum likelihood sense, i.e. so-called Maximum-Likelihood Eigen-decomposition (MLED)[6]. The adapted model can then be synthesized using $\mathbf{V}$ and $\mathbf{y}$. The speaker subspace basis, spanned by the columns of $\mathbf{V}$, is a collection of prototype speakers supervectors, so-called Eigenvoices, that constrain the possible directions of adaptation. Eigenvoices can be found by Principal Component Analysis (PCA) [Jolliffe, 1986] of Gaussian mean supervectors for a set of speakers, obtained using speaker-dependent training.

More elaborate approaches aim at estimating $\mathbf{V}$ in the maximum-likelihood sense. The maximum-likelihood eigenspace approach [Gales, 2000; Nguyen *et al.*, 1999], which is also used in Cluster Adaptive Training (CAT), proposes optimization by means of two-step EM optimization. Maximum-likelihood estimates of $\mathbf{y}$ are first computed for all training speakers using current estimates of $\mathbf{V}$. Using $\mathbf{y}$ estimates, the matrix $\mathbf{V}$ that maximizes the product of likelihood functions for all speakers is then sought. In a similar direction, although under the MAP criterion, Eigenvoice MAP [Kenny *et al.*, 2005] assumes that $\mathbf{y}$ is a hidden normally-distributed random vector. Instead of ML estimates, the posterior distribution of $\mathbf{y}$ is calculated first for each speaker. The matrix $\mathbf{V}$ is then estimated by linear regression using $\mathbf{y}$ as explanatory variables. This approach allows estimation of eigenvoices in situations were not enough speech is available for speaker-dependent training, which is often the case in ASV. It also provides the basic framework for estimation of eigenchannel and joint factor analysis model parameters for inter-session variability compensation, discussed in Section 2.5.2. Other alternatives involve Probabilistic Principal Component Analysis (PPCA) [Jolliffe, 1986] extended to the use of Gaussian mixtures [Tipping & Bishop, 1999], which is based on a model that mimics Equation 2.22.

2.3.3 MLLR Adaptation

MAP adaptation performs direct re-estimation of GMM parameters, being one major reason for a relatively slow convergence rate. Fast MAP adaptation techniques, such as Eigenvoices described in the previous Section, perform adaptation based on a parametric transform which is constrained in a low-dimension space. In a similar line, Maximum-Likelihood Linear Regression (MLLR) [Leggetter & Woodland, 1994; Leggetter & Woodland, 1995;

[6] A set of linear equations need to be solved to obtain the speaker factors $\mathbf{y}$.

Gales & Woodland, 1996] adapts the mean vectors of a Gaussian model using an affine transform of the non-adapted means as

$$\hat{\boldsymbol{\mu}} = \mathbf{A}\boldsymbol{\mu} + \mathbf{b} = \mathbf{W}\xi \tag{2.23}$$

parameterized by the linear transform $\mathbf{A}$ and offset vector $\mathbf{b}$, or alternatively, by the linear transform $\mathbf{W}$ and extended mean vector $\xi = (1, \mu^T)^T$. Adaptation consists of finding those parameters that maximize the likelihood of the adaptation data $\mathbf{X}$ given the transformed model. A solution to such optimization problem can be found using the EM algorithm [Leggetter & Woodland, 1995]. Given the adaptation data $\mathbf{X} = (\mathbf{x_1}, \ldots, \mathbf{x_t}, \ldots, \mathbf{x_T})^T$ for $0 \leq t \leq T$, the expectation step computes occupation probabilities $p(r|\mathbf{x_t})$ for each Gaussian R. The maximization step seeks the optimal $\mathbf{W}$ by solving

$$\sum_{t=0}^{T}\sum_{r=1}^{R} p(r|\mathbf{x_t})\boldsymbol{\Sigma_r^{-1}}\mathbf{x_t}\boldsymbol{\xi_r^T} = \sum_{t=0}^{T}\sum_{r=1}^{R} p(r|\mathbf{x_t})\boldsymbol{\Sigma_r^{-1}}\mathbf{W}\boldsymbol{\xi_r}\boldsymbol{\xi_r^T} \tag{2.24}$$

which has a row-by-row iterative solution. Equation 2.24 performs optimization over a set of R Gaussians, with r being a Gaussian index and $\boldsymbol{\xi_r}$ and $\boldsymbol{\Sigma_r}$ the corresponding extended mean vector and covariance matrix.

Covariance matrices can be adapted independently using a transform of the type

$$\hat{\boldsymbol{\Sigma}} = \mathbf{H}\boldsymbol{\Sigma}\mathbf{H}^T \tag{2.25}$$

which forces $\hat{\boldsymbol{\Sigma}}$ preserves positive definiteness. Joint mean and covariance adaptation is performed by iteration of two steps [Gales & Woodland, 1996] by estimating $\mathbf{A}$ and $\mathbf{b}$ first and then $\mathbf{H}$ using the mean-adapted model.

MLLR adaptation of a single Gaussian is never used given that it is equivalent to Gaussian training. Adaptation is performed instead on a set of Gaussians, so-called regression classes, which share the same transformation parameters. Parameter sharing reduces the overall number of parameters to be adapted, improving robustness of the estimation to noise present in the adaptation data. Diverse criteria can be used for to select which Gaussians are adapted. It is common to use a single transform in GMM, although automatic clustering of Gaussians based on acoustic distance measures can be used to obtain an arbitrary amount of regression classes. Alternatively, as proposed in [Karam & Campbell, 2008], we can keep a fixed Gaussian assignment per acoustic class if phonetic labeling of training data is available. When the acoustic models of a CSR system are used, phonemic HMM naturally give rise to more than one acoustic class, by tying together phonetically similar states, for instance. However, such an approach results in a fixed number of regression classes and the amount of data assigned per class may be insufficient for proper adaptation. More sophisticated approaches use knowledge-based or data-driven decision trees [Leggetter & Woodl, 1995] created dynamically from the adaptation data and accounting for the amount of data assigned to each class.

For computation of MLLR transforms using the acoustic models of a CSR system, state alignment can be performed in several ways. In scenarios where an orthographic transcription of the adaptation data is available, the models can be adapted in a supervised way, i.e. force-aligning the adaptation data against the transcripts and given the alignment, estimate MLLR transforms. When this is not possible, the acoustic models can still be adapted based on a hypothesis of the transcripts obtained from a first recognition pass of the CSR system.

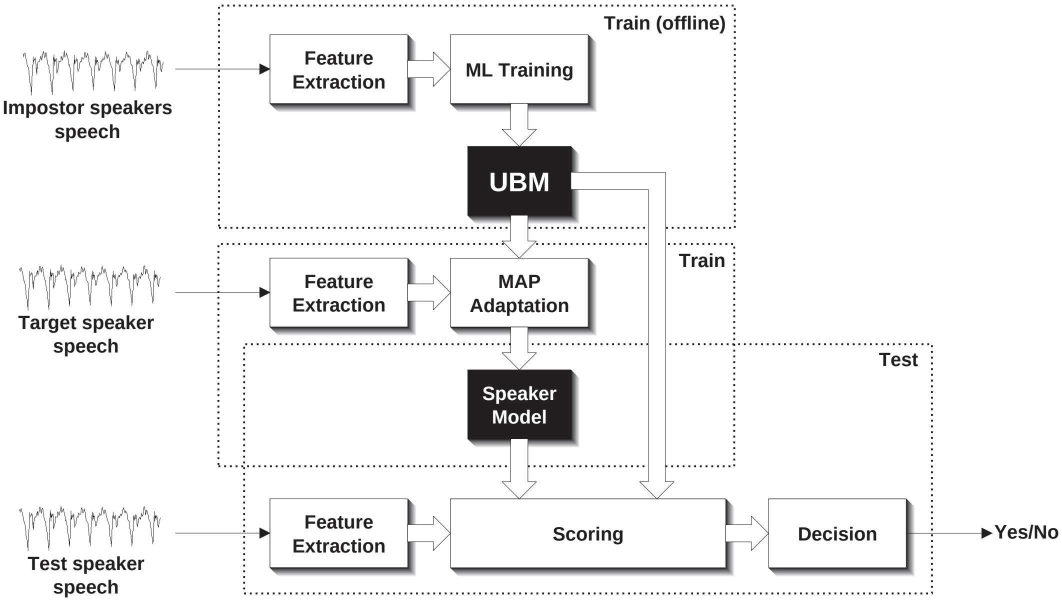

Figure 2.3 — Block diagram of the GMM-UBM system.

Otherwise, although possibly as not as robust, phone-loop alignments can be also used.

2.4 The GMM-UBM paradigm

Currently, the most performing acoustic ASV systems are based on the so-called GMM-UBM paradigm [Reynolds, 1995; Reynolds, 2000]. This approach uses GMM to model and compare spectral-envelope feature vectors uttered by the train and test speakers respectively. However, scarceness of speech data for the target speaker makes direct training of an accurate model rather difficult. The target speaker model is adapted from a Universal Background Model[7] (UBM) instead, taken to represent a priori knowledge about any speaker.

Figure 2.3 shows a diagram of such a system. The train phase assumes that the UBM, trained off-line, is ready for adaptation using the speech data available from the target speaker. The test phase scores test speech against the target speaker model, based on how more likely it is for test speech to be generated by the target model compared to the impostor model. Given a sequence of test speech feature vectors $\mathbf{X}$, scores are obtained by computation of the likelihood ratio of target H_1 and impostor models H_0 as

$$LR(\mathbf{X}) = \frac{p(\mathbf{X}|H_1)}{p(\mathbf{X}|H_0)} \approx \frac{p(\mathbf{X}|H_1)}{p(\mathbf{X}|UBM)} \tag{2.26}$$

where H_0 is approximated by the UBM. The test speech is either accepter or rejected by comparing the likelihood score to a threshold (see Section 1.4.5).

[7] The UBM is sometimes called world model.

2.4.1 Universal Background Model

A strict model of the null hypothesis H_0 should account for all speakers except the target speaker but, given that new impostor speakers can always appear, such task is clearly unfeasible. However, the likelihood function $p(\mathbf{X}|H_0)$ of Equation 2.26 can be reasonably approximated by the Universal Background Model (UBM), trained using many impostor speakers. As more and more speakers are involved, the UBM asymptotically converges to the speaker-independent feature vector distribution $p(\mathbf{X}|H_0)$.

In practice, the required amount of speech and the number of speakers used for UBM training are chosen empirically. Typically, many hours of speech data are used to train a GMM with up to a few thousands of Gaussians. Since the UBM is used as a priori model for target speaker models, knowledge about the type of target speaker data can be used to guide UBM training. For instance, since target speaker is often expected to be scarce, diagonal covariance matrices can be used for more reliable training. If the target speaker population is only female, using only female data for UBM training is likely to result in good adapted models. Similar examples follow from channel, language, and speech style and quality factors.

The UBM is generally trained by maximizing the likelihood function using the EM algorithm. If we are pooling all the speech data together we should make sure that we are balancing all the populations, e.g. gender, channel or language, equally to avoid biasing towards some of the factors explaining the feature distribution. An alternative is to train one UBM per population and later build a compound model from each sub-model.

Given the amount of speech data usually involved, training is often slow, which encourages further optimization. Since we are interested in speaker variability rather than message variability, limiting or downsampling the amount of speech per speaker can be a simple yet effective way to reduce computational cost. This assumes that message variability is averaged out across speakers which seems reasonable if we deal with text-independent tasks.

Variance flooring [Melin *et al.*, 1998] is an important technique used for UBM training. When many Gaussians are present in a GMM, some of them may specialize on a small amount of patterns, namely outlier vectors that are not representative of the overall population. This overfitting phenomenon can be avoided by bounding the minimum variance a Gaussian can be assigned in the maximization step of the EM algorithm. The generalization ability of the GMM is improved while sacrificing optimality of the maximum likelihood estimates.

2.4.2 Speaker Model

To obtain good speaker models, the intra-speaker variability of the speaker should be observed in the training data for the speaker. This often implies sampling the speaker acoustic space with respect to several source of variability such as channel condition or language. Such an approach requires an enormous amount of speech per speaker and, in practice, ASV systems must deal with under-sampled data regarding this variation. Only one recording session per speaker is often available for model training, involving not even enough data to train a GMM from scratch. Adaptation from the UBM is performed instead, specially if a respectable-sized model is sought. Depending on the amount of speech available different adaptation techniques, e.g. MAP, MLLR or any of their variations, can be more or less well-suited. However, MAP adaptation has shown to be the best fit in certain studies [Mak *et al.*, 2006] for about two minutes of speech data.

Several additional points reinforce the choice of MAP adaptation over other approaches. First, robustly estimated parameters in the UBM are used as prior knowledge for the speaker model, as the UBM is trained using many hours of speech. A similar argument follows for intra-speaker variability factors, given that speech data for different conditions can be used for UBM training. Second, the adapted model and the UBM are more comparable as only observed Gaussians are adapted. Third, fast scoring techniques can be applied to compute the likelihood ratio in Equation 2.26 [Reynolds, 2000] by assuming the top-scoring Gaussians to be the same for the UBM and the speaker model.

2.4.3 Scoring

Scoring provides a similarity measure between test speech and the target speaker model. In the GMM-UBM paradigm, scores are given by the likelihood ratio of Equation 2.26 which, using a logarithmic scale, can be computed as the difference of log-likelihood scores with respect the target model and the UBM, hence

$$LLR(\mathbf{X}) = \log p(\mathbf{X}|H_1) - \log p(\mathbf{X}|UBM) \tag{2.27}$$

where both log-likelihoods are evaluated using Equation 2.14. Logarithmic scaling greatly improves numerical stability of the scoring phase.

Speaker models can use many Gaussians since they are adapted from the UBM. Computing Equation 2.27 directly involves $2NT$ evaluations of the function $\mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$, with N being the number of Gaussians and T the length of the test segment. This is computationally expensive, especially for long segments. Assuming that the score is dominated by a few Gaussians only, e.g. the top-scoring ones, we can compute an approximation of the score faster. In practice, fast scoring of $\log p(\mathbf{X}|H_1)$ can be achieved by evaluating the top-scoring Gaussians found while evaluating $\log p(\mathbf{X}|UBM))$.

2.4.4 Score Normalization

Compensating undesired score variability plays an important role in GMM-UBM systems. In principle, likelihood functions are positive but not upper-bounded, evaluating to very large values for Gaussians with very low variance. Although the likelihood ratio computation naturally provides some sort of normalization with respect to the UBM likelihood, scores can still exhibit a large variability from one target model to another. T-norm normalization (see Section 1.4.4) is specially targeted to this type of variation, up to the point that T-norm has become an ubiquitous step in current GMM-UBM systems. Z-norm is not as popular since normalized scores involve an inherent dependency on the target speaker. However, it is more and more common to perform both ZT-norm or TZ-norm for improved stability.

2.5 Inter-session Variability Compensation

Adaptation techniques modify the parameters of a model according to newly observed data, regardless of its origin. If data comes from a different speaker, channel condition or both, the new model inherits these characteristics as well. In practice, any speech segment involves a particular speaker, channel or language thus making adapted models dependent on factors other than the speaker. In this section we discuss two major techniques for compensation of this type of variability namely Feature Mapping as well as Eigenchannel and Factor Analysis approaches.

2.5.1 Feature Mapping

Feature Mapping (FM) [Reynolds, 2003] is a transformation aimed at mapping channel dependent feature vectors onto a channel independent domain, thus removing channel distortion. Assuming a GMM supervector $\mathbf{m}$ can be decomposed into a speaker-dependent contribution s and a channel-dependent contribution $\mathbf{c}$ as

$$\mathbf{m} = \mathbf{m_s} + \mathbf{m_c} \tag{2.28}$$

the channel-compensated model s can be obtained by substracting the channel component $\mathbf{c}$. FM estimates $\mathbf{c}$ directly as the mean vectors of a GMM model trained using speech data from many speakers for the particular channel condition. In practice, channel-dependent models are adapted from a speaker- and channel-independent GMM model called root model. Channel compensation is carried out in the feature domain as

$$\hat{\mathbf{x}} = \boldsymbol{\Sigma}_{\boldsymbol{i}}^{\boldsymbol{UBM}}(\boldsymbol{\Sigma}_{\boldsymbol{i}}^{\boldsymbol{CH}})^{-1}(\mathbf{x} - \boldsymbol{\mu}_{\boldsymbol{i}}^{\boldsymbol{CH}}) + \boldsymbol{\mu}_{\boldsymbol{i}}^{\boldsymbol{UBM}} \tag{2.29}$$

where $\hat{\mathbf{x}}$ is a channel-compensated feature vector and i is the most likely Gaussian assigned to feature vector $\mathbf{x}$ in the channel-dependent model for condition CH. The channel-dependent effects are first removed by the mean vector $\boldsymbol{\mu}_{\boldsymbol{i}}^{\boldsymbol{CH}}$, variance-normalized according to the ratio $\boldsymbol{\Sigma}_{\boldsymbol{i}}^{\boldsymbol{UBM}}(\boldsymbol{\Sigma}_{\boldsymbol{i}}^{\boldsymbol{CH}})^{-1}$ to match root-model variance, and finally centered around the corresponding mean in the root model $\boldsymbol{\mu}_{\boldsymbol{i}}^{\boldsymbol{UBM}}$. Therefore, if the models involved are reliable, all compensated feature vectors assigned to Gaussian i have mean $\boldsymbol{\mu}_{\boldsymbol{i}}^{\boldsymbol{UBM}}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{i}}^{\boldsymbol{UBM}}$.

If more than one channel condition is present, computation of Equation 2.29 is preceded by channel identification, i.e. determining CH. Therefore, FM performs compensation based on a finite set of channel conditions only. For this purpose, the whole segment $\mathbf{X}$ is scored against all channel-dependent models, choosing the one exhibiting the largest likelihood score. Alternatively, a channel identification algorithm that iteratively refines channel membership of speech segments is proposed in [Mason *et al.*, 2005].

FM compensation has given excellent results in scenarios where enough data for the involved channel conditions are available a priori. Compensation beyond already seen channel conditions is addressed by approximating the closer channel component $\mathbf{c}$ in the maximum likelihood sense and applying the corresponding transform. Such an approach models observed channel components only, resulting in poor performance for unseen conditions.

2.5.2 Eigenchannel MAP

Based on the speaker-channel decomposition of Equation 2.28, Eigenchannel modeling finds a channel-induced subspace before channel compensation, allowing for continuous estimates of channel-only contributions. For this purpose, a framework similar to eigenvoice adaptation is used to charaterize channel variation as

$$\mathbf{m_{sh}} = \mathbf{m_s} + \mathbf{U}\mathbf{x_h} \tag{2.30}$$

where $\mathbf{m_{sh}}$ is a supervector depending on speaker s and channel h, $\mathbf{m_s}$ is a speaker-dependent supervector, and the channel supervector $\mathbf{c} = \mathbf{U}\mathbf{x_h}$ is approximated by the linear combination of the basis vectors, i.e. the columns of the matrix $\mathbf{U}$, with the components of $\mathbf{x_h}$ as coefficients. This linear model for the channel component allows adaptation to any new condition in the span of $\mathbf{U}$.

The channel subspace defined by $\mathbf{U}$ constrains channel adaptation to a relatively small amount of directions in the feature space, assuming that most of the channel variability is low-dimensional. This matrix is estimated off-line using a speech database containing a large number of speakers with multiple sessions (or channel conditions) per speaker. As in Eigenvoice adaptation, it can be estimated off-line using standard PCA or, otherwise, the EM algorithm proposed in [Kenny & Dumouchel, 2004] that maximizes the total likelihood of all the segments in the database given the model of Equation 2.30. Once trained, the matrix $\mathbf{U}$ is used for adaptation of the model parameters for any speaker and channel of interest.

The variability model of Equation 2.30 is formally equivalent to that used in Eigenvoice adaptation. However, $\mathbf{m_s}$ is speaker-independent in eigenvoice adaptation whereas it is speaker-dependent in eigenchannel adaptation. Given the small amount of speech and channel variability provided to train a model, usually no more than a few minutes on a single channel condition, standard MAP results in highly biased estimates of $\mathbf{m_s}$. Eigenchannel approaches such as the one in [Vogt *et al.*, 2005] seek joint optimization of $\mathbf{m_s}$ and $\mathbf{x_h}$. In practice, these supervectors are found using an EM-based iterative procedure that optimizes one or the other supervector in an alternate way under the MAP criterion. In such an approach, compensation is embedded in the training phase, as $\mathbf{m_s}$ is the speaker-dependent-only GMM supervector. In the test phase, a similar estimation process is used to find the channel factors $\mathbf{x_h}$ of the test segment, taking the $\mathbf{m_s}$ adapted for the target speaker, i.e. assuming target and test speakers are the same. The likelihood-ratio used in GMM-UBM systems is then computed in the regular way. However, this type of adaptation for test segments requires proper T-norm score normalization as scores depend on a different target speaker for each test speaker. [Vogt *et al.*, 2005] reports further improvements using ZT-norm.

Several variants of this approach are found in the literature. Based on the described eigenchannel approach, similar performance is obtained using the feature-domain compensation scheme proposed in [Vair *et al.*, 2006]. More sophisticated approaches address further modeling of $\mathbf{m_s}$. A straightforward combination of eigenchannel MAP and classical MAP variability models for channel and speaker adaptation results in the factor analysis (FA) [Gorsuch, 1983] model

$$\mathbf{m_{sh}} = \mathbf{m} + \mathbf{D}\mathbf{z_s} + \mathbf{U}\mathbf{x_h} \tag{2.31}$$

where $\mathbf{m}$ is now a speaker- and session-independent mean vector, $\mathbf{D}$ is a diagonal matrix and $\mathbf{z_s}$ a vector of speaker factors, specific factors in factor analysis terms. Based on this model, [Matrouf *et al.*, 2007] proposes a hybrid compensation scheme at the model and feature levels, for target and test speakers respectively, that uses an extension of the eigenvoice MAP estimator described in [Kenny *et al.*, 2005] to both specific factors $\mathbf{z_s}$ and common factors $\mathbf{x_h}$. Symmetrization of the compensation procedure alleviates the requirements in using T-norm score normalization since dependency on the target speaker model is removed for test speaker compensation.

Eigenchannel MAP has proven to be an efficient way to cope with inter-session variability, as competitive state-of-the-art GMM-UBM systems currently include this type of adaptation. However, its complexity renders this technique specially slow for the typical amount of Gaussians, up to a few thousands, currently used. Training of the matrix $\mathbf{U}$, optimized over a whole database, is one major bottleneck. Some optimizations have been proposed about this subject in [Luo *et al.*, 2008].

2.5.3 Joint Factor Analysis

The factor analysis model of Equation 2.31 is a model of utterance variability. Joint Factor Analysis (JFA) [Kenny *et al.*, 2007] uses a similar formal model for speaker modeling purposes as

$$\mathbf{s} = \mathbf{m} + \mathbf{V}\mathbf{y_s} + \mathbf{D}\mathbf{z_s} \tag{2.32}$$

where s is the GMM speaker supervector and the other variables keep the same properties as in Section 2.5.2. However, $\mathbf{y_s}$ and $\mathbf{z_s}$ are now common and specific speaker factors respectively, accounting for eigenvoice MAP and classical MAP estimates of the speaker GMM supervector. An additional model

$$\mathbf{m_{sh}} = \mathbf{s} + \mathbf{U}\mathbf{x_h} \tag{2.33}$$

is required to account for the channel variability observed in the utterance. Equations 2.32 and 2.33 result in a joint model of inter-speaker and inter-session variability. Strategies using the EM algorithm [Kenny *et al.*, 2007] based on those mentioned in Sections 2.3.2 and 2.5.2 [Kenny *et al.*, 2005; Matrouf *et al.*, 2007] for eigenvoice MAP and eigenchannel MAP are used to solve for all channel and speaker factors by linking both models.

JFA reduces to the eigenchannel model of Equation 2.31 when no speaker factors are used, i.e. the term $\mathbf{V}\mathbf{y_s}$ is dropped. Performance of GMM-UBM systems based on JFA adaptation [Kenny *et al.*, 2007] is dependent on the number of speaker and channel factors used. In practice, up to several hundreds of speaker factors are used compared to several tens for channel factors. In general terms, the more speaker factors are used the better performance is obtained, however highly depending on proper score normalization, namely ZT-norm. Otherwise, no large improvements over eigenchannel adaptation have been observed.

2.6 Summary and Conclusion

This chapter has discussed the use of generative models for speaker recognition. Targeting Chapter 6 dealing with (C)MLLR adaptation of phonemic acoustic models, a brief overview of HMM in the context of CSR has been given first. The remaining of the chapter has focused on GMM, obtained by adaptation of a UBM in particular. This is a constant feature found in speaker recognition systems since it allows proper estimation of detailed models of spectral envelope features. The presented adaptation techniques optimize two major criteria, Bayesian (MAP) and maximum likelihood (ML), the former extending the latter by including prior statistics. Those variants using parametric adaptation are better suited for scenarios with a limited amount of adaptation data such as speaker recognition. Other variants such as Eigenchannels or Joint Factor Analysis address inter-session variability compensation in the adaptation process, although requiring multi-session speaker data. The GMM-UBM paradigm still represents the state-of-the-art in speaker recognition especially when using session-compensated adaptation. A system based on such an approach, PLP-GMM, is used as baseline in later chapters of the manuscript.

Chapter 3

Support Vector Machine Approach

Until recently, state-of-the-art ASV systems have been using generative models of cepstral vectors and the GMM-UBM paradigm in particular, currently integrating inter-session variability compensation. These approaches ease the handling of missing data and benefit from the solid maximum likelihood and maximum a posteriori estimation frameworks as well as the experience acquired by the speech community in the field. However, ASV is a classification task that requires actual binary decisions, which suggests the use of boundaries that separate the involved classes. The GMM-UBM paradigm, for instance, incorporates such discriminant conception of the problem in the decision stage, thresholding the likelihood ratios obtained in the scoring phase.

Approaches to ASV using discriminant classifiers give up focusing on modeling speakers, aiming at the classification problem instead. In this line, targeting at accuracy is expected to bring improved accuracy. Support Vector Machines (SVM) [Vapnik, 1998] are binary discriminant classifiers that stand out for its ease in handling high dimensional and sparse data while exhibiting good generalization ability. However, even though SVM fit the binary nature of the ASV problem, they were late introduced probably due to limitations in the amount of data managed by SVM algorithms as well as the issue of dealing with variable-length speech data. Currently, abundance of freely available high-quality packages has made SVM approaches popular in the speaker recognition community, with systems achieving performance comparable to that obtained with GMM-UBM systems.

In this chapter we present the principles of linear and non-linear SVM classifiers and presents the most relevant SVM-based approaches to ASV. Section 3.1 is an overview of SVM covering the soft-margin variant as well as linear and non-linear kernels. In Section 3.2 we introduce the general structure of SVM-based speaker verification systems to later focus on more specific approaches. These address sequence kernels, mapping a feature vector sequence into a vector of fixed dimensionality and parameter-space kernels, using features derived from parameters of generative models. Section 3.3 discusses three techniques for coping with inter-session variability in the SVM feature space. Section 3.4 gives a brief summary and conclusions.

3.1 Support Vector Machines

Support Vector Machines (SVM) are supervised classifiers that use a discriminant function to separate feature vectors from two classes based on the decision surface it defines. Its parameters are optimized to leave the patterns from each of the classes as far as possible from the decision surface. The ASV task lends itself to this binary decision framework by replacing speaker modeling, likelihood ratio computation and score normalization of the GMM-UBM paradigm.

3.1.1 Large Margin Hyperplanes

A binary decision problem can be thought of as assigning a set of feature vectors $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ to the corresponding classes, i.e. a set of binary labels $\mathbf{y}$ with $y_i \in \pm 1, 1 \leq i \leq N$. Discrimination or separation of the input vectors $\mathbf{X}$ according the two classes involves a decision surface that splits the input space in two regions that contain the examples of each class.

A hyperplane (a line in $\mathcal{R}^2$) is a simple form of a decision boundary. In a D-dimensional space, those input patterns $\mathbf{x}$ belonging to a hyperplane $\mathcal{H}$ satisfy

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{3.1}$$

where $\mathbf{w} \in \mathcal{R}^D$ is a vector normal to $\mathcal{H}$ and $b \in \mathcal{R}$ is an offset value, accounting for orientation and shift respectively. When $||\mathbf{w}|| = 1$, the linear term $\mathbf{w} \cdot \mathbf{x}$ can be interpreted as the length of $\mathbf{x}$ in the direction of $\mathbf{w}$ whereas b adds a drift along the same direction, thus allowing a family of parallel hyperplanes. Patterns falling on one side of the hyperplane satisfy $\mathbf{w} \cdot \mathbf{x} + b > 0$ while patterns falling on the opposite side satisfy $\mathbf{w} \cdot \mathbf{x} + b < 0$. Therefore, for any input pattern $\mathbf{x}$, the function

$$\mathrm{f}(\mathbf{x}) = \mathrm{sgn}(\mathbf{w} \cdot \mathbf{x} + b) \tag{3.2}$$

can be used to predict the class that input patterns belong to.

It is common to estimate $\mathbf{w}$ from the training data $\mathbf{X}$ and class labels $\mathbf{y}$. Gradient descent can be used to adapt $\mathbf{w}$ based on the misclassified patterns, as in the perceptron learning rule or a more general approach based on minimum MSE optimization using all of the training samples and target classes (see [Duda *et al.*, 2001] for details).

SVM use a maximum-margin approach to estimating $\mathbf{w}$, i.e. choosing the hyperplane that leaves the closest samples from each of the classes as far as possible. Defining two normalized hyperplanes $\mathcal{H}_{+1}$ and $\mathcal{H}_{-1}$

$$\mathbf{w} \cdot \mathbf{x} + b = \pm 1 \tag{3.3}$$

lying on the closest samples of each class respectively, the maximum margin hyperplane criterion can be stated as

$$\underset{\mathbf{w},b}{\mathrm{argmax}} \ \mathrm{d}(\mathcal{H}_{+1}, \mathcal{H}_{-1}) \tag{3.4}$$

Figure 3.1 illustrates the maximal margin concept. Since $\mathcal{H}_{+1}$ and $\mathcal{H}_{-1}$ are parallel, the distance between them can be written as the difference of their corresponding distances to the origin $b/\,||w||$

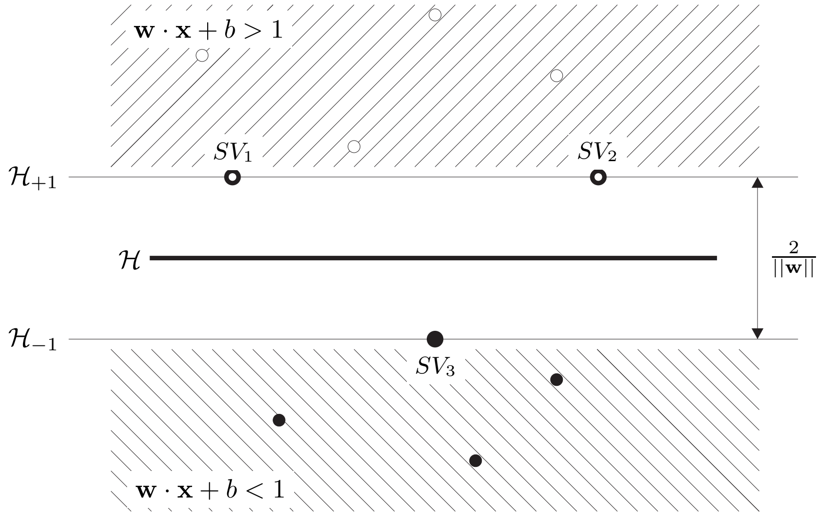
$\mathbf{w}\cdot\mathbf{x}+b>1$
SV_1
SV_2
$\mathcal{H}_{+1}$
$\mathcal{H}$
$\frac{2}{||\mathbf{w}||}$
$\mathcal{H}_{-1}$
SV_3
$\mathbf{w}\cdot\mathbf{x}+b<1$

Figure 3.1 — Margin maximization in SVM. $\mathcal{H}$ is the optimal margin hyhyperplane, and $\mathcal{H}_{+1}$ and $\mathcal{H}_{-1}$ the margin hyperplanes. Support vectors SV_1, SV_2 and SV_3 lie on the margin hyperplanes.

$$\mathrm{d}(\mathcal{H}_{+1},\mathcal{H}_{-1}) = \frac{1}{||w||} - \frac{-1}{||w||} = \frac{2}{||w||} \tag{3.5}$$

Maximization of the classification margin $\mathrm{d}(\mathcal{H}_{+1},\mathcal{H}_{-1})$ is thus equivalent to minimization of $||w||$. Assuming the data is linearly separable, the optimal solution for $\mathbf{w}$ must classify all the patterns in $\mathbf{X}$ resulting in the constrained problem

$$\begin{array}{rl} \text{argmin}_{\mathbf{w}} & \frac{1}{2}\,||w||^2 \\ \text{subject to} & y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 1 \quad 1\leq i\leq N \end{array} \tag{3.6}$$

The primal optimization problem of Equation 3.6 is a convex quadratic optimization problem involving as many inequality constraints as training examples $(\mathbf{x}_i, y_i)$. The following section discusses how to address this problem.

3.1.2 Solving the Dual Problem

Constrained optimization problems can be solved using the method of Lagrange multipliers which, in practical terms, augments the objective function by incorporating a linear combination of the constraints. The resulting auxiliary function is called the Lagrangian function which, for the problem of Equation 3.6, is

$$\mathcal{L}(\mathbf{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\,||w||^2 - \sum_{i=0}^{N}\alpha_i(y_i(\mathbf{w}\cdot\mathbf{x}_i+b)-1) \tag{3.7}$$

where α_i are the so-called Lagrange multipliers, lower-bounded as $\alpha_i \geq 0$, since we are seeking to minimize $\mathcal{L}(\mathbf{w},b,\boldsymbol{\alpha})$. Setting the derivative with respect to the primal variables $\mathbf{w}$ and b to 0 yields conditions

$$\mathbf{w} = \sum_{i=0}^{N} \alpha_i y_i \mathbf{x}_i \tag{3.8}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{3.9}$$

which states that the optimal hyperplane defined by $\mathbf{w}$ can be expanded as a linear combination of the training patterns. Introducing this expansion into Equation 3.2 yields

$$\mathrm{f}(\mathbf{x}) = \mathrm{sgn}(\sum_{i=0}^{N} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b) \tag{3.10}$$

which is the decision function in terms of the training patterns. Those patterns with $\alpha_i = 0$ have no contribution in the expansion. In fact, the hyperplane is synthesized only using those samples $\mathbf{x}_i$ that satisfy $\alpha_i > 0$, which correspond to those samples that lie on any of the two margin hyperplanes $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1$, so-called support vectors.

Equation 3.8 also shows that the optimal hyperplane does not depend on the dimensionality, D. This one the reason for which SVM can exhibit good generalization ability even in high-dimensional feature spaces. Keeping a relatively small number of free parameters, the support vectors, regardless of D prevents overtraining by upbounding the complexity of the optimal hyperplanes. It is common to use SVM to classify undersampled data, i.e. where less training examples than dimensions are available, situations in which other methods result in structural overfitting or algorithm breakdown.

Including Equations 3.8 and 3.9 into $\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha})$ leads to an equivalent formulation, the so-called dual problem, of the optimization problem of Equation 3.6

$$\begin{array}{rl} \mathrm{argmax}_\alpha & \mathrm{W}(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ \text{subject to} & \alpha_i \geq 0 \quad \text{with} \quad 1 \leq i \leq N \\ \text{and} & \sum_{i=1}^{N} \alpha_i y_i = 0 \end{array} \tag{3.11}$$

which can be shown to have the same solutions. This is a quadratic program where only variables α_i need to be optimized. $W(\alpha)$ is a convex function with a single minimum in the regions defined by the constraints. Given the optimal α, the optimal hyperplane is found using Equation 3.8 and b by averaging

$$b = y_j - \sum_{i=1}^{N} y_i \alpha_i \mathbf{x}_i \cdot \mathbf{x}_j \tag{3.12}$$

over all training patterns $(\mathbf{x}_j, y_j)$, $\forall j$ such that $\alpha_j > 0$. Equation 3.12 is obtained by adjusting b so that y_j is correctly classified.

For problems with a small amount of training samples, the dual problem in Equation 3.11 can be solved using any convex quadratic program solver. Larger problems can be solved using a variety of optimization techniques, which have in common the partial handling of training data at a time [Boser *et al.*, 1992]. Projection methods [Moré & Toraldo, 1998] use iterative line search and projection onto the region defined by the constraints from an initial solution estimate α out of that region. However, this method involves inversion of the Hessian matrix $\mathbf{H}_{ij} = y_i y_j \mathbf{x}_i^T \mathbf{x}_j$ which is prohibitive for large scale problems involving in the order of 10^4 constraints. Subset Selection methods [Scholkopf & Smola, 2002] are popular for large scale problems where a large amount of support vectors are expected to be found or where precision in the support vectors found is not a priority. In these situations, the optimization problem is iteratively split into less complex subproblems each of which is solved at a time. For instance, chunking-based approaches choose a partition of the training data and use a suitable algorithm to solve for the support vectors, while discarding the rest. The latter are then replaced with non-observed data to retrain the model. Alternatively, working set algorithms [Osuna *et al.*, 1997] perform optimization over a subset of variables at each iteration while keeping the rest fixed. Only the set of constraints involving the variables in the working set must be considered for optimization. Sequential Minimal Optimization (SMO) [Platt, 1999] uses a chunking approach by selecting subsets of size two, for which the optimization problem can be solved analytically. This is the most popular method for solving large-scale problems as it is easily implementable and has very low memory requirements.

3.1.3 Linear Separability and Soft Margins

Both optimization problems of Equations 3.6 and 3.11 assume that the patterns $(\mathbf{x}_i, y_i)$, $1 \leq i \leq N$ are linearly separable, i.e. a single hyperplane is able to perfectly separate the populations of the two classes. It is rarely the case that we encounter such a problem. In practice, a certain amount of overlap between the populations is present and, even otherwise, patterns may be not representative of their respective population or patterns may be even corrupted by noise. A striking situation is found for outlier patterns, as they may dramatically change the optimal hyperplane solution, given that it depends on a small amount of support vectors.

A modification can be introduced into the both primal and dual optimization problems to account for those patterns that cannot be correctly classified. Since misclassified patterns satisfy

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \leq 0 \tag{3.13}$$

one way to continuously account for all classification situations is relaxing the constraints by introducing so-called slack variables ξ_i as

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \tag{3.14}$$

where $\xi_i > 0\,, \forall i$. As shown in Figure 3.2, for values of $\xi_i = 0$, the pattern is correctly classified. As ξ_i approaches 1, the pattern goes through the margin error and for $\xi > 1$ a classification error occurs. The three regions defining these classification errors are illustrated in Figure 3.2. Since the constraint can be always satisfied by letting ξ_i grow large, an average penalty term, e.g. the mean classification error, is included in the objective function. This leads to the primal problem

$$\begin{array}{lll} \mathrm{argmin}_{\mathbf{w},\boldsymbol{\xi}} & \frac{1}{2}\left\|w\right\|^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i & \\ \text{subject to} & y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i & 1 \leq i \leq N \\ & \xi_i > 0 & 1 \leq i \leq N \end{array} \tag{3.15}$$

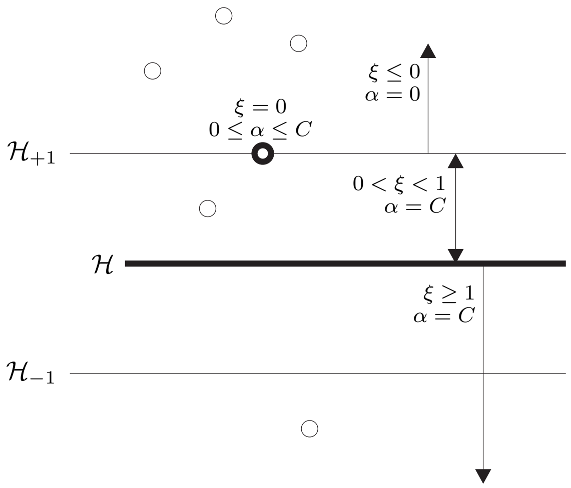

Figure 3.2 — Classification error in soft-margin SVM (C-SVM). The three regions correspond to different types of error as captured by the slack variable ξ. Support vectors result in positive α in the dual optimization problem, upbounded by C in C-SVM. Symmetric regions are found for the opposite class.

where a cost C is assigned to each classification error. The optimization problem has been enlarged and we now seek optimal values for both $\mathbf{w}$ and $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_N)^T$. The dual problem results in minimal modifications. Basically, the Lagrange multipliers α_i become upper-bounded too, so that $0 \leq \alpha_i \leq C/N$ (see [Scholkopf & Smola, 2002] for more details).

3.1.4 Non-linear Classifiers

Section 3.1.3 presented a way to deal with non-linearly separable data. Another alternative for such a problem is to use more flexible decision boundaries that can adapt to the nature of the problem. The non-linear kernel SVM approach described here assumes that modifications in the decision boundary can be equivalently performed on the input patterns.

The underlying idea is to transform patterns from the input space to a high-dimensional space. Although this mapping is somehow artificial, we expect discrimination to be easier if the mapping is made in a non-linear way. Say Φ: $\mathbf{x} \longrightarrow \Phi(\mathbf{x})$ is such a mapping function, resulting in an increased dimensionality of $\mathbf{x} \in \mathcal{R}^D$ up to $\Phi(\mathbf{x}) \in \mathcal{R}^H$. As stated in Equation 3.11, solving the optimal hyperplane can be done in terms of dot products. If we expand the training patterns to this high-dimensional space, we can solve the dual problem just by replacing any dot product $\mathbf{x}_i \cdot \mathbf{x}_j$ by $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$. However, an explicit expansion of the patterns can be expensive in terms of time and space resources, being out of reach for very high-dimensional spaces. Since we are interested in the dot product value and not in the expanded patterns themselves, we can bypass their explicit computation if a function $k(\mathbf{x_i}, \mathbf{x}_j)$ exists such that

$$k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \tag{3.16}$$

Therefore, the optimization problem in the linear case can be used for the non-linear case at the price of evaluating kernel functions instead of regular dot products. A decision function similar to the linear case

$$\mathrm{f}(\mathbf{x}) = \mathrm{sgn}(\sum_{i=0}^{N} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b) \tag{3.17}$$

is derived, where the support vectors are $\Phi(\mathbf{x}_i)$, which are not explicitly computed either for those patterns that satisfy $0 < \alpha_i < C$.

Non-linearity is encoded in the choice of the kernel function. Being inspired on regular dot products, kernel functions are also similarity measures between two patterns. Thus, using different kernel functions sensibly leads to different optimal hyperplanes as well as different classification accuracy. Besides the linear kernel, i.e. a dot product, the most common vector kernels are polynomial, Gaussian and sigmoid kernels. As similarity measures, though, kernels can take more complex data structures such as Gaussian models, graphs or trees. All of these kernels satisfy the so-called Mercer's condition, which states that the kernel function must be positive definite as

$$\int_{\chi^2} k(\mathbf{x}, \mathbf{y})\mathrm{f}(\mathbf{x})\mathrm{f}(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0 \tag{3.18}$$

where $\mathbf{x}$ and $\mathbf{y}$ are arbitrary patterns in vector space χ and $\mathrm{f}(x)$ is any finite-energy function in a Hilbert space. Mercer kernels are real-valued and symmetric. These properties automatically render the derived kernel matrix[1] positive-definite and invertible, a requirement for some quadratic program solvers.

The use of non-linear kernels has the effect of adapting the decision boundary to the training data. However, as we choose more and more powerful expansion functions Φ, more and more patterns in the training data are successfully classified, potentially learning the whole training set while still exhibiting poor generalization to other data. This overtraining (or overfitting) phenomenon demands a compromise between training accuracy and machine complexity. A reliable way to account for this issue is to evaluate the quality of the model, e.g. in terms of accuracy, on an unseen test data set. The soft-SVM approach inherently balances training accuracy and model complexity by minimizing the so-called regularized-risk. The first term in the objective function of Equation 3.15 is $||\mathbf{w}||$, i.e. the length of $\mathbf{w}$ which is related to the number of support vectors found in the training phase. The second term is the overall cost of misclassification errors of the training set, so-called empirical risk. Therefore, for the right choice of the parameter C, both training accuracy and model complexity can be simultaneously minimized.

3.2 SVM Approaches to ASV

The binary nature of SVM classifiers seems well-suited to the ASV task. However, a decade has been necessary for the speaker recognition community to adopt them. Beyond reluctance of switching to a new approach, there are powerful reasons for such a delay. One might consider, for instance, a system using spectral envelope features classified using SVM, in the same way as in generative models. Performance of such an approach does not compare to that obtained for generative approaches besides it does not scale well either. We discuss how these issues are overcome in the following.

[1] The kernel matrix, or Gram matrix, results from the evaluation of the kernel function over all patterns in a dataset arranged in matrix form.

Speech segments typically used in text-independent ASV systems have a minimum duration of a few minutes. This quickly represents over 10000 feature vectors for as little as 2 minutes of effective speech, linearly increasing with respect to the duration. Accounting for a minimum of one target speaker segment to be classified plus impostor speaker segments, i.e. feature vectors for negative speakers, the system rapidly requires tens or hundreds of thousands of feature vectors to be operational, or otherwise use very short segments. Even in this situation, unless SMO-based methods[2] are used, the SVM optimization problem can not be solved. Therefore, directly classifying feature vectors using SVM is not feasible for ASV tasks of this size.

A second issue with the conception of such a system is model complexity. Assuming it is feasible to solve the SVM problem for the amount of data we are given, the amount of parameters used in the SVM can be some orders of magnitude lower than the amount used in generative models, which gives a rough idea of the difference in model complexity. Feature vector distributions are intricate and involve class overlap due to a large amount of intra-speaker variability (see Section 1.3.2). This suggests that a single decision boundary, even if non-linear, be not able to successfully discriminate the two classes as desired.

Given that SVM are gifted for classification of high-dimensional vectors, current approaches to ASV using SVM convert the time-dependent nature of spectral envelope feature vectors into one or few high-dimensional feature vectors per speaker. These hopefully collect speaker-relevant information that is compacted in a time-independent representation. Using one feature vector per speaker reduces the number of constraints in the SVM optimization problem down to the sum of target plus impostor speakers used in the system, for which optimization is feasible.

Following these directions, diversity of SVM approaches to ASV mainly focus on the design of speaker-relevant features, including strategies for mapping time-varying features to fixed-size vectors, techniques that increase their inter-speaker variability and decrease inter-session variability and feature normalization. Some of these can be alternatively seen as deriving from new kernels adapted to speech-related features. In the following sections we present the global structure of SVM-based systems later focusing on the current state-of-the-art approaches found in the literature.

3.2.1 Structure of SVM-based Systems

ASV systems using SVM tend to share a common structure which is conceptually close to that of GMM-UBM systems. As illustrated in Figure 3.3, high-dimensional feature vectors are computed in the training phase from the cepstral features of all the speech segments involved in the system, i.e. impostor, targets and test speakers, according to the approach used. For this purpose, as long as it is feasible in terms of computational resources, the explicit expansion $\Phi(\mathbf{x})$ of the kernel is often used. This brings flexibility to further process the feature vectors before classification with techniques that can not be incorporated into the kernel function. A feature normalization step is also included before SVM training to ease optimization, avoiding conditioning issues in the Hessian matrix, for instance. A SVM model is obtained for each target speaker segment of interest using all of the impostor speakers. In the test phase, scores are computed as the distance of test feature vectors to the optimal hyperplane using the decision function of Equation 3.17.

Much like all machine learning techniques, the choice of speaker data is highly relevant in SVM-based systems. In particular, only feature vectors from speakers others than the

[2] SMO methods appeared in 1999 [Platt, 1999].

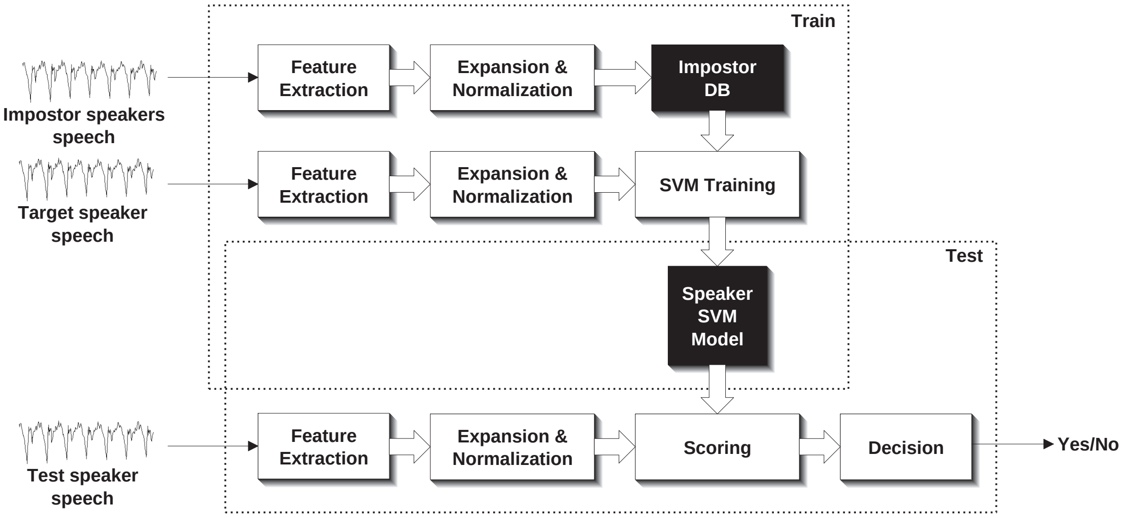

Figure 3.3 — Block diagram of a generic SVM-based system.

target speaker should be used as impostor speakers. Otherwise, conditioning problems in the Hessian matrix may render the SVM optimization problem ill-posed, leading to unpredictable results. This is still something to avoid in GMM-UBM systems although not as critical as in SVM. Matching acoustic conditions for target and test speakers also reduces the span of the problem, generally resulting in improved classification accuracy. However, it is reasonable to assume that mismatched-condition feature vectors lie further in average than matched-condition vectors. Based on this interpretation, since SVM use only those feature vectors closest to the decision boundary, i.e. support vectors, mismatched-condition vectors are unlikely to be included in the optimal hyperplane expansion anyway. Another consideration is balancing the amount of training samples for the target (+1) and impostor (-1) classes. A large asymmetry in the amount of training data has the effect of biasing the score distribution of the function $\sum_{i=0}^{N} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b$, that is, before a threshold is applied. This shift implies re-calibration of the decision threshold of the system. Several techniques are found in the literature to cope with SVM score bias due to unbalanced training data [Tao *et al.*, 2005; Li *et al.*, 2008].

Feature normalization is commonly used prior to SVM training. The methods found in the literature assume independence of feature components and focus on component-by-component normalization. Data centering and reduction across impostor speakers is one major method which forces each component to have zero mean and unit variance. Centering and scaling can also be used aiming at normalization of the kernel matrix. By fitting the vector components in the range $[-\frac{1}{\sqrt{D}}, +\frac{1}{\sqrt{D}}]$, where D is the vector dimensionality, we make sure that the absolute value of any dot product evaluates to 1 at most. Alternatively, rank normalization [Shriberg *et al.*, 2004; Stolcke *et al.*, 2008] has proven to be effective for a wide range of features. This method is analogous to histogram equalization, where the feature vector component distribution across impostor speakers is mapped onto a uniform distribution in a non-parametric way. The resulting transform is then applied to every feature vector component. High-density areas of the distribution are stretched and low-density areas are shrunk. Unlike the previous techniques, rank normalization does not assume the distributions are Gaussian. For feature vectors containing frequency count information, the Term Frequency Log Likelihood Ratio (TFLLR) [Campbell *et al.*, 2000] kernel and its generalized version TFLOG [Campbell *et al.*, 2007] can also be used. These compensate for n-gram sparsity by weighting low-count components with large weights and conversely for high-count components.

The great majority of approaches in the speaker recognition community use linear-kernel SVM. As already mentioned, the high-dimensional features used are the result of the explicit expansion related to some kernel function. In this context, the use of non-linear kernels is often unnecessary and typically brings no additional gains. In linear-kernel SVM, only the misclassification cost C needs to be tuned. Although it is optimized by validation on a test set, members in the community informally agree that systems tend to use large values for C, forcing the SVM to optimize based on misclassification error only.

It is common to use some implementation tricks that can speed up the system considerably. Training can be sped up enormously by caching the kernel matrix $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. This matrix is symmetric and, in situations where a few vectors are used for the target speaker, most of the kernel evaluations involve impostor speakers. Since impostor speakers are kept the same from one target speaker to another, only those rows and columns affecting the target speaker segments need to be updated, while the rest of the matrix can be computed only once. As for test, when using linear kernels, the optimal hyperplane equation can be collapsed so that it does not depend on the dot product with the support vectors anymore. The function $\sum_{i=0}^{N} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b$ can be re-written as $\mathbf{w} \cdot \mathbf{x} + b =$ using Equation 3.8. Thus, we explicitly compute $\mathbf{w}$ as $\sum_{i=0}^{N} \alpha_i y_i \mathbf{x}_i$ after training. By doing this, scoring reduces to a single dot product whereas regular scoring requires as many dot products as support vectors.

3.2.2 Sequence Kernels

Speech data is available for each speaker and session in the form of a variable-length feature vector sequence. This can be a suitable representation of a speaker for those systems where modeling and scoring are done on a frame-by-frame basis, e.g. GMM-UBM systems. As discussed in the introduction of the current section, such an approach is not feasible using SVM classification, and the feature vector sequence is mapped onto a fixed-length vector instead. By this, we aim at classifying the whole sequence as belonging to the target or an impostor speaker. This section discusses several alternatives to perform this mapping.

3.2.2.1 Generalized Linear Discriminant Sequence Kernel

Linear discriminant classifiers, e.g. linear-kernel SVM, use a function of the form $\mathrm{f}(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$ to decide upon one of two possible hypotheses. This family of classifiers can be generalized by transforming the input vector $\mathbf{x}$ using an arbitrary vector function $\Phi(\mathbf{x})$. According to this extension, an analogous generalized linear discriminant function (GLDF) can be written as $\mathrm{g}(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) + b$. Quadratic and higher-order monomial expansions of $\mathbf{x}$ are examples of $\Phi(\mathbf{x})$ that lead to so-called polynomial classifiers.

The Generalized Linear Discriminant Sequence (GLDS) kernel [Campbell, 2002] maps a sequence of T feature vectors $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T]^T$ into a fixed-length vector

$$\overline{\mathbf{\Phi}}(\mathbf{X}) = (\overline{\Phi}_1(\mathbf{X}), \ldots, \overline{\Phi}_K(\mathbf{X}))^T \tag{3.19}$$

by appending the outputs of a set of discriminant functions $\overline{\Phi}_k(\mathbf{X})$. Here, overline denotes mean expectation of the expansions $\Phi(\mathbf{x})$ computed on every feature vector in the sequence, therefore,

$$\overline{\Phi}_k(\mathbf{X}) = \frac{1}{T}\sum_{t=1}^{T}\Phi_k(\mathbf{x}_t) \tag{3.20}$$

Based on this mapping, a kernel function can be derived as the dot product the expansions of two arbitrary sequences $\mathbf{X}$ and $\mathbf{Y}$ as

$$k(\mathbf{X},\mathbf{Y}) = \overline{\boldsymbol{\Phi}}(\mathbf{X})\mathbf{R}^{-1}\overline{\boldsymbol{\Phi}}(\mathbf{Y}) \tag{3.21}$$

which additionally decorrelates the frame-by-frame expansions $\boldsymbol{\Phi}(\mathbf{x}_t)$. Systems using this kernel are typically implemented using explicitly expanded vectors and a regular linear kernel. $\mathbf{R}$ is estimated from a training data set whose expanded vectors have been computed. Inversion of $\mathbf{R}$ for expansions in high-dimensional spaces is not feasible. It is common to consider it as a diagonal matrix, which reduces to component-by-component variance normalization and speeds up computation time enormously. Note that the term "sequence kernel" is somewhat misleading since any permutation of the feature vectors leads to the same mean expanded vector.

It is fairly usual to use monomial expansions as $\Phi_k(\mathbf{x}_t)$, which reduces to computing all possible cross-products of $\mathbf{x}_t$ up to a certain order P. In this case, Equation 3.20 can be interpreted as the estimation of the statistical moments of a random vector $\boldsymbol{\Phi}(\mathbf{x})$. Given that the number of dimensions of the expanded vectors is $\frac{(D+P)!}{D!P!}$ orders higher than 3 become prohibitive.

Explicit mapping limits the GLDS approach to finite-dimension expansions. A generalization to high-dimension and infinite-dimension expansions is presented in [Louradour *et al.*, 2006] which uses the empirical kernel map technique [Scholkopf & Smola, 2002]. In this approach, $\overline{\boldsymbol{\Phi}}(\mathbf{X})$ is approximated in terms of a finite expansion of B feature vectors from a background data set. The GLDS kernel can then be rewritten using the approximated vectors $\overline{\boldsymbol{\Psi}}(\mathbf{X})$ as

$$k(\mathbf{X},\mathbf{Y}) = \overline{\boldsymbol{\Psi}}(\mathbf{X})\mathbf{K}^{-2}\overline{\boldsymbol{\Psi}}(\mathbf{Y}) \tag{3.22}$$

where average over the whole sequence is taken as in Equation 3.20 as

$$\overline{\boldsymbol{\Psi}}(\mathbf{X}) = \begin{pmatrix}\overline{\boldsymbol{\Psi}}_1(\mathbf{X})\\ \vdots \\ \overline{\boldsymbol{\Psi}}_B(\mathbf{X})\end{pmatrix} = \frac{1}{T}\sum_{t=1}^{T}\begin{pmatrix}\boldsymbol{\Psi}_1(\mathbf{X})\\ \vdots \\ \boldsymbol{\Psi}_B(\mathbf{X}))\end{pmatrix} = \frac{1}{T}\sum_{t=1}^{T}\begin{pmatrix}k(\mathbf{x}_t,\mathbf{b}_1)\\ \vdots \\ k(\mathbf{x}_t,\mathbf{b}_B)\end{pmatrix} \tag{3.23}$$

and the kernel matrix $\mathbf{K}$

$$\mathbf{K}_{ij} = k(\mathbf{b}_i,\mathbf{b}_j) \tag{3.24}$$

is computed from the background data. Therefore, vectors $\overline{\boldsymbol{\Psi}}_B(\mathbf{X})$ are computed in the kernel space and not explicitly. Equation 3.22 is derived by means of Singular Value Decomposition (SVD) of the second moment matrix $\mathbf{R}$ computed using background data. In practice, B, the number of prototype vectors in the codebook, can be very large, implying that inversion of $\mathbf{K}$ may not be feasible. The incomplete Cholesky decomposition has been proposed [Louradour *et al.*, 2006] as an alternative factorization of $\mathbf{R}$ to overcome this issue.

3.2.2.2 Score-space Kernels

Generative models classify feature vector sequences based on frame-level log-likelihood or log-likelihood-ratio scores. A sequence-level score is readily obtained as the average score over all frames in the sequence. Score-space kernels project frame-level scores onto a score space of fixed dimensionaly where sequences are actually classified. These approaches involve both generative and discriminant modeling, namely a GMM-UBM system to compute scores and a SVM classifier to classify the derived score sequences.

First developed in [Jaakkola & Haussler, 1999], the Fisher operator performs sequence mapping based on the sensitivity of the log-likelihood scores of the feature vector sequence $\mathbf{X}$ with respect to the model parameters $\boldsymbol{\Theta}$ as

$$\boldsymbol{\Psi}(\mathbf{X}) = \nabla_{\boldsymbol{\Theta}} \log P(\mathbf{X}|\boldsymbol{\Theta}) \tag{3.25}$$

Otherwise stated, the mapped vector $\Psi(\mathbf{X})$ is the direction that maximizes $\log P(\mathbf{X}|\boldsymbol{\Theta})$ at the point given by the observed feature vector sequence $\mathbf{X}$. Based on this mapping, the Fisher kernel is defined as the normalized dot product of the involved vectors as

$$k(\mathbf{X}, \mathbf{Y}) = \boldsymbol{\Psi}(\mathbf{X})^T \mathbf{I}^{-1} \boldsymbol{\Psi}(\mathbf{Y}) \tag{3.26}$$

where $\mathbf{I}$ is the so-called Fisher information matrix

$$\mathbf{I} = \mathrm{E}_{\mathbf{Z}}[\boldsymbol{\Psi}(\mathbf{Z})\boldsymbol{\Psi}(\mathbf{Z})^T] \tag{3.27}$$

that is, the correlation matrix of the Fisher score vectors. An analytic expression for $\boldsymbol{\Psi}(\mathbf{X})$ can be derived for a particular model in terms of its parameters. For GMM using log-likelihood scores, the gradient with respect the mean parameters of Gaussian i becomes

$$\boldsymbol{\Psi}_{\boldsymbol{\mu}_i}(\mathbf{X}) = \nabla_{\boldsymbol{\mu}_i} \log P(\mathbf{X}|\boldsymbol{\Theta}) = \sum_{t=1}^{T} p(i|\mathbf{x}_t) \boldsymbol{\Sigma}_i^{-1} (\mathbf{x}_t - \boldsymbol{\mu_i}) \tag{3.28}$$

An overall vector is then obtained by concatenation of $\Psi_{\boldsymbol{\mu}_i}(\mathbf{X})$ for each Gaussian as

$$\boldsymbol{\Psi}_{\boldsymbol{\mu}}(\mathbf{X}) = (\boldsymbol{\Psi}_{\boldsymbol{\mu}_1}(\mathbf{X})^T, \ldots, \boldsymbol{\Psi}_{\boldsymbol{\mu}_M}(\mathbf{X})^T)^T \tag{3.29}$$

for a model with M Gaussians. Similar developments can be obtained for GMM-UBM systems using log-likelihood-ratio scores by considering the means of of the speaker and background models. Zeroth and higher order derivatives can be also considered but, given the dimensionality of the models used in practice, $\boldsymbol{\Psi}_{\boldsymbol{\mu}}(\mathbf{X})$ already has a very high-dimesionality using first derivatives[3]. For instance, concatenating derivatives up to second order into a single vector is proposed in [Wan, 2003].

The Fisher operator was extended to alternative score spaces in [Smith & Gales, 2002], allowing for score functions f other than a logarithm and any score operator F as

$$\boldsymbol{\Psi}_F^f(\mathbf{X}) = \Psi_F f(P_k(\mathbf{X}|\boldsymbol{\Theta}_k)) \tag{3.30}$$

[3] Assuming M Gaussians with D dimensions per Gaussian, first derivative vectors have MD dimensions and second order derivative vectors would involve $(MD)^2$ dimensions.

which uses a set of generative models with parameters $\mathbf{\Theta}_k$ for model k.

3.2.3 Parameter-space Kernels

In section 3.2.2 we discussed score-space kernels that operate on variable-length sequences of scores provided by a model. An alternative way to use generative models together with SVM is to classify their parameters directly. Speaker features are readily obtained by arranging model parameters into vector form. In this section we discuss two types of features introduced in recent years, GMM supervectors and MLLR coefficients, that result in state-of-the-art ASV systems.

3.2.3.1 GMM Supervectors

GMM are used to estimate the distribution of spectral envelope features of a particular speaker. In the case of GMM-UBM and score-space kernel approaches, this distribution is used to score speech segments from another speaker to estimate their degree of matching. However, every mean vector in the GMM model can be roughly seen as a prototype vector contributing to the modeled distribution. In fact, the feature distribution can be alternatively seen as resulting from the application of a Gaussian kernel on these prototype vectors. The GMM supervector approach stacks the parameters of the model together to form a high-dimensional supervector that gathers speaker-relevant information. These supervectors are then classified using SVM.

Given that supervector coefficients represent a Gaussian mixture distribution, it makes sense to use a distance measure that is natural to statistical distributions. For this purpose, the Kullback-Leibler (KL) divergence

$$D_{KL}(f)||g) = \int_{-\infty}^{\infty} f(\mathbf{x}) \log \frac{f(\mathbf{x})}{g(\mathbf{x})} d\mathbf{x} \tag{3.31}$$

is often used to quantify the dissimilarity between two arbitrary distributions $f(\mathbf{x})$ and $g(\mathbf{x})$. Equation 3.31 can be alternatively seen as the difference in the entropy using an optimal code for $f(\mathbf{x})$ versus using an optimal code for $g(\mathbf{x})$. This measure can not be directly used as a kernel given that it does not satisfy the Mercer conditions, namely the symmetry property[4]. A common way to proceed is by upper bounding the D_{KL} of two Gaussian mixture model distributions $f_{\mathbf{X}}$ and $f_{\mathbf{Y}}$ [Do, 2003] trained using speaker segments $\mathbf{X}$ and $\mathbf{Y}$ as

$$\begin{aligned} D_{KL}(f_{\mathbf{X}}||f_{\mathbf{Y}}) &\leq \sum_{i=1}^{M} \lambda_i D_{KL}(\mathcal{N}(;\boldsymbol{\mu}_i^{\mathbf{X}}, \boldsymbol{\Sigma}_{\mathrm{i}})||\mathcal{N}(;\boldsymbol{\mu}_i^{\mathbf{Y}}, \boldsymbol{\Sigma}_{\mathrm{i}})) && (3.32) \\ &\leq \sum_{i=1}^{M} \lambda_i (\boldsymbol{\mu}_i^{\mathbf{X}} - \boldsymbol{\mu}_i^{\mathbf{Y}})^T \boldsymbol{\Sigma}_{\mathrm{i}}^{-1} (\boldsymbol{\mu}_i^{\mathbf{X}} - \boldsymbol{\mu}_i^{\mathbf{Y}}) && (3.33) \end{aligned}$$

where we have assumed both models are obtained by mean adaptation of the same UBM. Equation 3.32 can be seen as a weighted sum of the KL divergence of two Gaussian

[4] $D_{KL}(f||g) \neq D_{KL}(g||f)$.

distributions or, alternatively, as the distance of the mean supervectors of the two models in the space normalized by Σ. A kernel can be derived from this distance as

$$k(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{M} \lambda_i \boldsymbol{\mu}_{\boldsymbol{i}}^{\mathbf{X}^T} \boldsymbol{\Sigma}_{\boldsymbol{i}}^{-\mathbf{1}} \boldsymbol{\mu}_{\boldsymbol{i}}^{\mathbf{Y}} = \sum_{i=1}^{M} (\sqrt{\lambda_i} \boldsymbol{\Sigma}_{\boldsymbol{i}}^{-\frac{\mathbf{1}}{\mathbf{2}}} \boldsymbol{\mu}_{\boldsymbol{i}}^{\mathbf{X}})^T (\sqrt{\lambda_i} \boldsymbol{\Sigma}_{\boldsymbol{i}}^{-\frac{\mathbf{1}}{\mathbf{2}}} \boldsymbol{\mu}_{\boldsymbol{i}}^{\mathbf{Y}}) \tag{3.34}$$

which satisfies the Mercer conditions. As shown on the right hand part of Equation 3.34, this kernel results in normalized GMM mean supervectors and, therefore, it can be applied before SVM classification.

Systems based on this approach result in high-performing systems which can easily integrate inter-session compensation techniques such as Eigenchannel and Factor Analysis approaches into the SVM framework. Furthermore, leaving inter-session variability compensation apart, they are relatively easy and fast to implement.

3.2.3.2 MLLR Coefficients

So far we have seen that ASV systems use adaptation techniques extensively to obtain speaker-adapted models, eventually compensated for inter-session variability. In the adaptation process a speaker-independent model is turned into a speaker-dependent model. For parametric adaptation, the adaptation parameters capture the difference between generic and speaker models, hence speaker specificities. In recent years, MLLR transform coefficients [Stolcke *et al.*, 2005] have been proposed as speaker-relevant features in ASV systems resulting in excellent performance.

Regression coefficients depend on the adaptation data and the acoustic models used. As opposed to GMM mean supervectors for which a Gaussian prior distribution can be assumed, no prior information is available for MLLR coefficients. In practice, since speaker models are assumed to be close to the speaker-independent model, regression matrices have a rather diagonal look. Further structure in the feature vectors, e.g. Δ and $\Delta\Delta$ features is also visible in the regression matrix, although not relevant for speaker discrimination. Indeed, regression coefficients convey any information between non-adapted and adapted models in correlation terms only.

MLLR coefficients are arranged in supervector form, normalized and classified using SVM given the high feature dimensionality, hence MLLR-SVM. Regarding feature normalization, dynamic range scaling as well as rank normalization has been used in [Stolcke *et al.*, 2005; Stolcke *et al.*, 2006], the latter reporting improved performance.

Former MLLR-SVM approaches used the acoustic models of a LVCSR system to compute MLLR transforms. Using phonemic HMM eases estimation of several transforms corresponding to different acoustic space splits. Based on the transcription for the segment of interest, a forced-alignment is first performed to map each observation vector to a regression class. MLLR transforms are then estimated for each class and the corresponding MLLR transform vectors are concatenated to form a supervector of MLLR coefficients involving the whole acoustic space. A wide range of adaptation schemes are used in LVCSR differing on the way regression classes are found and used, e.g. CMLLR followed by MLLR or gender-dependent transforms. All of them can be used as speaker features.

MLLR approaches using GMM models are typically less performing than LVCSR's due to less precise modeling. Speech dynamics as well as phonetic constraints are not modeled in a GMM-UBM. As a consequence, splitting the acoustic space into several regression classes is not straightforward in linguistic terms and data-driven approaches must be addressed.

Given the MLLR regression matrix $\mathbf{A}$ and offset vector $\mathbf{b}$, one possible arrangement in vector form is

$$\mathbf{m} = [\mathbf{A}_{11}, \dots, \mathbf{A}_{1D}, \dots, \mathbf{A}_{D1}, \dots, \mathbf{A}_{DD}, \mathbf{b}_1, \dots, \mathbf{b_N}]^T \tag{3.35}$$

resulting in $(D^2 + D)$-dimensional vectors if $\mathbf{b}$ is included. These vectors are typically classified using a linear kernel as

$$k(\mathbf{X}, \mathbf{Y}) = \mathbf{m_X}^T \mathbf{m_Y} \tag{3.36}$$

where $\mathbf{m_X}$ and $\mathbf{m_Y}$ are MLLR supervectors for segments $\mathbf{X}$ and $\mathbf{Y}$.

For MLLR adaptation of GMM, the KL divergence based kernel used for GMM mean supervectors (see Section 3.2.3.1) can be rewritten in terms of MLLR regression coefficients and adapted parameters [Karam & Campbell, 2007] as

$$k(\mathbf{X}, \mathbf{Y}) = \mathbf{m_X}^T \mathbf{Q} \mathbf{m_Y} \tag{3.37}$$

where $\mathbf{Q}$ is a positive-definite block-diagonal matrix with D blocks of the form

$$\mathbf{B_k} = \begin{pmatrix} \mathbf{R}_k & \mathbf{r_k} \\ \mathbf{r_k} & \delta_\mathbf{k} \end{pmatrix} \tag{3.38}$$

with $\mathbf{B_k} \in \mathcal{R}^{(D+1)(D+1)}$, $\mathbf{R_k} = \sum_{i=1}^{M} \sqrt{\lambda_i} \boldsymbol{\Sigma_i}^{-\frac{1}{2}} \boldsymbol{\mu_i} \boldsymbol{\mu_i}^T$, $\mathbf{r_k} = \sum_{i=1}^{M} \sqrt{\lambda_i} \boldsymbol{\Sigma_i}^{-\frac{1}{2}} \boldsymbol{\mu_i}$ and $\delta_k = \sum_{i=1}^{M} \sqrt{\lambda_i} \boldsymbol{\Sigma_i}^{\frac{1}{2}}$, following the GMM notation used in previous sections. This kernel was extended to multiple transforms in [Karam & Campbell, 2008].

3.3 Inter-session Variability Compensation in SVM-based Systems

Inter-session variability (ISV) compensation can be carried out in the space of Φ-expanded vectors, be it explicitly or implicitly. Channel distortion, to take an example, has a significant effect on cepstral features, in the form of additive noise. After expansion of the cepstra sequence by an arbitrary mapping function $\Phi(\mathbf{X})$[5], inter-session variability components suffer a same transformation. However, SVM classifiers still have to deal with the noise associated to this undesired variability.

Current approaches to ISV variability compensation for SVM-based systems lean on the flexibility of transformations applied in high-dimensional spaces. They have in common the decomposition of the expanded space into ISV and complementary subspaces. ISV-compensated vectors are obtained by projection of non-compensated vectors onto the ISV-compensated subspace. We discuss Nuisance Attribute Projection (NAP) as well as a few of its variants in the following sections.

[5] Here, we use $\Phi(\mathbf{X})$ to denote a strictly arbitrary mapping, including sequence kernels but also more complex forms of mapping such as GMM supervectors and MLLR transform coefficients.

3.3.1 Nuisance Attribute Projection

Nuisance Attribute Projection (NAP) is founded on the fact that ISV is observable in the first Kernel PCA (KPCA)dimensions of the feature space of an SVM classifier. The NAP approach consists of estimating these directions and projecting them out from the feature space. For this purpose, a database with many speakers and more than one session per speaker is used as training data.

Given two expanded vectors $\Phi(\mathbf{x_i})$ and $\Phi(\mathbf{x_i})$ belonging to classes $\mathcal{C}_1$ and $\mathcal{C}_2$, e.g. two different channel conditions or sessions, we seek a projection operator $\mathbf{P}$ such that the overall inter-class distance over pairs of vectors is minimized after projection, that is

$$\hat{\mathbf{P}} = \underset{\mathbf{P}}{\operatorname{argmin}} \sum_{i \in \mathcal{C}_1} \sum_{j \in \mathcal{C}_2} ||\mathbf{P}(\Phi(\mathbf{x_i}) - \Phi(\mathbf{x_j}))||^2 \tag{3.39}$$

where $\mathbf{P}$ is a projection matrix[6] as

$$\mathbf{P} = \mathbf{I} - \sum_{k=1}^{K} \mathbf{e_k}\mathbf{e_k}^T = \mathbf{I} - \mathbf{E}\mathbf{E}^T \tag{3.40}$$

where $\mathbf{E} = (\mathbf{e_1}, \ldots, \mathbf{e}_K)$ and directions $\mathbf{e_k}$ are removed from the feature space. Based on Equation 3.39 these directions, so-called nuisance vectors, define the space exhibiting maximal inter-class variability. According to the approach in [Solomonoff *et al.*, 2004] a solution to Equation 3.39 can be found by solving the eigenvalue problem

$$\mathbf{A}(\text{diag}(\mathbf{W1}) - \mathbf{W})\mathbf{A}^T\mathbf{v_k} = \lambda\mathbf{v_k} \tag{3.41}$$

where eigenvectors $\mathbf{v_k}$ are normalized so that $\mathbf{v_k}^T\mathbf{K}\mathbf{v_k} = 1$ and converted back to the feature space by means of the linear transform $\mathbf{e_k} = \mathbf{A}\mathbf{v_k}$. $\mathbf{A}$ is the matrix containing N Φ-expanded training data vectors

$$\mathbf{A} = (\mathbf{\Phi}(\mathbf{x_0}), \ldots, \mathbf{\Phi}(\mathbf{x_N})) \tag{3.42}$$

and $\mathbf{W}$ is a weight matrix with 1 for pairs of samples i, j that we want to spread apart and 0 otherwise. Then, if vector $\mathbf{x}$ belongs to class $\mathcal{C}(\mathbf{x})$

$$\mathbf{W_{ij}} = \begin{cases} 1 & \text{if } \mathcal{C}(\mathbf{x_i}) = \mathcal{C}(\mathbf{x_j}) \\ 0 & \text{otherwise} \end{cases} \tag{3.43}$$

Note that the same framework can be used to improve inter-speaker variability by spreading pairs of samples from different speakers, which would lead to an alternative choice of matrix $\mathbf{W}$. In fact, an hybrid matrix including both behaviors is also proposed in [Solomonoff *et al.*, 2004]. Other variants for $\mathbf{W}$ are presented in [Solomonoff *et al.*, 2005]. Once the session subspace spanned by the columns of $\mathbf{E}$ has been estimated, session projections are removed from each vector $\mathbf{\Phi}(\mathbf{x})$ using the projection operator of Equation 3.40 as

[6] Projection matrices are required to be square and idempotent, i.e. $\mathbf{P}^2 = \mathbf{P}$, such that projecting again projected vectors has no effect.

$$\hat{\mathbf{\Phi}}(\mathbf{x}) = (\mathbf{I} - \mathbf{E}\mathbf{E}^T)\mathbf{\Phi}(\mathbf{x}) = \mathbf{\Phi}(\mathbf{x}) - \mathbf{E}(\mathbf{E}^T\mathbf{\Phi}(\mathbf{x})) \tag{3.44}$$

where $\hat{\mathbf{\Phi}}(\mathbf{x})$ is the session-compensated vector. The right hand side of Equation 3.44 avoids explicit computation of matrix $\mathbf{E}\mathbf{E}^T$ which can be expensive in space and time resources.

A simplified version of NAP assuming only one speaker with multiple sessions per speaker is presented in [Campbell *et al.*, 2006]. A modified weight matrix $\mathbf{W}$ can be obtained such that the eigenvalue problem in Equation 3.41 becomes the symmetric problem

$$\mathbf{AJA}\mathbf{v_k} = (\mathbf{AJ})(\mathbf{AJ})^T\mathbf{v_k} = \lambda\mathbf{v_k} \tag{3.45}$$

where $\mathbf{J} = \mathbf{I} - (1/n)\mathbf{1}\mathbf{1}^T$ and $\mathbf{1} = (1, \ldots, 1)^T$. This reduces $(\mathbf{AJ})(\mathbf{AJ})^T$ to the inter-session kernel matrix, obtained by substracting the mean vector across all sessions from each session vector in $\mathbf{A}$. For the multiple speaker case, an averaged kernel matrix for all speakers is used. Therefore, for N_s speakers

$$\mathbf{K} = \frac{1}{N_s}\sum_{i=1}^{N_s}\sum_{j\in S_i}(\mathbf{\Phi}(\mathbf{x_j}) - \overline{\mathbf{\Phi}}(\mathbf{x_j}))^T(\mathbf{\Phi}(\mathbf{x_j}) - \overline{\mathbf{\Phi}}(\mathbf{x_j})) \tag{3.46}$$

This is the most widely used variant of NAP. It has the advantage of reducing labeling effort in the database, since only speakers must be labeled, an obvious requirement in speaker recognition tasks. Furthermore, optimization is performed using standard PCA. In the particular case where this variant is used for ISV compensation of GMM mean supervectors, it has been shown in [Campbell *et al.*, 2006] that NAP is equivalent to the Eigenchannel approach.

A similar approach called Within-Class Covariance Normalization (WCCN) [Hatch *et al.*, 2006] projects features on the subspace with maximal inter-session variability and its complement as well. The projected features are then weighted separately for each subspace, concatenated and classified. A comparative study of WCCN and NAP methods [Kajarekar & Stolcke, 2007] shows similar performance of both techniques across different train and test corpora.

3.3.2 Discriminant NAP

We have just discussed that a simplified variant of NAP estimates nuisance vectors using PCA analysis of the inter-session kernel matrix. Kernel matrices are dual forms of covariance matrices, much like the direct and dual formulation of the SVM optimization problem. By applying PCA on these matrices proportional eigenvalues are obtained. Eigenvectors keep a tight relation as well, since they can be converted back and forth by linear transformation using the data matrix $\mathbf{A}$ or $\mathbf{A}^T$ respectively. In fact, the same solution to NAP can be obtained by applying PCA to the inter-session covariance matrix.

Discriminant NAP (DNAP) extends the standard approach to account for inter-speaker variability as well. Much like in Linear Discriminant Analysis (LDA), we seek those nuisance directions maximizing the ratio of intra-speaker (inter-session) to inter-speaker covariance matrices, i.e. spreading speakers apart while shrinking variability within speakers. Nonetheless, computing such a ratio requires inversion[7] of one of the inter-speaker covariance matrix which, given the high-dimensionality of the feature space, is not possible due to structural singularity problems[8]. The approach in [Vogt *et al.*, 2008] overcomes the singularity problem by using Scatter Difference Analysis (SDA) instead of LDA. A weighted difference of the inter-session $\mathbf{C_h}$ to the inter-speaker $\mathbf{C_s}$ covariance matrices is used, leading to the symmetric eigenvalue problem

$$(\mathbf{C_h} - m\mathbf{C_s})\mathbf{v} = \lambda\mathbf{v} \tag{3.47}$$

which is solved using standard PCA. Such an approach avoids non-orthogonality of the resulting eigenvectors in LDA, which would force orthogonalization of projection matrices[9]. The parameter m is set a priori, e.g. tuned by cross-validation tests. Although conceptually intuitive, this variant has not been shown to improve system performance significantly [Vogt *et al.*, 2008] over standard NAP.

3.3.3 Kernel NAP

The preceding variants of the NAP technique have been applied on explicitly Φ-expanded vectors. In principle, if we want to use high-dimensional (or infinite) expansion functions Φ, the kernel matrix would need to be factorized in the expanded space, which is not feasible. Introducing $\mathbf{Z} = (\text{diag}(\mathbf{W1}) - \mathbf{W})$ for convenience, the NAP eigenvalue problem of Equation 3.41 can be written as

$$\mathbf{A}\mathbf{Z}^{\frac{1}{2}}\mathbf{Z}^{\frac{1}{2}}\mathbf{A}^T\mathbf{v} = \lambda\mathbf{v} \tag{3.48}$$

after factorization of $\mathbf{Z}$, for which the eigenvectors $\mathbf{v}$ lie in the span of the columns of $\mathbf{A}\mathbf{Z}^{\frac{1}{2}}$. Replacing $\mathbf{v} = \mathbf{A}\mathbf{Z}^{\frac{1}{2}}\mathbf{u}$ in Equation 3.48 and leads to

$$\mathbf{Z}^{\frac{1}{2}}\mathbf{A}^T\mathbf{A}\mathbf{Z}^{\frac{1}{2}}\mathbf{u} = \lambda\mathbf{u} \tag{3.49}$$

where the kernel matrix $\mathbf{K} = \mathbf{A}^T\mathbf{A}$ is directly identified, thus avoiding its factorization in the Φ-expanded feature space.

This approach allows projection of non-nuisance directions implicitly using the kernel trick. The compensation kernel

$$\begin{aligned} k_{NAP}(\mathbf{x_i}, \mathbf{x_j}) &= k(\mathbf{x_i}, \mathbf{x_j}) - \mathbf{\Phi}(\mathbf{x_i})^T\mathbf{v}\mathbf{v}^T\mathbf{\Phi}(\mathbf{x_j}) \qquad (3.50)\\ &= k(\mathbf{x_i}, \mathbf{x_j}) - \tilde{\mathbf{v}}_\mathbf{i}\mathbf{Z}^{\frac{1}{2}}\mathbf{u}\mathbf{u}^T\mathbf{Z}^{\frac{1}{2}}\tilde{\mathbf{v}}_\mathbf{j} \qquad (3.51) \end{aligned}$$

where $\tilde{\mathbf{v}}_\mathbf{i}$ can be computed using the kernel trick as

[7] Other alternatives to matrix inversion in LDA are using pseudo-inverses, at the expense of not using the null space of the denominator matrix in the estimation which brings most of the discrimination power, and the Generalized Singular Value Decomposition (GSVD), at the expense of overfitting. Regularization of the covariance matrix by adding a constant to its diagonal is often considered as well.

[8] The rank of the inter-speaker covariance matrix is upbounded by the number of speakers considered.

[9] The projection operator for a non-orthogonal linearly-independent set of vectors $\mathbf{E} = (\mathbf{e_1}, \ldots, \mathbf{e_K})$, is $\mathbf{P} = \mathbf{I} - \mathbf{E}(\mathbf{E}^T\mathbf{E})^{-1}\mathbf{E}^T$

$$\tilde{\mathbf{v_i}} = \boldsymbol{\Phi}(\mathbf{x_i})^T\mathbf{A} = (k(\mathbf{x_i}, \mathbf{x_1}), \ldots, k(\mathbf{x_i}, \mathbf{x_N})) \tag{3.52}$$

A similar development can be obtained taking all nuisance-related eigenvectors in matrix form as $\mathbf{V} = \mathbf{A}\mathbf{Z}^{\frac{1}{2}}\mathbf{U}$. This approach has been recently introduced in [Zhao *et al.*, 2008] obtaining slight improvements over standard NAP.

3.4 Summary and Conclusion

This chapter has discussed the use of SVM classifiers and adapted features for the speaker verification task. Soft-margin SVM use an optimization criterion that balances training classification and generalization performances. This enables working on high dimensional spaces where other classifiers and model overfit. Using SVM for the ASV task requires mapping speech data onto a moderate amount of vectors of fixed dimensionality. For this purpose, sequence kernels and parameter-space kernels stand out, the former performing explicit mapping to obtain features and the latter directly using model parameters as features. Session compensation can also be successfully addressed for these features using a several algorithms.

Part II

Contributions to the MLLR-SVM Approach

Chapter 4

Baseline Speaker Verification Systems

Current ASV systems tend to exploit several types of features and models for improved reliability. We have developed several acoustic systems based on GMM and SVM modeling that are used as baseline systems in later chapters. combined via a logistic regression model of score fusion. All SVM-based systems were developed by the author as part of his thesis work based on a variety of available libraries.

Section 4.1 details PLP-GMM, PLP-SVM and GSV-SVM baseline systems as well as a front-end setup systematically used in most systems evaluated in this thesis. Section 4.3 evaluates all three systems, individually and combined, on the NIST SRE 2005 and 2006 corpora. Section 4.4 gives a brief summary and conclusions.

4.1 System Description

Acoustic ASV systems at the LIMSI laboratory are based on several state-of-the-art techniques discussed in Chapters 2 and 3, mostly based on GMM and SVM approaches to modeling and hybrid generative-discriminative approaches to feature extraction and intersession compensation. Our systems are identified by the type of features (PLP, GSV) dash the type of model they use (GMM, SVM). Accounting for this, the three systems presented in this section are:

- **PLP-GMM**, using PLP features and modeled using GMM, based on the GMM-UBM paradigm.
- **PLP-SVM**, using PLP features and SVM classification, based on the GLDS kernel.
- **GSV-SVM**, using Gaussian mean supervectors based on PLP features and SVM classification.

All of these systems use the same PLP front-end, from which any further processing is based on. We describe it in the next section.

4.1.1 Front-End Setup

The front-end setup used in our systems was set while participating in past NIST SRE campaigns. Although optimized for the PLP-GMM system, we kept it the same across systems to ease system comparison and internal maintenance.

Assuming we are dealing with telephone speech sampled at a rate of 8kHz, we provide front-end specifications in the following list:

- **30 ms** (240 samples) analysis window size, **10ms** (80 samples) shift period
- Only **voiced frames** with a **minimum energy**[1] are considered. Voicing is detected using the ESPS get_f0[2] pitch extraction algorithm. Unvoiced frames are dropped
- **15 MEL-PLP** coefficients with Δ and $\Delta\Delta$ derivatives, and Δ and $\Delta\Delta$ energies, a total of **47** features
- **0-3.8kHz** filterbank bandwidth
- **Feature Mapping** using two gender- and three channel-dependent models (GSM, CDMA and landline handsets) trained on 6 hours/gender (24769 segments) of speech data taken from test speakers segments in NIST SRE 1997 to 2002 evaluation data
- **Feature warping**[3] using a 3 second sliding window

These feature extraction steps are applied from top-to-bottom and left-to-right order in the list. Experimentation confirmed such choice over other ordering possibilities for the PLP-GMM system. Feature mapping and feature warping may not be applied for those systems using inter-session variability compensation at the model or SVM levels. Unless otherwise stated, we assume the above configuration is used from here on.

4.1.2 PLP-GMM System

The PLP-GMM system is based on the GMM-UBM paradigm using hybrid Eigenchannel inter-session variability compensation based on a factor analysis model of utterance variability, as proposed in [Matrouf *et al.*, 2007]. We bypass feature mapping in the front-end.

We use two gender-dependent UBM with 1536 Gaussians trained using about 24 hours of speech data per gender taken from the NIST SRE 2000 training data and SRE 2001 training and development data. We perform 5 iterations of maximum-likelihood training with a 1% variance floor of the global cepstral variance. Speaker models are adapted based on the factor analysis model of Equation 2.31, hence performing model-level inter-session compensation in the training stage. We use 3 iterations and a relevance factor of 10. We used a channel subspace dimension of 40 as well as 20 iterations to train the channel-factor loading matrix **U** using the NIST SRE 2004 training data consisting of 310 speakers and 2972 sessions, that is, an average of about 10 sessions per speaker.

In the test phase we score feature-level-compensated test speaker segments against compensated target speaker models using the standard likelihood ratio formulation. Likelihood scores are obtained based on the 20 top-scoring Gaussians obtained during UBM

[1] The energy threshold was obtained empirically so that residual cross-talk from the opposite telephonic side is filtered out.

[2] KTH Sotfware, `http://www.speech.kth.se/software`.

[3] Note that feature warping is also used for non-GMM-based systems.

scoring. We further normalize the resulting likelihood ratio scores using gender-dependent T-norm on two 250 cohort speaker sets taken from the NIST SRE 2004 training data. This configuration was optimized for the NIST SRE 2008 campaign.

4.1.3 SVM-based Systems

In this thesis, we deal with SVM-based systems other than PLP-SVM and GSV-SVM systems described below. Whenever it is possible, and aiming at fair system comparison, we have chosen to use keep setups as close as possible from system to system[4]. In that sense, SVM-based systems extract base supervectors using whatever method keeping further processing homogeneous for all systems. This includes NAP intersession compensation, min-max normalization of supervector coefficients as well as SVM classification, all of them along with their tuning parameters. We present first the common processing steps and we describe the base supervectors used in PLP-SVM and GSV-SVM systems in the following sections.

In a first stage we apply NAP to compensate for inter-session variability of base supervectors. As discussed in Section 3.3.1, NAP finds a linear transform that removes the subspace exhibiting maximal inter-session variability in the feature space. Such transform is trained using NIST SRE 2004 training data, which is known to involve a large inter-session variability[5]. We use the NAP version maximizing Equation 3.46, i.e. the inter-session covariance matrix pooled across speakers. We assume a single gender-independent transform can be applied to gender-dependent features, as NAP aims at removing intra-speaker variability only, leaving inter-speaker components, e.g. gender, as they are. We set the session subspace dimension to 50 which was experimentally found to be close to optimal for all systems discussed in the thesis.

After session compensation, the resulting supervectors are normalized by means of min-max feature scaling. Every feature in the supervector is fit into the range [-1/$\sqrt{M}$,1/$\sqrt{M}$], where M is the number of features in the supervector. This forces unitary dot products for any pair of supervectors, which is also useful to avoid conditioning issues of Hessian and kernel matrices involved in SVM training. The resulting mean value of the features is expected to be 0, so any offset before normalization is removed. Min-max statistics are collected from the impostor speaker set detailed in the next paragraph. In previous experiments, this method was found to outperform mean and variance normalization as well as rank normalization for several SVM-based acoustic systems.

The impostor speaker data set consists of 2243 speech segments[6] from the NIST SRE 2004 training data plus 4854 speech segments[7] from the Switchboard I (SWB1) corpus, all in English language with a minimum and average effective duration of 10 seconds and 2 minutes[8] respectively. Transcripts are available for all of the segments as well. SRE 2004 transcripts were obtained automatically using a LVCSR system and they were provided by NIST for the SRE 2004 evaluation. SWB1 transcripts were obtained manually. This configuration allows all SVM-based systems to share the same impostor data, as we need transcripts for some systems using MLLR adaptation but we certainly do not for other acoustic systems.

[4] Optimizing each system independently is another valid criterion in the sense that each type of supervector can interact with the post-processing steps in a specific way.

[5] Most of the 310 speakers have more than 10 sessions per speaker

[6] About 60 hours of effective speech, after speech activity detection.

[7] About 170 hours of effective speech.

[8] For homogeneity with train and test data, which have an average duration of 2 minutes as well.

We use a soft-SVM (C-SVM) for classification, trained using gender-dependent impostor speaker data and using a linear kernel. We took the SVMTorch[9] package developed at the IDIAP laboratory. We do not normalize scores as it was found not to be beneficial[10].

4.1.4 PLP-SVM System

The PLP-SVM system is based on the GLDS kernel (see Section 3.2.2.1) using a explicit polynomial mapping and SVM classification. We use two-way channel compensation, i.e. feature mapping used by the PLP15N front-end and NAP at the polynomial feature level.

We obtain one supervector per cepstral vector first, by concatenating first, second and third monomial expansions of cepstra. These are normalized to unity variance within the segment and averaged to form one base supervector per speaker segment, i.e. 20824 features[11]. Such supervectors are then post-processed using NAP, min-max scaling and classified using the setup described in Section 4.1.3.

4.1.5 GSV-SVM System

The GSV-SVM system uses Gaussian mean supervectors of GMM classified using SVM. Channel and session compensation are addressed at the cepstral-level as well as at the GMM supervector level using NAP[12].

We use two gender-dependent UBM with 256 Gaussian components and diagonal covariance matrices trained using the impostor speaker speech data, i.e. 120 hours per gender taken from NIST SRE 2004 and Switchboard I corpora. We performed 5 iterations of maximum-likelihood training with a threshold of 1% of the global variance as variance floor. Speaker GMM are obtained by standard MAP mean adaptation using 3 iterations and a relevance factor of 10. The resulting Gaussian means are normalized as in Equation 3.34 obtaining one supervector per speaker segment. These supervectors are post-processed using NAP, min-max scaling and classified using the setup described in Section 4.1.3.

4.2 Experimental Protocol

Performance of the systems explored in this thesis is evaluated under the NIST SRE protocol. Adhering to such protocol allows direct and meaningful comparison of systems or approaches from different laboratories.

We focus on NIST SRE 2005 and 2006 protocols, excluding year 2008 campaign[13]. Such protocols provide several task conditions to participate in, mainly differing in the amount

[9] SVMTorch, a Support Vector Machine for Large-Scale Regression and Classification Problems `http://www.idiap.ch/learning/SVMTorch.html`

[10] We believe this unusual performance loss could be due to SVM processing in a highly unbalanced training data scenario, as score distributions exhibit a large asymmetry, with most of the scores gathered around -1.

[11] The number of distinct monomials of order p with D features is $\frac{(D+p)!}{D!p!}$. Then, concatenating monomials from order 1 up to order P results in $\sum_{p=1}^{P} \frac{(D+p)!}{D!p!}$ features per supervector, i.e. 20824 for $D = 47$ and $P = 3$.

[12] NAP compensation of GMM supervectors is equivalent to eigenchannel-based compensation using the model of Equation 2.30.

[13] NIST SRE2008 campaign focused on unseen-channel mismatched-condition, cross-language and telephone/interview style trials. Although more challenging than the preceding campaigns, none of these is the focus of the thesis work, leaving aside the large amount of resources required to evaluate systems on such database.

of target and test speech involved. This thesis targets the so-called core condition only, i.e. 1conv4w-1conv4w. This condition provides 5-minute-long telephonic conversations in English language involving two sides with about 2 minutes of effective speech available per side. Only one speaker is present in each conversation side, so there is no need for speaker segmentation, only speech activity detection. Speech data are taken from the LDC Mixer[14] project involving topic-oriented telephone speech obtained on a reward basis.

As for verification, every access trial involves a target speaker and a test segment of the same gender, which is known a priori. The target speaker identity is used to train the speaker model using the associated segment and the test segment is to be scored against the target model. Officially, only the involved test segment can be used for making decisions. The main performance measure used in the evaluation is DCF discussed in Sections 1.4.5 and 1.4.7.

In order to exclude system calibration error for evaluation purposes, we use the DCF obtained for the optimal threshold instead of the actual DCF after decision making, so-called Minimum Detection Cost (MDC). We use the corresponding evaluation data, i.e. SRE 2005 or 2006, to find such an optimal threshold. Hence, scores are biased in the sense that the optimal threshold is found a posteriori, using the same scores we decide on. Still, NIST uses both measures although it ranks systems based on actual DCF. Using MDC provides a performance measure not sensitive to calibration interactions which seems more appropriate to observe gain and loss of performance resulting from structural changes in a system. For SRE 2005 we use the latest keyfile available, i.e. version 7b involving 23118 trials, while we use keyfile version 9a for SRE 2006, involving 22316 trials.

Scoring is performed in two different ways depending on the role of target and test speaker data in the system. In forward scoring we score test speaker speech against the target speaker model, in the usual way, while in backward scoring we score target speaker speech against the test speaker model. Such an approach is aimed at symmetrizing train and test phases, being specially appealing when target and test segments have the same duration, which is the case of NIST SRE in average. Therefore, for individual systems, we provide a forward score per trial as well as the fusion of forward and backward scores using an uniformly-weighted average, i.e. each score weighted with 0.5. Based on this scoring scheme, we can improve system performance using the very same features and modeling paradigm.

As we briefly discussed in Section 1.4.6, score fusion is an effective way to further improve performance by combining different speaker features and modeling paradigms. We also provide system fusion results based on the logistic regression model of Equation 1.26 for different combinations of the baseline systems with the approaches proposed in each chapter. We use the SRE 2005 evaluation data to train the logistic regression model and we provide system fusion results for SRE 2006 only.

Experimental results are identified by the system name given the evaluation year. Tables include three main columns:

- **System Name**, including distinctive parameters if necessary. For long system names, specific naming strategies may be used.
- Forward (column F) and forward-backward-combined (column FB) **MDC** and **EER** in % of error rate for NIST **SRE 2005**
- Forward (column F) and forward-backward-combined (column FB) **MDC** and **EER** in % of error rate for NIST **SRE 2006**

[14] Mixer Telephone Study, `http://mixer.ldc.upenn.edu/`.

	SRE 2005				**SRE 2006**			
	MDC		EER (%)		MDC		EER (%)	
System	F	FB	F	FB	F	FB	F	FB
PLP-GMM	.0287	.0202	5.82	4.74	.0218	.0177	4.69	3.72
PLP-SVM	.0211	.0204	4.82	4.48	.0198	.0189	4.41	4.14
GSV-SVM	**.0177**	**.0172**	**4.66**	**4.45**	**.0182**	**.0174**	**3.54**	**3.26**

Table 4.1 — MDC and EER of individual baseline systems for SRE 2005 and SRE 2006. Columns F and FB show forward and averaged forward-backward scores respectively. The best scores in each column are shown in boldface.

Non-available results are shown as a dash sign.

4.3 Baseline System Results

In this section we provide performance results for the baseline systems described in Section 4.1, i.e. PLP-GMM, PLP-SVM and GSV-SVM, either individually or in combination. Since systems have evolved during the length of this thesis, we present results for the latest implementations evaluated on the SRE 2005 and 2006 data, rather than results of the actual systems submitted to each of the evaluation campaigns. This provides fair system comparison across the manuscript as of 2008. We give individual results in Section 4.3.1 and fused system results in Section 4.3.2.

4.3.1 Individual System Results

Table 4.1 shows MDC and EER measures for PLP-GMM, PLP-SVM and GSV-SVM systems, the latter using 256 Gaussians, evaluated on SRE 2005 and SRE 2006 corpora. A quick look at it reveals lower error rates for the SRE 2006 corpus, which could be explained by slight structural differences between both databases, e.g. percentage of native or bilingual speakers. This is a trend that will be further confirmed as more and more results are introduced in the manuscript. PLP-SVM has the most stable behavior from SRE 2005 to SRE 2006.

GSV-SVM outperforms the two other systems, obtaining a relative improvement over 20% MDC and EER versus PLP-GMM using forward scoring. Smaller gains are found versus PLP-SVM. The difference of performance among the three systems is particularly visible in the constellation plots of Figures 4.5(a) and 4.5(b), where GSV-SVM lies in the bottom-left corner, PLP-GMM in the top-right corner and PLP-SVM in between. DET curves of Figures 4.1(a) and 4.2(a) confirm such behavior across operating points, although GSV-SVM is slightly outperformed by PLP-SVM in the high false-alarm region. Forward-backward combination is very effective for PLP-GMM, obtaining gains over 18% MDC and EER versus forward scoring while these reduce to 2%-8% for the other systems. Figures 4.1(b) and 4.2(b) show DET curves of these systems. PLP-GMM curves have approached those of GSV-SVM and PLP-SVM considerably, to the extent that PLP-GMM outperforms PLP-SVM at all operating points for SRE 2006, and in the high false-alarm rate area for SRE 2005.

As mentioned above, the GSV-SVM system above uses 256-Gaussian GMM as speaker models. This corresponds to the optimal number of Gaussians as confirmed in the results

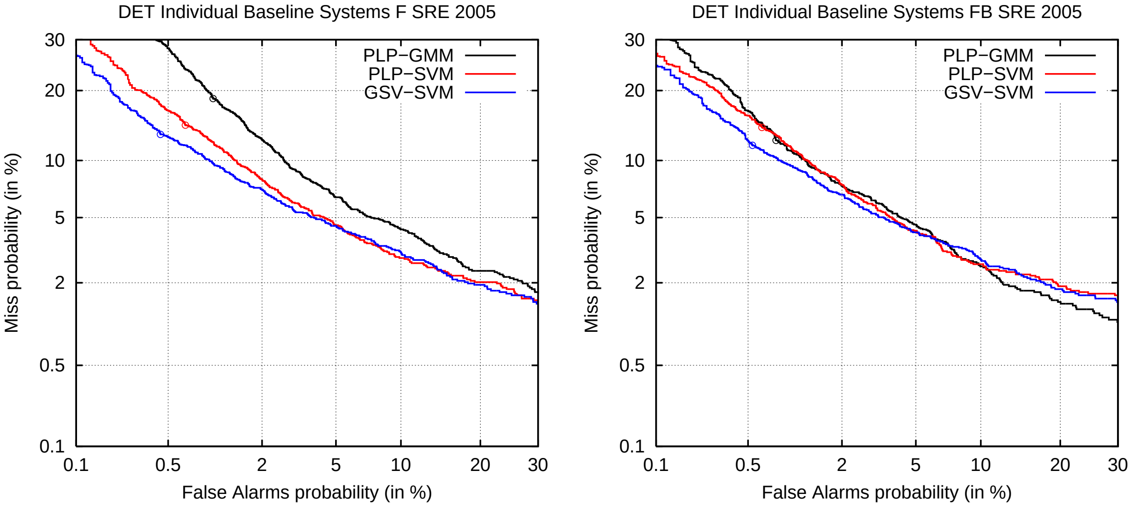

Figure 4.1 — DET curves of PLP-GMM, PLP-SVM and GSV-SVM individual baseline systems using forward scoring (left, a) and forward-backward scoring (right, b) for SRE 2005.

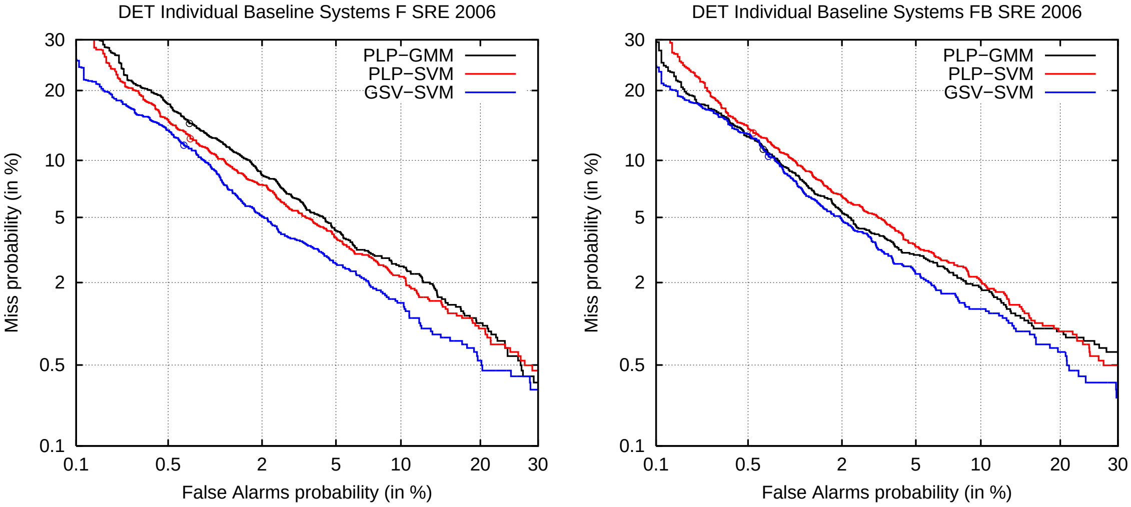

Figure 4.2 — DET curves of PLP-GMM, PLP-SVM and GSV-SVM individual baseline systems using forward scoring (left, a) and forward-backward scoring (right, b) for SRE 2006.

presented below. We assessed GSV-SVM performance for several configurations differing in the number of Gaussians used. Table 4.2 shows MDC and EER error measures using from 64 to 1024 Gaussians in exponential steps.

We observe that MDC and EER decrease down to an optimal performance point as we use more and mode Gaussians. We obtain relative gains ranging from 10% to 20% in MDC or EER with respect to the worst performing system. Using even more Gaussians performance drops again, probably caused by generalization issues of the adapted speaker models. The optimal number of Gaussians is about 256, appearing in the left-bottom cor-

	SRE 2005				SRE 2006			
	MDC		EER (%)		MDC		EER (%)	
System	F	FB	F	FB	F	FB	F	FB
GSV-SVM 64g	.0226	.0214	5.32	5.24	.0207	.0201	4.82	4.59
GSV-SVM 128g	.0192	.0190	5.11	5.03	**.0181**	.0174	4.03	3.91
GSV-SVM 256g	**.0177**	**.0172**	**4.66**	4.45	.0182	**.0174**	**3.54**	**3.26**
GSV-SVM 512g	.0186	.0179	4.91	**4.40**	.0200	.0193	4.09	3.77
GSV-SVM 1024g	.0199	0.187	5.03	4.62	.0218	.0184	4.26	3.67

Table 4.2 — MDC and EER of GSV-SVM systems using from 64 to 1024 Gaussians for SRE 2005 and SRE 2006. Columns F and FB show forward and averaged forward-backward scores respectively. The best scores in each column are shown in boldface.

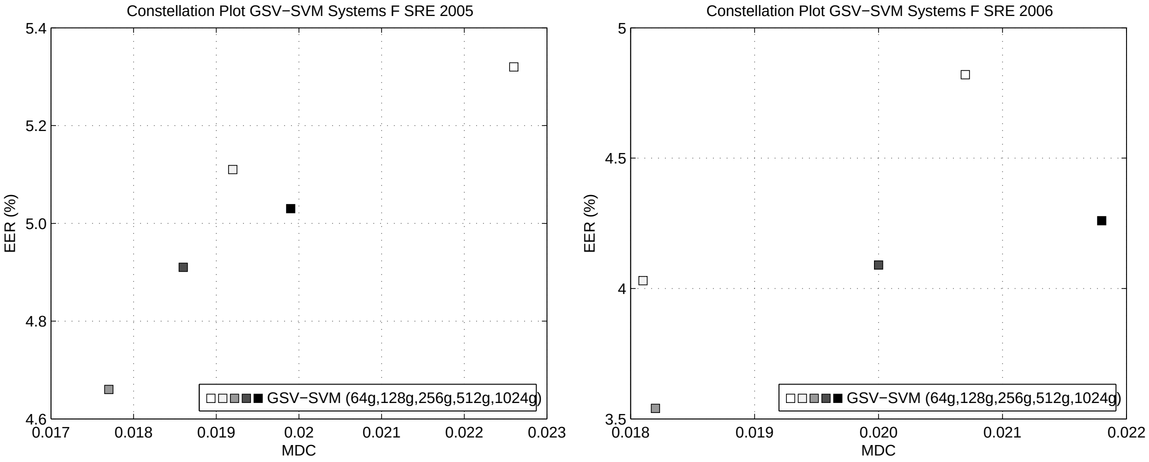

Figure 4.3 — Constellation plots of forward-scoring GSV-SVM systems using a different number of Gaussian components for SRE 2005 (left, a) and SRE 2006 (right, b). The more Gaussians used the darker the symbols.

ner of the constellation plots of Figures 4.3(a) and 4.3(b). Note that, in these plots, the other systems lie fairly far away from GSV-SVM 256g for both SRE 2005 and SRE 2006, with the exception of GSV-SVM 128g obtaining slightly better MDC for SRE 2006. Figures 4.4(a) and 4.4(b) show DET curves of GSV-SVM systems using forward scoring only for both corpora. Optimality of 256 Gaussians depends on the operating point targeted by the system, as 512 Gaussians obtains comparable or better performance at certain operating points for both SRE 2005 and SRE 2006, although not overall. On the other side, 256 Gaussians is slightly lower than the optimal number found in other studies such as [Brummer *et al.*, 2007]. Regarding forward-backward system combination, gains are system-dependent exhibiting a large variability although roughly in the range 3%-15% MDC and 2%-13% EER.

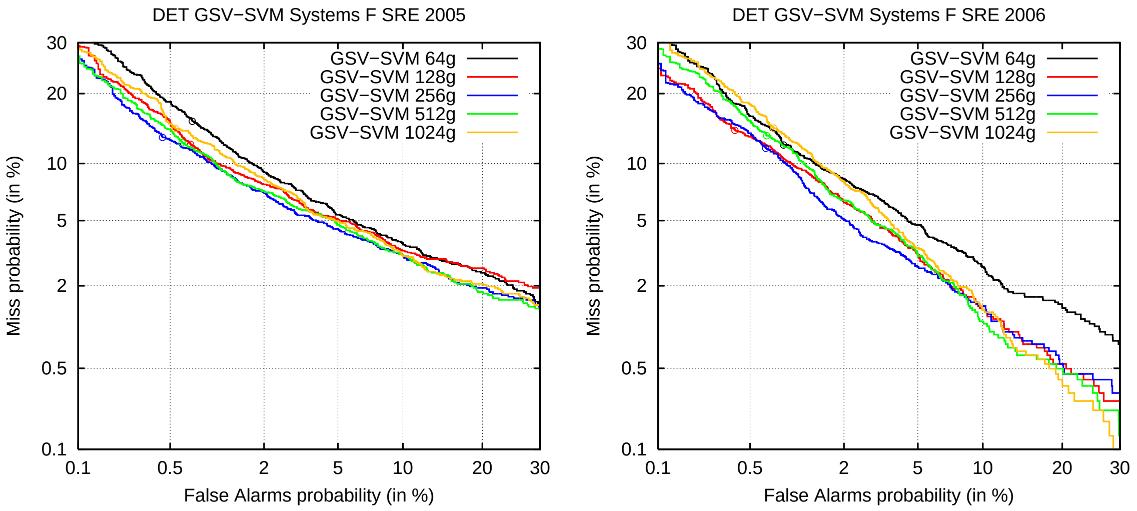

Figure 4.4 — DET curves of GSV-SVM systems using 64, 128, 256, 512 and 1024 Gaussians in the speaker models for SRE 2005 (left, a) and SRE 2006 (right, b).

	SRE 2005				SRE 2006			
	MDC		EER (%)		MDC		EER (%)	
System	F	FB	F	FB	F	FB	F	FB
PLP-GMM (a)	.0287	.0202	5.82	4.74	.0218	.0177	4.69	3.72
PLP-SVM (b)	.0211	.0204	4.82	4.48	.0198	.0189	4.41	4.14
GSV-SVM (c)	**.0177**	**.0172**	**4.66**	**4.45**	**.0182**	**.0174**	**3.54**	**3.26**
(a)+(b)	–	–	–	–	.0169	.0155	3.45	3.35
(b)+(c)	–	–	–	–	.0162	.0157	3.45	3.30
(a)+(c)	–	–	–	–	.0157	.0150	3.32	**3.12**
(a)+(b)+(c)	–	–	–	–	**.0149**	**.0147**	**3.13**	**3.12**

Table 4.3 — MDC and EER of baseline and fused baseline systems for SRE 2005 and SRE 2006 data. Columns F and FB show forward and averaged forward-backward scores respectively, which also applies to system combination. The best scores in each column and block are shown in boldface.

4.3.2 Fused System Results

In this section we provide results for different combinations of individual systems using the logistic regression model of Equation 1.26 for score fusion. We used SRE 2005 data to train such model and applied the obtained parameters to SRE 2006 scores, thus being able to assess system performance on the latter database only. Logistic regression applies a scaling and an offset to each of the system scores which, after application of the logistic function, results in calibrated log-likelihood ratios as output scores as well. We used the FoCal toolkit[15] for this purpose.

[15] FoCal Toolkit, `http://www.dsp.sun.ac.za/~nbrummer/focal/index.htm`

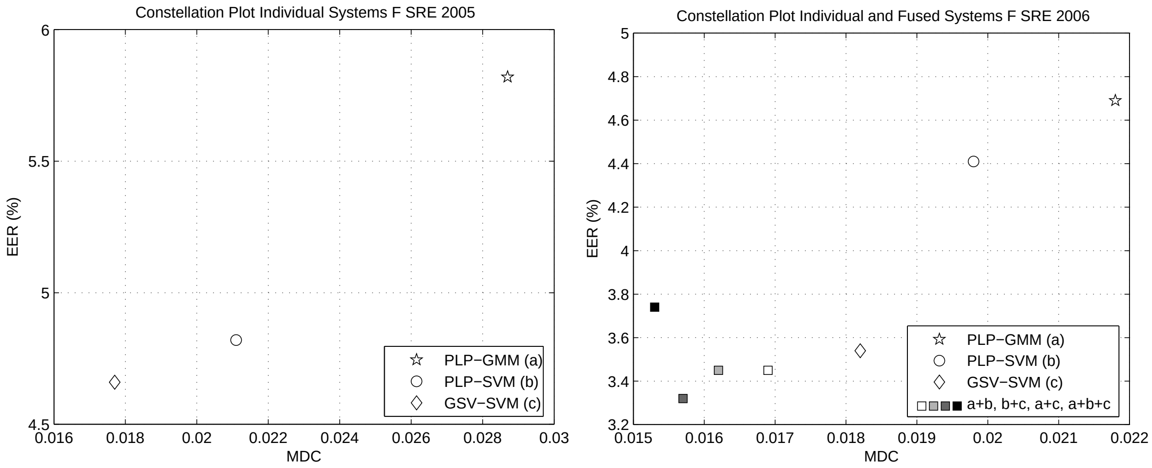

Figure 4.5 — Constellation plots of forward-scoring individual systems for SRE 2005 (left,a) and both individual and fused systems for SRE 2006 (right,b). The more systems included in the fusion the darker the symbols.

Table 4.3 shows MDC and EER measures for PLP-GMM, PLP-SVM and GSV-SVM in the upper block and for all possible two-system and three-system combination in the lower block. We observe that fused systems significantly outperform individual systems in up to 16% relative MDC, although only a slight improvement in EER is eventually found for certain combinations, e.g (a)+(c). These results also show that including more and more systems in the combination does not necessarily result in more performance, e.g three-system versus two-system fusion, suggesting that non-redundant information in the fused scores can be an important factor for successful system fusion. The constellation plots of Figure 4.5(b) shows that fused systems are all clustered in the bottom-left, most performing, area of the plot. MDC improves consistently as symbols become darker, i.e. more systems included in the fusion, while it is not the case for EER.

Forward-backward scoring brings small improvement for fused systems, most of it being provided by fusing heterogeneous features and modeling paradigms. Figures 4.6(a) and 4.6(b) show DET curves of forward- and forward-backward-scored individual and fused systems, with grayed and black curves respectively. DET curves for fused systems are well below PLP-GMM and PLP-SVM curves in both scoring approaches, with differences getting smaller for forward-backward scoring. GSV-SVM curves are rather overlapped with less performing fused system curves. Three-system combination (a)+(b)+(c) is the more performing system combination overall, with (a)+(c) eventually outperforming it in the area near EER, as it is also indicated in Table 4.3. Including PLP-SVM in the fusion results in the smallest improvements.

It may be of interest to define a system combination baseline to fuse the approaches presented in further chapters with. We use (a)+(b)+(c) for this purpose. First, DET curves show it is the most performing system over a wide range of operating points. Second, it includes a high diversity of features and models which confers improved stability of the overall system.

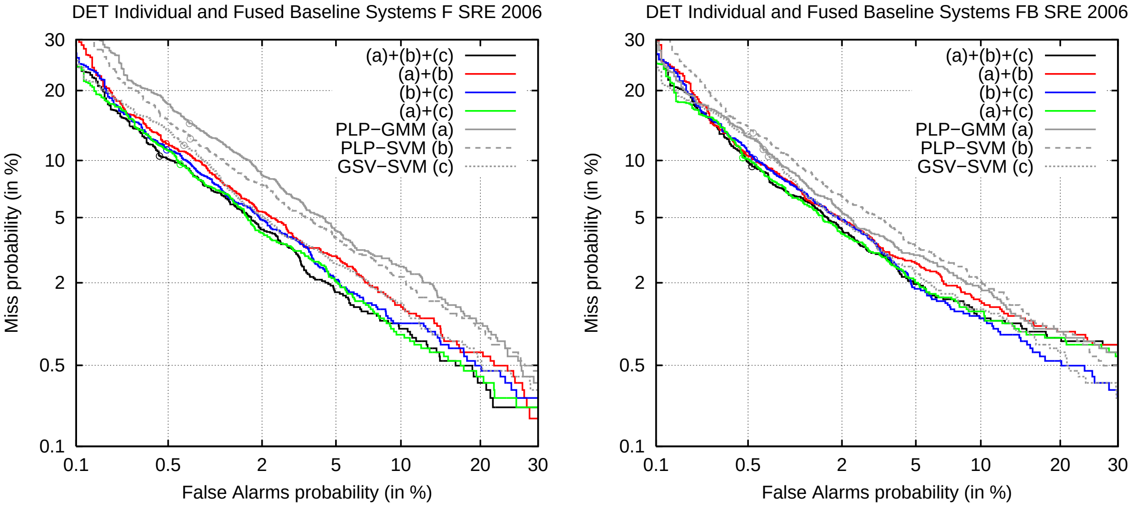

Figure 4.6 — DET curves of fused baseline systems for SRE 2006 using forward scoring (left, a) forward-backward scoring (right, b).

4.4 Summary and Conclusion

This chapter has described and evaluated PLP-GMM, PLP-SVM and GSV-SVM baseline systems. PLP-GMM is based on the GMM-UBM paradigm and uses hybrid factor analysis for inter-session variability compensation. PLP-SVM and GSV-SVM systems use polynomial features based on the GLDS kernel and Gaussian mean supervectors as features respectively, both post-processed using NAP session compensation, dynamic range scaling and SVM classification using a linear kernel. Individual systems obtained relatively similar performance, with GSV-SVM 256g outperforming PLP-GMM and PLP-SVM overall. All fusion schemes outperformed the best individual systems. Optimal performance was found by combining all systems obtaining gains around 16% MDC compared to GSV-SVM.

Chapter 5

Constrained MLLR-SVM Systems

Maximum-Likelihood Linear Regression (MLLR) transform coefficients have been recently proposed as alternative speaker features in ASV systems. Given the parametric nature of these transforms, much of the speaker-relevant information is captured by the regression coefficients resulting from adaptation of a speaker-independent model to speech data. The coefficients of each regression class, rather than the adapted model, are then classified using SVM. Such MLLR-SVM paradigm (see Section 3.2.3.2) has shown performance [Stolcke *et al.*, 2005; Stolcke *et al.*, 2006] comparable to other state-of-the-art ASV systems, especially those using the acoustic models of a LVCSR system.

In this chapter we present an approach to MLLR-SVM based on its Constrained MLLR (CMLLR) variant which uses regression coefficients computed using a Universal Background Model (UBM) in a Speaker Adaptive Training (SAT) framework. The resulting CMLLR-SVM approach aspires to be a simple yet effective alternative to standard MLLR-SVM, given that it avoids the use of transcripts and language constraints as well.

The chapter is organized as follows: An overview of the mean-variance constrained variant of MLLR is given first in Section 5.1. Section 5.2 discusses different ways in which CMLLR can be taken advantage of in speaker verification continuing with speaker adaptive training in Section 5.3. Sections 5.4 and 5.5 present and evaluate the base CMLLR-SVM system without and with NAP inter-session variability compensation. Section 5.6 discusses different representations of CMLLR transforms for SVM classification. The chapter continues with a CMLLR-based algorithm for feature-level inter-session variability compensation in Section 5.7. A brief summary and conclusions are given in Section 5.8.

5.1 Constrained MLLR

A main concern regarding MLLR adaptation is reliable estimation of regression coefficients given the available adaptation data. It is common practice to reduce the amount of parameters in the linear regression model [Gales & Woodland, 1996], e.g. use diagonal or block-diagonal adaptation matrices or share mean and variance transforms. In this section we focus on the latter type of transforms, namely those using a single full matrix transform for mean and covariance matrix adaptation. Constrained MLLR (CMLLR) [Digalakis *et al.*, 1995; Gales, 1998] transforms each Gaussian mean vector and covariance matrix in a regression class as

$$\begin{aligned} \hat{\boldsymbol{\mu}} &= \mathbf{A}\boldsymbol{\mu} + \mathbf{b} \\ \hat{\boldsymbol{\Sigma}} &= \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^T \end{aligned} \tag{5.1}$$

where $\boldsymbol{\mu}$ and $\hat{\boldsymbol{\mu}}$ are non-adapted and adapted Gaussian mean vectors, respectively, and $\boldsymbol{\Sigma}$ and $\hat{\boldsymbol{\Sigma}}$ are non-adapted and adapted covariance matrices. As in MLLR, the transform parameters $\mathbf{A}$ and $\mathbf{b}$ are optimized under the maximum likelihood criterion. Given some adaptation data $\mathbf{X}$

$$\hat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} \log P(\mathbf{X}|\boldsymbol{\Theta}) \tag{5.2}$$

where $\boldsymbol{\Theta} = (\mathbf{A}, \mathbf{b})$ and $\hat{\boldsymbol{\Theta}} = (\hat{\mathbf{A}}, \hat{\mathbf{b}})$ are the non-adapted and adapted model parameters, respectively. Optimization is performed using the EM algorithm by computing sufficient statistics for current estimates of $\mathbf{A}$ and $\mathbf{b}$ in the expectation step and then maximizing with respect to these parameters in the maximization step.

In the case that only one regression class is used, the feature-space affine transform

$$\hat{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b} \tag{5.3}$$

results in the adapted model parameters of Equation 5.1. This implies that CMLLR adaptation can be performed in model or feature spaces indistinctively. In the latter case, the adaptation data is actually transformed to maximize the likelihood of the non-adapted model instead. Adapting in the opposite direction results in the inverse model-space transform. Then, following the notation in Equation 5.3

$$\mathbf{x} = \mathbf{A}^{-\mathbf{1}}\hat{\mathbf{x}} - \mathbf{A}^{-\mathbf{1}}\mathbf{b} \tag{5.4}$$

where $\hat{\mathbf{x}}$ is now a generic feature vector in the adaptation data and $\mathbf{x}$ is the corresponding feature vector adapted to the speaker-independent model. Assuming $\mathbf{A}$ is invertible[1], features and model parameters can be adapted in both directions using parameters $(\mathbf{A}, \mathbf{b})$ and $(\mathbf{A}^{-\mathbf{1}}, -\mathbf{A}^{-\mathbf{1}}\mathbf{b})$.

Therefore, CMLLR presents two main advantages over MLLR. First, considering the same amount of parameters, CMLLR performs covariance matrix adaptation while MLLR does not. Visually, CMLLR shifts and reshapes Gaussians while only the former is performed by MLLR mean adaptation. Second, CMLLR can be used in both feature and model spaces, while MLLR only works in the model space, although only when a single regression class is used.

5.2 CMLLR and Speaker Recognition

As an adaptation method of HMM/GMM, CMLLR can be used in speaker recognition systems in a variety of ways. Its major use is, however, adaptation of a speaker-independent model to speaker-dependent speech data or viceversa, via model- or feature-space transformations respectively. The resulting regression coefficients represent the difference between data and model, thus capturing relevant speaker information. Model- and feature-space

[1] $\mathbf{A}$ should be invertible if the feature covariance matrix $\boldsymbol{\Sigma}$ is invertible too, which is assured if it is positive-definite.

transforms have different regression coefficients but represent the same phenomenon.

Model-space CMLLR adaptation can replace any other adaptation method, e.g. standard MAP or Eigenvoices, with the advantage of adapting both mean vectors and covariance matrices at the same time. As in standard MLLR, CMLLR adaptation can be performed for a set of regression classes as well. The resulting speaker-dependent models can be used in any system, including GMM-UBM, GSV-SVM, and the regression coefficients in MLLR-SVM. One point to account for is that all of the Gaussians in a regression class are adapted regardless of the amount of adaptation data available[2]. Although this has important advantages, e.g. adaptation is performed without the need of observing the whole acoustic space, it could impair likelihood ratios in GMM-UBM systems, as the UBM and the target model become less comparable with respect to using standard MAP adaptation.

The feature-space approach is more appealing than its counterpart for certain purposes, for instance, for improving a speaker-independent model. An UBM is typically trained using data from many speakers, which can be achieved in different ways. One popular approach used in speech recognition is Speaker Adaptive Training (SAT) [Anastasakos *et al.*, 1996], which uses feature-space CMLLR transforms that project speaker-dependent data onto a speaker-independent space. The resulting features are then used to obtain a refined speaker-independent model in a later stage. This approach is described in section 5.3.

In scenarios with multiple acoustic conditions inter-session variability compensation plays an important role. Eigenchannel approaches based on eigen-analysis of CMLLR transform supervectors, which is equivalent to NAP in this case, are also worth investigating. Given that the compensated CMLLR transforms can also be used at the feature level, direct compensation can be addressed at this level as well.

5.3 Speaker Adaptive Training

A typical use of feature-space CMLLR is speaker adaptive training (SAT) [Anastasakos *et al.*, 1996] of a speaker-independent model. This approach seeks to jointly estimate a speaker-independent model and a set of speaker adaptation transforms, one per speaker, that result in such model. Given a set of B speakers and their corresponding adaptation cepstra $\mathbf{X_i}$ for $1 \leq i \leq B$, SAT optimizes the maximum likelihood criterion in a per-speaker basis as

$$\arg\max_{\mathbf{\Theta},\mathcal{C}_i^{-1}} \prod_{i=1}^{B} p(\mathcal{C}_i^{-1}(\mathbf{X_i})|\mathbf{\Theta}) \tag{5.5}$$

where individual speaker-dependent transforms $\mathcal{C}_i$ and the model parameters $\mathbf{\Theta}$ are jointly estimated. In practice, such optimization is commonly done in two steps by, first, estimating feature-space CMLLR transforms $\mathcal{C}_i$ that project speaker-dependent features onto a speaker-independent space and, second, re-training the speaker-independent model $\mathbf{\Theta}$ using those features. This process, illustrated in figure 5.1 for two re-estimation steps, can be iterated several times in an EM manner, obtaining a more speaker-independent model at each iteration. Starting from a speaker-independent model $\mathbf{\Theta}$ trained using speech data from many speakers, iteration 1 computes CMLLR transforms $\mathcal{C}_{i,1}^{-1}$ for each speaker i. A new model $\mathbf{\Theta^1}$ is then trained using the CMLLR-transformed cepstra, obtained as $\mathbf{X_i^1} = \mathcal{C}_{i,1}^{-1}(\mathbf{X_i})$, for the next iteration.

[2] However, a minimum amount of data is required for optimization to be feasible.

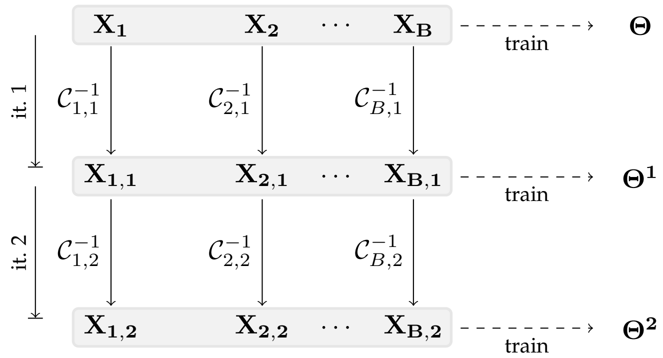

Figure 5.1 — Diagram of two iterations of speaker adaptive training (SAT).

5.4 CMLLR-SVM Base System

The CMLLR-SVM approach proposed in this section is based on the MLLR-SVM system. For a speaker segment of interest, the system uses a speaker-independent model to estimate a CMLLR transform whose coefficients are used as a representation of the speaker. These coefficients are then classified using an SVM. The main difference compared to the standard approach is that we use a GMM as speaker-independent model, which has the advantage of being easier and faster to train than the acoustic models of an LVCSR system. This eases multiple iterations of speaker adaptive training, for instance. Furthermore, such an approach does not require any transcripts as opposed to standard MLLR-SVM. It must be noted, though, that more recent MLLR-SVM approaches do not require transcripts either as they use phone-loop MLLR transforms based on a mono-lingual [Stolcke *et al.*, 2007] context-independent phone set.

The CMLLR-SVM system is structured in three stages: GMM-UBM training using a SAT approach, feature extraction based on CMLLR-transform coefficients using the refined GMM-UBM and, finally, classification using SVM. These three stages are shown in Figure 5.2 in more detail. In the following sections we focus on the two former stages, given that we use standard SVM classification for the latter, already described in Chapter 3.

5.4.1 GMM-UBM Training

In an attempt to simplify model training in the standard MLLR-SVM approach, we use a GMM-UBM as speaker-independent model. Standard MLLR-SVM systems use the speaker-independent acoustic models of a LVCSR system, typically context-dependent tri-phone models. Such models are precise in the sense that each of them are specialized in a small region of the acoustic space in a certain linguistic context. GMM-UBM can represent the whole acoustic space with arbitrary precision, but they perform frame-by-frame characterization assuming independence among observations. However, since the goal of text-independent modeling for speaker recognition is characterizing the acoustic space of a speaker and not the linguistic content of speech itself, we assume a GMM is able to provide sufficient precision.

The GMM-UBM can be trained using speech from many speakers, as usual. However, given that our goal is to compute CMLLR transforms in the most meaningful way, rendering the UBM more speaker-independent leads to more speaker-dependent transforms. We

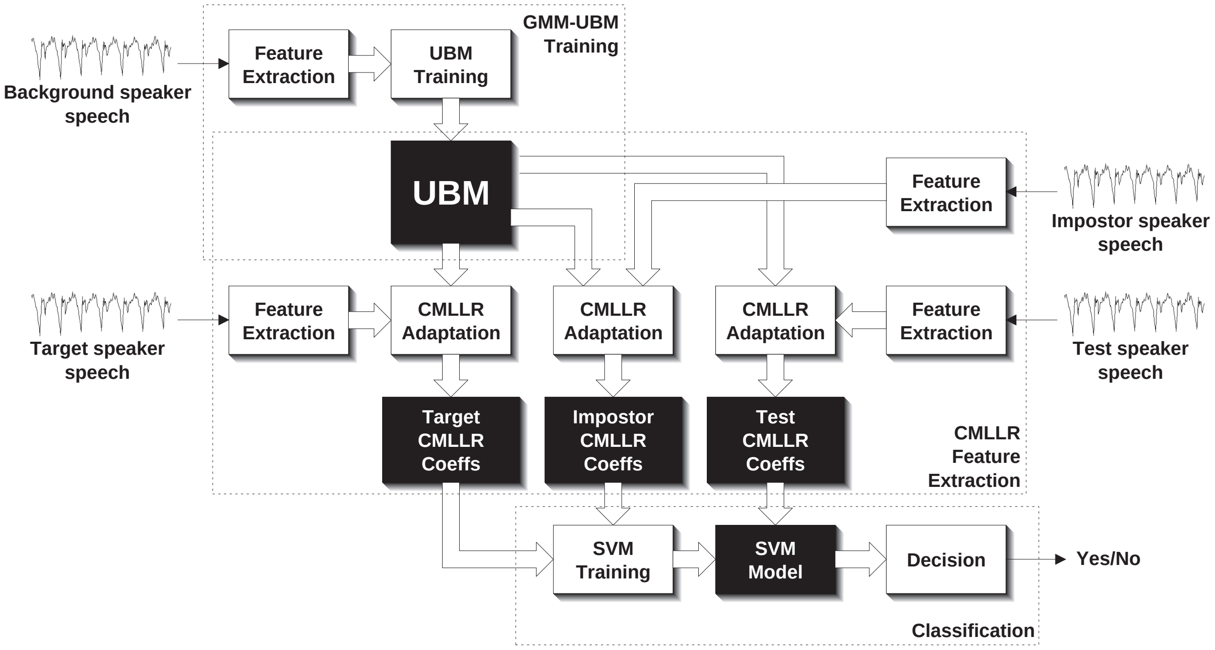

Figure 5.2 — Block diagram of the CMLLR-SVM system.

use a SAT-based approach for this purpose. Given B speech segments $\mathbf{X_s} = (\mathbf{x_{s1}}, \ldots, \mathbf{x_{sT}})$ with $1 \leq s \leq B$ and $1 \leq t \leq T$ for a certain number of speakers, we train a GMM with parameters $\boldsymbol{\Theta}$. Then, the re-estimation procedure requires iterating the following three steps:

1. Estimate one feature-space CMLLR transform, $\mathcal{C}_s^{-1}$, per segment $\mathbf{X_s}$ based on $\boldsymbol{\Theta}$. Such transforms have parameters $(\mathbf{A^{-1}}, -\mathbf{A^{-1}b})$ and project cepstra onto a speaker-independent space.

2. Apply $\mathcal{C}_s^{-1}$ onto the corresponding segment $\mathbf{X_s}$ to remove speaker-dependency. A new set of segments $\mathbf{X_s}$ is obtained.

3. Train a GMM using the speaker-independent segments $\mathbf{X_s}$, obtaining new parameters $\boldsymbol{\Theta}$.

This approach is analogous to SAT except that transforms are computed at the segment level instead of the speaker level. LVCSR systems typically collect all speech data available for every speaker to estimate a single CMLLR transform for that speaker while we do for every speaker segment instead. Given that CMLLR transforms adapt to whatever data is presented, the strong channel or session variation from segment to segment present is also captured by the regression coefficients[3]. Therefore, feature-space CMLLR transforms as used above remove linear estimates of speaker and session variability, resulting in an overall speaker- and session-independent UBM.

Being based on the EM algorithm, SAT converges to a local optimal solution of equation 5.5, within a few iterations, in practice. Using a large amount of speech data and speakers

[3] Such reasoning applies to other sources of variability present in the cepstral feature vectors, e.g. language or dialect.

assures good initial parameter estimates to bootstrap the re-estimation process. The use of a GMM-UBM allows much faster re-estimation compared to using the acoustic models of an LVCSR system, as alignment and the amount of estimated parameters is more manageable. However, iterating is still computationally expensive compared to standard ML training, especially when the amount of training data and Gaussians are large.

5.4.2 CMLLR Feature Extraction

Given a UBM trained either using a standard maximum likelihood or a SAT approach previously described, we compute CMLLR transforms for every segment of interest. Their coefficients are a parametric representation of the mapping between the speaker-independent model and the speaker data, i.e. a parametric model of the speaker. We can readily obtain a high-dimensional feature vector for the speaker by stacking regression coefficients in vector form as

$$\begin{aligned}\mathbf{m} \;=\; & [\mathbf{A_{1C}}, \ldots, \mathbf{A_{1C}}, \ldots, \mathbf{A_{D1}}, \ldots, \mathbf{A_{CC}}, \\ & \mathbf{b_1}, \ldots, \mathbf{b_C}]^T \end{aligned} \tag{5.6}$$

where C is the dimension of the cepstral vectors and $\mathbf{A}$ and $\mathbf{b}$ are matrix and offset vectors of the CMLLR transform.

CMLLR transforms for a target speaker are expected to include more speaker information if estimated using a more speaker-independent UBM. In the same line using a speaker- and session-independent UBM results in more speaker- and session-dependent CMLLR transforms. This latter contribution is rather harmful, as using SAT in a per-segment basis for UBM training we increase session variability of CMLLR supervectors as well. Depending on the ratio of actual speaker and channel variances, per-segment SAT can be better suited than regular SAT or viceversa. For instance, if cepstral features have larger speaker variance than session variance, CMLLR transforms will contain strong speaker components and weak session components overall. However, since it is not easy to assess such interactions, we believe it is preferable to gather both speaker and session variability in the CMLLR coefficients so that more sophisticated session compensation algorithms such as NAP can be used in a post-processing step (See section 5.5 for further details).

The information captured by CMLLR regression coefficients can be represented in two native forms. As a feature-space transform, CMLLR coefficients correspond to those values in the matrix $\mathbf{A^{-1}}$ and in the offset vector $\mathbf{A}^{-1}\mathbf{b}$. These capture unadaptation of the target speaker data rather than adaptation to the speaker data. Feature-space transforms can be transformed into model-space transforms by means of matrix inversion, obtaining $\mathbf{A}$ and $\mathbf{b}$. These dual representations of the adaptation process are not possible using, say, unconstrained MLLR adaptation which can be performed in a single direction only[4].

Before classification, all target, test and impostor speaker CMLLR supervectors are normalized across a large amount of speakers. As with other SVM systems developed at LIMSI (see section 4.1.3), we use affine transforms that fit each vector component into the range [-1/$\sqrt{M}$,1/$\sqrt{M}$], where $M = C^2 + C$, i.e. the number of components in the CMLLR supervectors.

[4] Model-space adaptation in the reverse direction would require a speaker-dependent model for the target speaker which is actually the goal of adaptation itself. On the other hand, also note that any feature-space transform can not adapt mean vectors only, since both the mean vector and the co-variance matrix are transformed.

			SRE 2005				SRE 2006			
			MDC		EER (%)		MDC		EER (%)	
System	Coeff.	SAT it.	F	FB	F	FB	F	FB	F	FB
CG15	$\mathcal{C}^{-1}$	×	.0368	.0344	8.98	8.40	.0326	.0304	7.07	6.89
CG15	$\mathcal{C}$	×	.0322	.0303	7.48	7.24	.0295	.0280	6.98	6.57
CG15	$\mathcal{C}^{-1}, \mathcal{C}$	×	.0333	.0315	7.86	7.57	.0302	.0286	6.71	6.43
CG15	$\mathcal{C}^{-1}$	1	.0358	.0327	7.90	7.78	.0319	.0292	7.26	6.66
CG15	$\mathcal{C}$	1	**.0308**	.0285	7.08	6.61	.0290	.0271	6.66	6.48
CG15	$\mathcal{C}^{-1}, \mathcal{C}$	1	.0317	.0292	6.90	6.79	.0293	.0271	6.57	6.21
CG15	$\mathcal{C}^{-1}$	2	.0352	.0324	8.01	7.78	.0304	.0281	6.76	6.66
CG15	$\mathcal{C}$	2	**.0308**	**.0280**	6.82	6.60	**.0276**	**.0258**	6.74	**6.06**
CG15	$\mathcal{C}^{-1}, \mathcal{C}$	2	.0325	.0288	**6.73**	**6.49**	.0280	.0262	**6.24**	**6.06**

Table 5.1 — MDC and EER of CMLLR-SVM systems as a function of the transform type and number of SAT iterations used in UBM training for SRE 2005 and SRE 2006. Systems with $\mathcal{C}^{-1} : \hat{\mathbf{x}} \mapsto \mathbf{A}^{-1}\hat{\mathbf{x}} - \mathbf{A}^{-1}\mathbf{b}$ use feature-space CMLLR transform coefficients, systems with $\mathcal{C} : \boldsymbol{\mu} \mapsto \mathbf{A}\boldsymbol{\mu} + \mathbf{b}$ use model-space CMLLR transform coefficients and systems using $\mathcal{C}^{-1}\mathcal{C}$ use the concatenation of feature-space and model-space CMLLR transform coefficients. Columns F and FB show forward and averaged forward-backward scores respectively. The best scores in each column are shown in boldface.

5.4.3 Experimental Results

In this section we evaluate performance of the CMLLR-SVM base system along two axis: type of CMLLR coefficients and number of SAT iterations in UBM training. We explore the former using feature-space CMLLR transform coefficients, model-space CMLLR coefficients obtained by inversion[5] of feature-space transforms and the concatenation of both types of coefficients. These three configurations are named $\mathcal{C}^{-1}$, $\mathcal{C}$ and $\mathcal{C}^{-1}\mathcal{C}$ respectively. We study each of these as a function of the number of iterations used in SAT, i.e. none, one or two, which correspond to the three blocks of results identifiable in Table 5.1. For the sake of readability we introduce the acronym CG15 (CMLLR using a GMM and a front-end with 15 PLP coefficients) for CMLLR-SVM systems in this chapter[6]. The base UBM are two gender-dependent GMM with 512 Gaussians each trained using 5 iterations of maximum likelihood estimation and a variance floor of 1% of the total variance. Using the front-end described in Section 4.1.1, CMLLR transform supervectors result in 2256 coefficients, $39 \cdot 39 + 39$, including offset vector $\mathbf{b}$. We discuss the results of these experiments, shown in Table 5.1, in the following.

Absolute performance for SRE 2006 is systematically better than for SRE 2005, which is a general trend caused by structural differences between both corpora. Regarding the use of $\mathcal{C}^{-1}$, $\mathcal{C}$ and $\mathcal{C}^{-1}\mathcal{C}$ coefficients, using model-space transform coefficients $\mathcal{C}$ significantly outperforms feature-space transform coefficients $\mathcal{C}^{-1}$ with gains of 10%-15% MDC and EER for SRE 2005 and a bit lower, 7%-9% MDC for SRE 2006, all for both forward and forward-

[5] We used LU matrix inversion as CMLLR transforms depend directly on the data used for adaptation as well as the front-end. We assumed no a priori structure for these matrices.

[6] This adds naming homogeneity across the whole manuscript, as further chapters use a wide range of configurations which might confuse the reader.

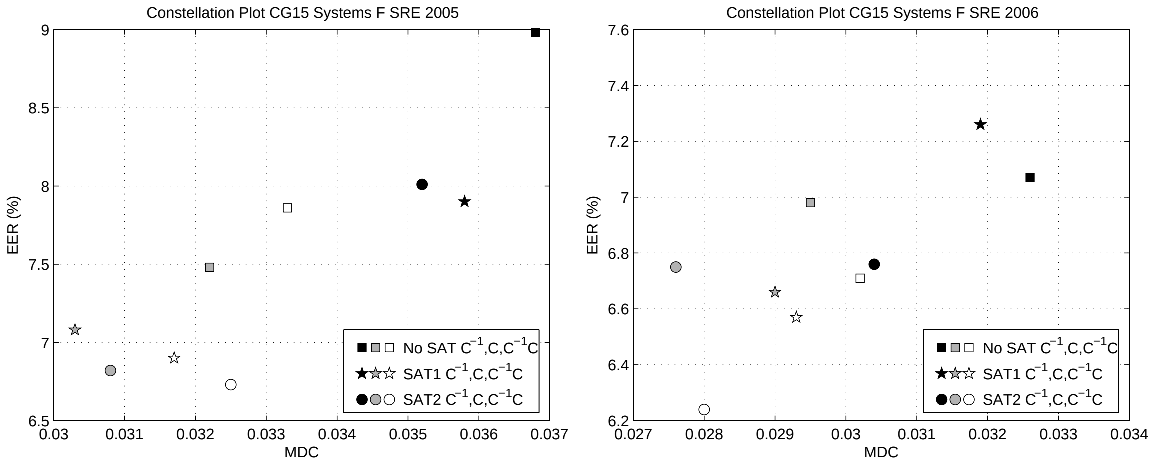

Figure 5.3 — Constellation plots of CG15 systems using forward scoring for SRE 2005 (left, a) and SRE 2006 (right, b). Systems using regular UBM training, 1 SAT iteration and 2 SAT iterations use symbols ■, $\star$ and • respectively. Colors black, gray or white correspond to $\mathcal{C}^{-1}$, $\mathcal{C}$ and $\mathcal{C}^{-1}\mathcal{C}$ coefficients.

backward scoring. These are rather surprising results given that both feature-space and model-space transforms focus on the adaptation between model and data, although they do in the opposite direction. It seems that a linear SVM classifier is able to more easily classify the model adaptation direction. We retake this issue in section 5.6 where we discuss alternative representations of transform coefficients. Using concatenated $\mathcal{C}^{-1}$ and $\mathcal{C}$ coefficients as features results in important improvements compared to using $\mathcal{C}^{-1}$ coefficients alone, but rarely outperforming $\mathcal{C}$ alone. Such behavior is rather expectable, given that both sets of coefficients are highly correlated although not in a linear way. Figures 5.3(a) and 5.3(b) show constellation plots of all systems in Table 5.1 for SRE 2005 and SRE 2006. Note that points are rather clustered based on the coefficients used, i.e. color-wise, with an overall trend towards the lower-left corner as more SAT iterations are used, i.e. ■, $\star$ and • symbols respectively. Figures 5.4(a) and 5.4(b) show DET curves for the first three systems in Table 5.6 using forward scoring for SRE 2005 and SRE 2006. Model-space transform coefficients outperform feature-space coefficients while their combination lies in between for almost all operating points in the curve. For SRE 2006, systems using $\mathcal{C}$ and $\mathcal{C}^{-1}\mathcal{C}$ coefficients have almost overlapped DET curves. Given that systems using one and two SAT iterations for UBM training show the same behavior, we judged it not necessary to include DET curves for these systems.

Speaker adaptive training of the UBM brings interesting gains of performance. An average improvement of 5% is obtained using one SAT iteration compared to regular training and an additional 2.5% using a second iteration, summing up to a 7.5% of relative improvement in two SAT iterations. Only slightly higher for EER, these gains are rather balanced for both MDC and EER measures and a significant positive correlation is found with the type of coefficients used. For instance, using SAT in systems using $\mathcal{C}^{-1}\mathcal{C}$ transform coefficients achieves relative gains around 25% larger than systems using $\mathcal{C}^{-1}$ coefficients, while $\mathcal{C}$ coefficients lie in between. It has been observed that performing more iterations results in progressively smaller gains, which effectively suggests that the UBM is converging to the optimal parameters attainable within the CMLLR framework. Figures 5.5(a) and 5.5(b) show DET curves of the CG15 $\mathcal{C}$ system using regular training, 1 SAT or 2 SAT iterations. The cumulative effect of multiple SAT iterations on performance is appreciable along most

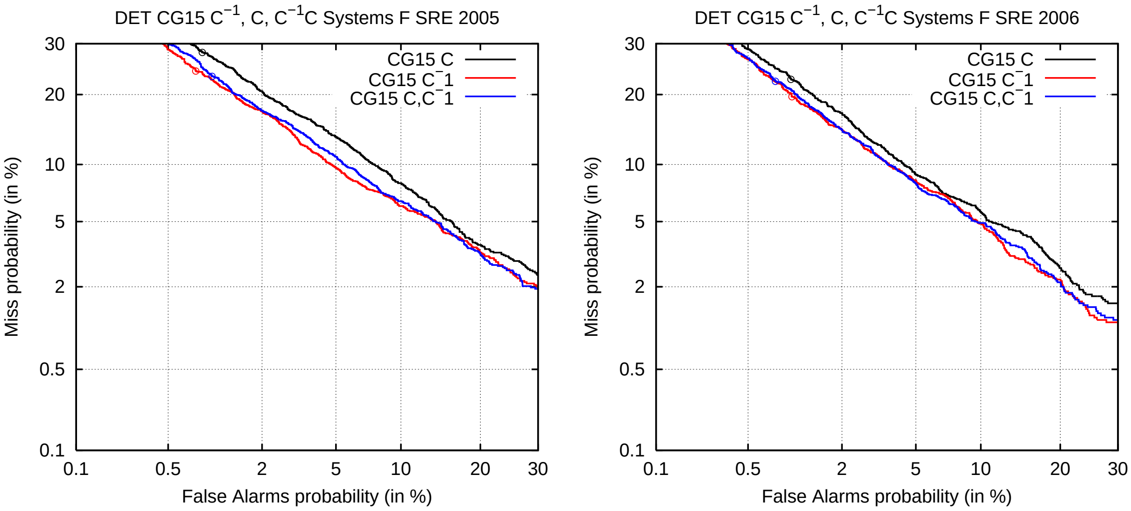

Figure 5.4 — DET curves of CG15 $\mathcal{C}^{-1}$, CG15 $\mathcal{C}$ and CG15 $\mathcal{C}^{-1}\mathcal{C}$ systems using ML training and forward scoring for SRE 2005 (left, a) and SRE 2006 (right, b).

of the operating points in the DET curve.

Regarding forward-backward scoring, we observe slight improvements systematically compared to using forward scoring only. Gains are around 4% MDC and 2% EER in average, although the can reach 10% for certain systems. There is a slight positive correlation between these gains and the number of SAT iterations used. Figures 5.6(a) and 5.6(b) show forward and forward-backward scoring DET curves for CG15 $\mathcal{C}$ systems using regular UBM training and 2 SAT iterations. We observe that DET curves for each of the systems do not cross each other, indicating that gains are rather stable for all operating points, with the exception of the high false-alarm region for SRE 2006.

Therefore, we have found three sources of improvement for CMLLR-SVM systems, namely the coefficients used, the amount of SAT iterations performed for UBM training and forward-backward scoring. The results in Table 5.1 are still far from those of the baseline systems, see Table 4.1 presented in chapter 4. A major difference between CMLLR-SVM and the baseline systems is the use of NAP inter-session variability compensation of the base supervectors. We address this subject in the next section.

5.5 CMLLR-SVM with NAP Inter-session Variability Compensation

In this section we study the behavior of CMLLR-SVM systems together with NAP inter-session variability compensation. Prior to evaluating CMLLR-SVM systems, we give a theoretical interpretation of NAP in the context of (C)MLLR transform supervectors.

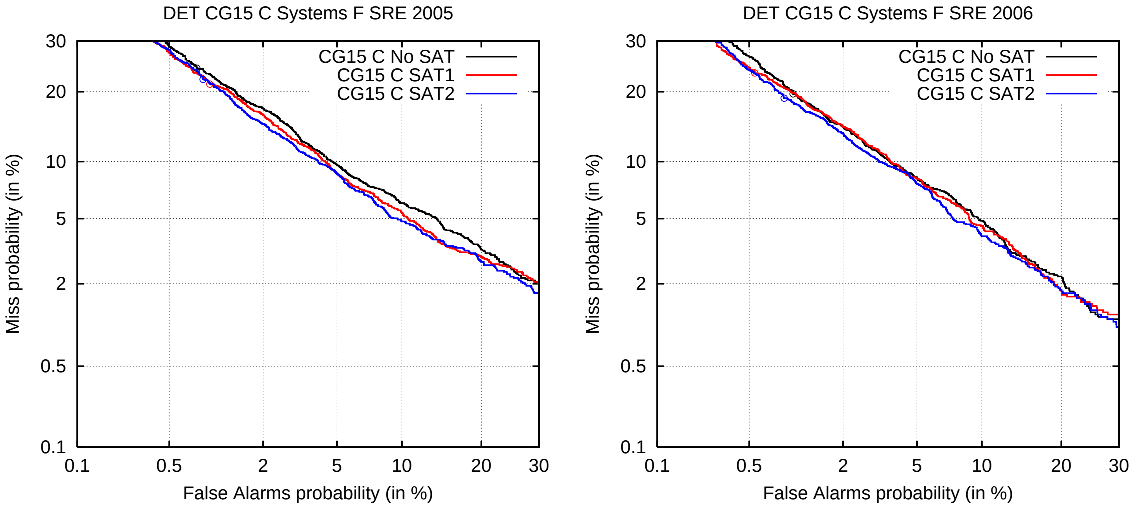

Figure 5.5 — DET curves of CG15 $\mathcal{C}$ systems using ML training, 1 SAT, 2 SAT iterations and forward scoring for SRE 2005 (left, a) and SRE 2006 (right, b).

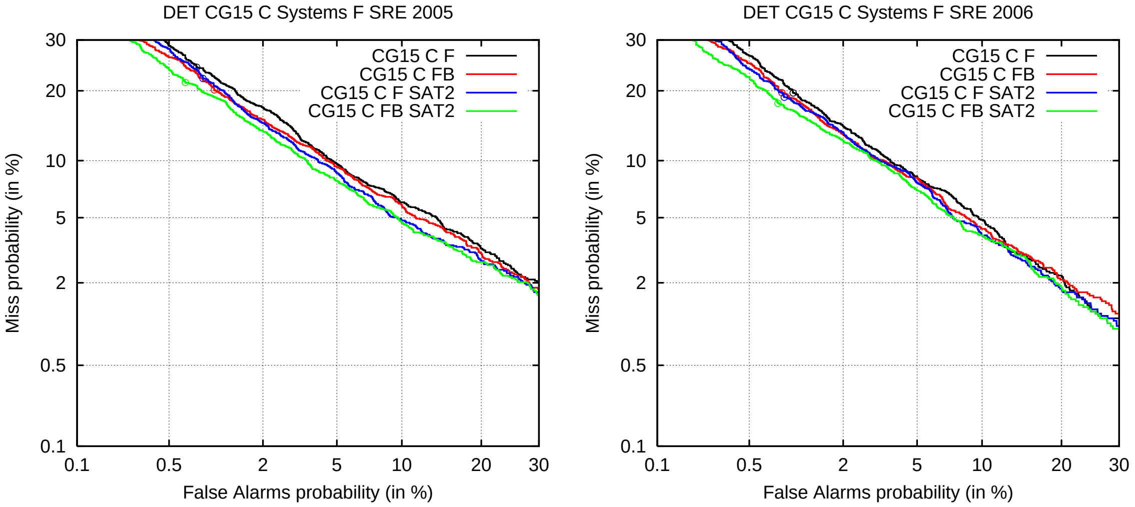

Figure 5.6 — DET curves of CG15 $\mathcal{C}$ systems using ML training and 2 SAT iterations. We show two curves per system corresponding to forward and forward-backward scoring for SRE 2005 (left, a) and SRE 2006 (right, b).

5.5.1 Eigenchannels, NAP and CMLLR Transforms

For the most part, inter-session variability compensation algorithms make the assumption that the observed utterance can be linearly decoupled into speaker and session contributions. In Eigenchannel modeling, the variability model of Equation 2.30 can be rewritten in terms of GMM mean vectors as

$$\begin{aligned} \boldsymbol{\mu}^{\boldsymbol{i}}_{\boldsymbol{sh}} &= \boldsymbol{\mu}^{\boldsymbol{i}}_{\boldsymbol{s}} + \boldsymbol{\mu}^{\boldsymbol{i}}_{\boldsymbol{h}} && (5.7) \\ &= \boldsymbol{\mu}^{\boldsymbol{i}}_{\boldsymbol{s}} + \mathbf{U^i x_h} && (5.8) \end{aligned}$$

that is, speaker-only and session-only components $\boldsymbol{\mu}^{\boldsymbol{i}}_{\boldsymbol{s}}$ and $\boldsymbol{\mu}^{\boldsymbol{i}}_{\boldsymbol{h}} = \mathbf{U^i x_h}$ which add together to make up the observed speaker- and session-dependent mean vector $\boldsymbol{\mu}^{\boldsymbol{i}}_{\boldsymbol{sh}}$ for Gaussian i. The term $\mathbf{U^i x_h}$ estimates the session contribution for this Gaussian, with $\mathbf{x_h}$ being the session factors and $\mathbf{U^i}$ a rectangular matrix relating session factors and mean GMM vectors. Eigenchannel adaptation seeks maximum-likelihood or maximum-a-posteriori estimates of $\mathbf{m_s}$ and $\mathbf{x_h}$ assuming $\mathbf{U^i}$ is known.

Maximum-likelihood adaptation of GMM can be also performed using model-space CMLLR transforms, in the same way that NAP compensation of GMM supervectors can be considered equivalent to Eigenchannel modeling [Campbell *et al.*, 2006] in its simplest form. Based on a UBM with N Gaussians as speaker-independent model and assuming a single regression class, CMLLR adaptation can be written in terms of operator $\mathcal{C}_{sh}$ as

$$\boldsymbol{\mu}^{\boldsymbol{i}}_{\boldsymbol{sh}} = \mathbf{A_{sh}}\boldsymbol{\mu}^{\boldsymbol{i}} + \mathbf{b_{sh}} = \mathcal{C}_{sh}(\boldsymbol{\mu}^{\boldsymbol{i}}) \tag{5.9}$$

where $\boldsymbol{\mu}^{\boldsymbol{i}}$ and $\boldsymbol{\mu}^{\boldsymbol{i}}_{\boldsymbol{sh}}$ are GMM mean vectors of the UBM and the adapted model, and subscripts s and h stand for speaker and session. Thus, if we want to include session compensation into the CMLLR adaptation framework, we must deal with $\mathcal{C}_{sh}$ given that $\boldsymbol{\mu}^{\boldsymbol{i}}$ are related to the UBM only. We can think of decomposing $\mathcal{C}_{sh}$ into speaker-only and session-only components, $\mathcal{C}_s : \mathbf{x} \mapsto \mathbf{A_s x} + \mathbf{b_s}$ and $\mathcal{C}_h : \mathbf{x} \mapsto \mathbf{A_h x} + \mathbf{b_h}$ respectively for this purpose. Given that these operate on cepstral vectors, we can use the decomposition of Equation 5.7 to obtain

$$\begin{aligned} \boldsymbol{\mu}^{\boldsymbol{i}}_{\boldsymbol{sh}} &= \boldsymbol{\mu}^{\boldsymbol{i}}_{\boldsymbol{s}} + \boldsymbol{\mu}^{\boldsymbol{i}}_{\boldsymbol{h}} \\ &= \mathcal{C}_s(\boldsymbol{\mu}^{\boldsymbol{i}}) + \mathcal{C}_h(\boldsymbol{\mu}^{\boldsymbol{i}}) && (5.10) \\ &= (\mathcal{C}_s + \mathcal{C}_h)(\boldsymbol{\mu}^{\boldsymbol{i}}) \\ &= \mathcal{C}_{sh}(\boldsymbol{\mu}^{\boldsymbol{i}}) \end{aligned}$$

where we assumed $\mathcal{C}_{sh} = \mathcal{C}_s + \mathcal{C}_h$. We address such decomposition using a supervector formulation of CMLLR transforms, as used in Eigen-MLLR adaptation [Chen *et al.*, 2000]. Since matrix and offset addition further correspond to regular addition of the corresponding components in the supervector space, then

$$\mathbf{c_{sh}} = \mathbf{c_s} + \mathbf{c_h} \tag{5.11}$$

where $\mathbf{c_{sh}}$, $\mathbf{c_s}$ and $\mathbf{c_h}$ are supervectors obtained by stacking all coefficients of CMLLR transforms $\mathcal{C}_{sh}$, $\mathcal{C}_s$ and $\mathcal{C}_h$ respectively. A current technique allowing the decomposition of Equation 5.11 is NAP (see Section 3.3.1), which estimates the session subspace as the subspace spanned by those vectors maximizing the inter-session covariance matrix. In practice, NAP reduces to PCA analysis of session-wise variability of $\mathbf{c_{sh}}$ supervectors, which requires a multi-speaker and multi-session database. Retaking NAP compensation in Equation 3.44, speaker-only and session-only components are obtained by projecting $\mathbf{c_{sh}}$ onto complementary speaker and session subspaces as

	Estimator	
Scheme	Speaker	Session
Eigenchannel	$\boldsymbol{\mu}_{\boldsymbol{s}}^{\boldsymbol{i}}$	$\mathbf{U^i x_h}$
CMLLR+NAP	$\mathcal{C}_s(\boldsymbol{\mu}^{\boldsymbol{i}})$	$\mathcal{C}_h(\boldsymbol{\mu}^{\boldsymbol{i}})$

Table 5.2 — Comparison of speaker and session GMM mean vector estimation for Eigenchannel and CMLLR+NAP compensation.

$$\mathbf{c_s} = (\mathbf{I} - \mathbf{E}\mathbf{E}^T)\mathbf{c_{sh}} \tag{5.12}$$

$$\mathbf{c_h} = (\mathbf{E}\mathbf{E}^T)\mathbf{c_{sh}} \tag{5.13}$$

$\mathbf{E} = (\mathbf{e_1}, \ldots, \mathbf{e}_K)$ are the eigenvectors corresponding to the K largest eigenvalues of the inter-session covariance matrix. $\mathbf{E}\mathbf{E}^T$ projects supervector $\mathbf{c_{sh}}$ onto the session subspace first, with K dimensions only, and projects the resulting vector back to the original supervector space[7]. Conversely, $\mathbf{I} - \mathbf{E}\mathbf{E}^T$ performs projection onto the speaker subspace, which is the complement of the session subspace.

In light of the discussion above, model-level inter-session variability compensation can be achieved by compensating CMLLR transforms in the supervector space. Table 5.2 provides a comparison of speaker and session estimators for both methods. We find two major divergences about them. CMLLR+NAP uses the same linear transform to estimate speaker and session components regardless of the Gaussian to be compensated while Eigenchannel invests specific parameters for each Gaussian, namely speaker component $\boldsymbol{\mu}_{\boldsymbol{s}}^{\boldsymbol{i}}$ and matrix $\mathbf{U^i}$. Hence, compensation is dependent on the region of the acoustic space for Eigenchannel modeling, but not for CMLLR+NAP. Nonetheless, this can be overcome by using several CMLLR transforms, each specialized in a certain region of the acoustic space. A second difference is that the number of parameters in CMLLR+NAP compensation transforms is dependent on the number of cepstral features regardless of the dimension of the session subspace K. While matrix $\mathbf{U^i}$ grows linearly with K, involving more complex compensation, CMLLR+NAP maps the overall effect of the compensating transforms back into the same amount of parameters. Although the dimension of the session subspace is assumed to be generally small, this may be a major limitation for CMLLR+NAP compensation.

We focus next on experimental work regarding CMLLR-SVM using NAP compensation. We further discuss NAP compensation of CMLLR transforms, namely its application to cepstra, in Section 5.7.

5.5.2 Experimental Results

In this section we assess performance of CMLLR-SVM systems using NAP-compensated CMLLR transform coefficients as features. These transforms are computed using a channel-compensated front-end using feature mapping as described in Section 4.1.1. We use the version of NAP maximizing Equation 3.46 as used in all SVM-based systems (see Section 4.1.3) in this manuscript. We explore CMLLR-SVM systems using model-space transform

[7] The eigenvector matrix $\mathbf{E}$ is obtained as the solution of a symmetric eigenvalue problem, hence an orthonormal matrix.

				SRE 2005				SRE 2006			
				MDC		EER (%)		MDC		EER (%)	
System	Coeff.	SAT	D_{NAP}	F	FB	F	FB	F	FB	F	FB
CG15	$\mathcal{C}$	×	×	.0322	.0303	7.48	7.24	.0295	.0280	6.98	6.57
CG15	$\mathcal{C}$	1	×	.0308	.0285	7.08	6.61	.0290	.0271	6.66	6.48
CG15	$\mathcal{C}$	2	×	.0308	.0280	6.82	**6.60**	.0276	.0258	6.74	6.06
CG15	$\mathcal{C}$	×	25	.0297	.0286	7.69	7.56	.0263	.0248	5.83	5.70
CG15	$\mathcal{C}$	1	25	.0291	.0271	7.07	6.82	.0252	.0246	5.65	5.47
CG15	$\mathcal{C}$	2	25	.0284	.0264	**6.70**	6.61	.0256	.0240	5.69	5.42
CG15	$\mathcal{C}$	×	50	.0303	.0276	7.93	7.86	.0258	.0241	5.96	5.61
CG15	$\mathcal{C}$	1	50	.0292	.0265	7.27	6.99	.0247	.0241	5.88	5.61
CG15	$\mathcal{C}$	2	50	.0285	**.0256**	6.74	6.70	.0247	.0238	5.65	**5.38**
CG15	$\mathcal{C}$	×	75	.0307	.0284	8.02	7.98	.0255	.0239	5.87	5.61
CG15	$\mathcal{C}$	1	75	.0296	.0273	7.57	7.61	.0251	.0237	5.61	5.51
CG15	$\mathcal{C}$	2	75	**.0283**	.0261	7.36	7.31	**.0244**	**.0227**	**5.56**	5.47

Table 5.3 — MDC and EER of CMLLR-SVM systems as a function of the number of SAT iterations and session subspace dimension for SRE 2005 and SRE 2006. Systems use model-space CMLLR transform coefficients, i.e. $\mathcal{C} : \boldsymbol{\mu} \mapsto \mathbf{A}\boldsymbol{\mu} + \mathbf{b}$. Columns F and FB show forward and averaged forward-backward scores respectively. The best scores in each column are shown in boldface.

coefficients given that performance obtained for these features in Section 5.4.3 was substantially better than that obtained using feature-space transform coefficients. We also study the interaction between NAP and the number of SAT iterations in the UBM.

Table 5.3 shows MDC and EER measures of CMLLR-SVM systems using model-space CMLLR transform coefficients as features, i.e. $\mathcal{C}$. We first focus on the number of dimensions used in the NAP session subspace, thus keeping the number of SAT iterations fixed. Average relative gains are in the range 10%-15% MDC and EER for SRE 2006, getting larger as we introduce more session subspace dimensions. NAP should probably be explored using more dimensions for this corpus. These improvements are obtained using a front-end which already uses feature mapping channel compensation. Therefore, NAP addresses session compensation in ways feature mapping cannot, which is not surprising. Feature mapping uses discrete estimates of channel contributions while NAP, which is equivalent to Eigenchannel modeling for GMM mean supervectors in its simplest form, uses continuous session contributions. Figure 5.7(a) shows DET curves for these systems for SRE 2006. Improvements obtained by NAP are consistent along all operating points, but systems differing in the number of session subspace dimension show in very close curves. This can be roughly explained if eigenvalues of the inter-session covariance matrix are assumed to decay exponentially. Since most of the inter-session variance is captured by the first eigenvalues further including more and more dimensions results in smaller and smaller amounts of variance in the session subspace. We observe a different scenario for SRE 2005 data. In this corpus, we see slight improvements in MDC, around 4%-8% while EER is systematically degraded by NAP. This effect is noticeable as a slight rotation of the DET curves for systems using NAP, as shown in Figure 5.7(a). Such a different behavior for both corpora can be explained by the amount of channel variability included in SRE 2006. Only hand-held

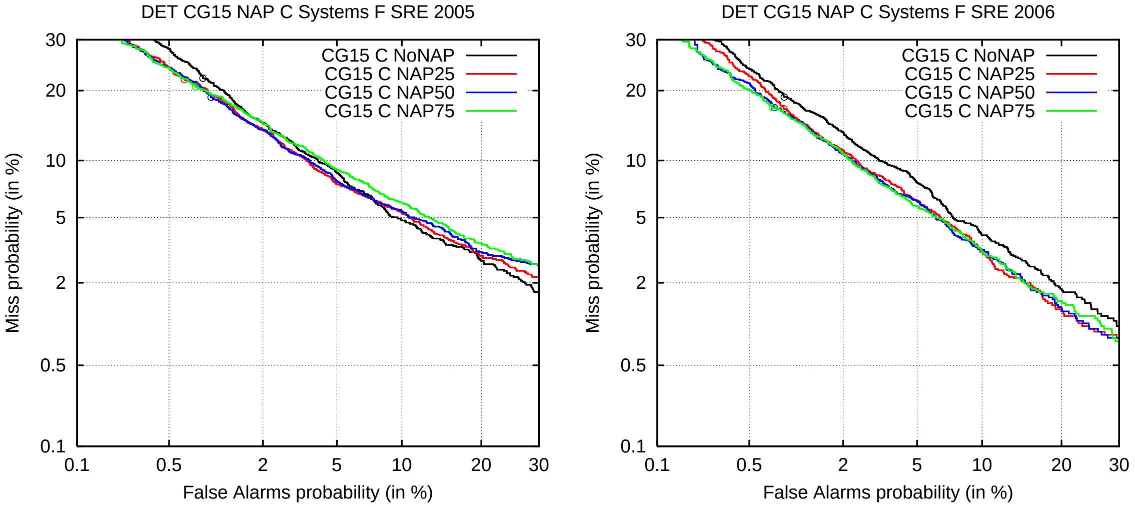

Figure 5.7 — DET curves of CG15 $\mathcal{C}$ systems using two SAT iterations and 0, 25, 50 and 75 dimensions for the NAP session subspace for SRE 2005 (left, a) and SRE 2006 (right, b).

instruments are used in the common condition of SRE 2005 while speaker-phone, head-mounted, ear-bud and hand-help instruments are used for SRE 2006. Thus, NAP shows to be effective in scenarios with a large channel or session variability. Figures 5.8(a) and 5.8(b) show constellation plots of all of the systems on SRE 2005 and SRE 2006 corpora. For SRE 2006, note the large gap between systems using NAP and regular CMLLR-SVM as well as the global trend towards the bottom-left corner, i.e. improving performance, as we move from $\star$ to $\bullet$ and $\diamond$ symbols. For SRE 2005, the constellation plot is more sparse with regular CMLLR-SVM systems and systems using NAP clustered on the right and left hand sides of the graph respectively. Given the large dispersion of performance, we decide on using 50 dimensions of the session subspace. This is the setting we keep for further CMLLR-SVM systems and besides it is the same used for all other SVM-based systems in this manuscript.

The constellation plots of Figure 5.8 also highlight the interaction of NAP and the number of SAT iterations. See how systems are visually clustered by color in Figure 5.8(a), i.e for SRE 2005. Systems using regular training are clustered the top-right low performing area of the graph while systems using 2 SAT iterations are around the bottom-left high performing area. This indicates that improvement due to SAT is rather dominant over that obtained by NAP in this corpus. Note that the main clustering criterion is swapped for SRE 2006, with two major clusters corresponding to systems using or not using NAP. Table 5.3 allows performance comparison for systems using different SAT setups. We observe that SAT brings significant gains for most of the configurations, eventually reaching 7% MDC and 15% EER for SRE 2005 and up to 5% MDC and EER for SRE 2006 using two SAT iterations and 50 dimensions for the NAP session subspace. This performance gap can be due to the presence of more channel variability in SRE 2006[8]. The more channel variability in the corpus the more channel-dependent CMLLR transform supervectors are (see Section 5.4.1) thus becoming more difficult to compensate. Average SAT gains are around 6.5%, similar to 7% obtained for regular CMLLR-SVM systems explored in Section 5.4.3 using 2 SAT iterations. This indicates that NAP and SAT are rather independent sources of performance improvement.

[8] We focus on the common condition here.

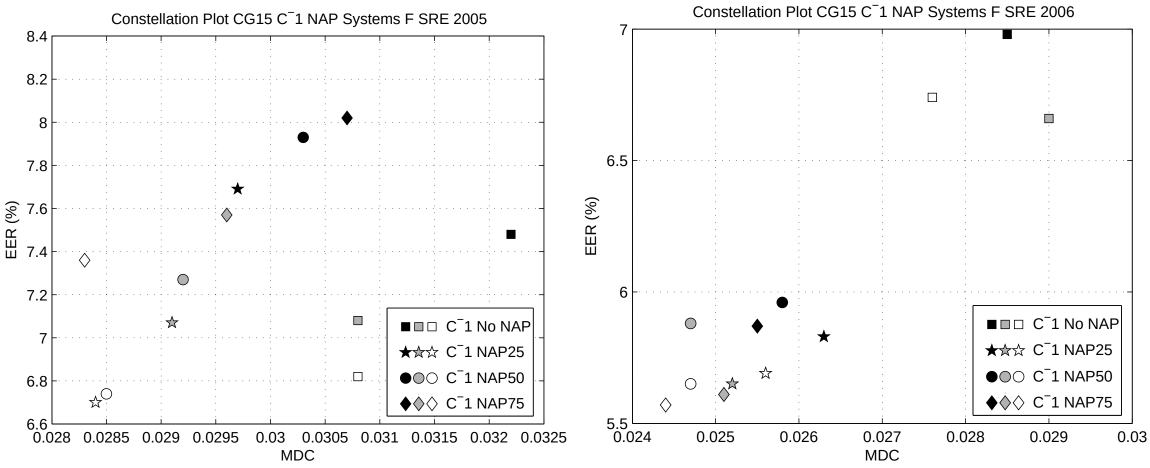

Figure 5.8 — Constellation plots of CG15 systems using forward scoring for SRE 2005 (left, a) and SRE 2006 (right, b). Systems with NAP compensation using 0, 25, 50 and 75 dimensions for the session subspace use symbols $\blacksquare$, $\star$, $\bullet$ and $\diamond$. Colors black, gray and white correspond to systems using regular UBM training, 1 SAT iteration and 2 SAT iterations.

Forward-backward scoring is still an effective way to get further gains for CMLLR-SVM systems using NAP. We find relative improvements around 7% MDC but none in terms of EER for SRE 2005 and around 4% MDC and EER for SRE 2006. A slight positive correlation between MDC improvement and the number of dimensions used in NAP is also observed. Figures 5.9(a) and 5.9(b) show DET curves for CMLLR-SVM $\mathcal{C}$ using two SAT iterations and 50 dimensions for the NAP subspace. Forward-backward scoring improves performance in the low false-alarm region but is useless for the rest of the curve for SRE 2005. Gains for SRE 2006 are smaller but more uniform over the whole range of operating points.

5.6 CMLLR-SVM with Alternative Representations of CMLLR Transforms

We have seen in Section 5.4 that CMLLR allows for two different representations of transform coefficients depending on the direction of the adaptation process, i.e. transforming cepstra by means of feature-space CMLLR to match the speaker-independent model and transforming mean and variance parameters of the model to match the adaptation data by means of model-space CMLLR. This is no issue as long as we deal with transformed data and models, as it is common in more classical approaches to speaker recognition. However, we have seen in Section 5.4.3 that both sets of coefficients lead to substantially different error rates when classified using a linear kernel SVM. We get further insight into this question in the following section. We propose alternative representations of CMLLR transform coefficients later and we compare these approaches to standard CMLLR transform coefficients experimentally.

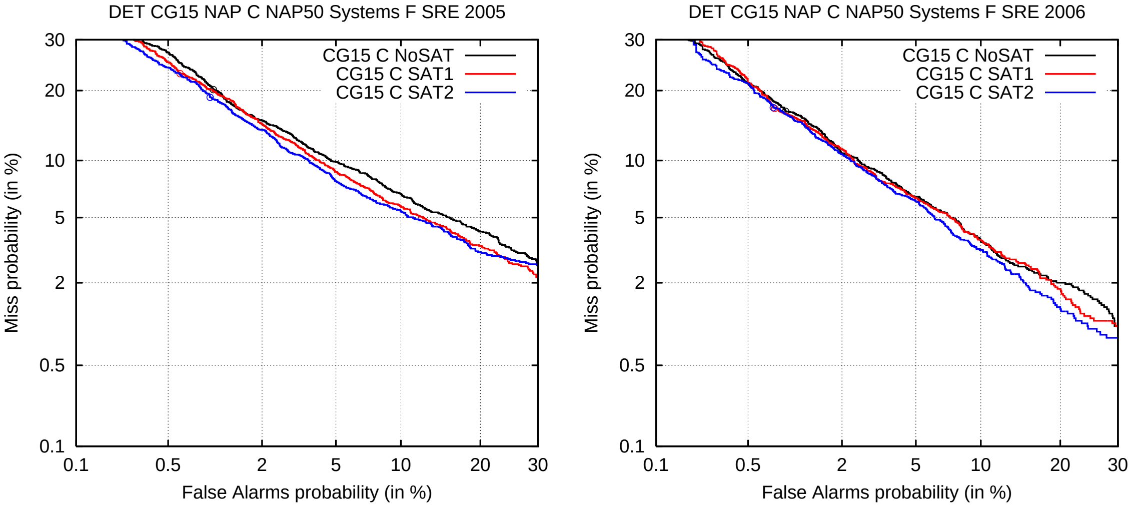

Figure 5.9 — DET curves of CG15 $\mathcal{C}$ systems using 50 dimensions for the session subspace and zero, one and two SAT iterations for SRE 2005 (left, a) and SRE 2006 (right, b).

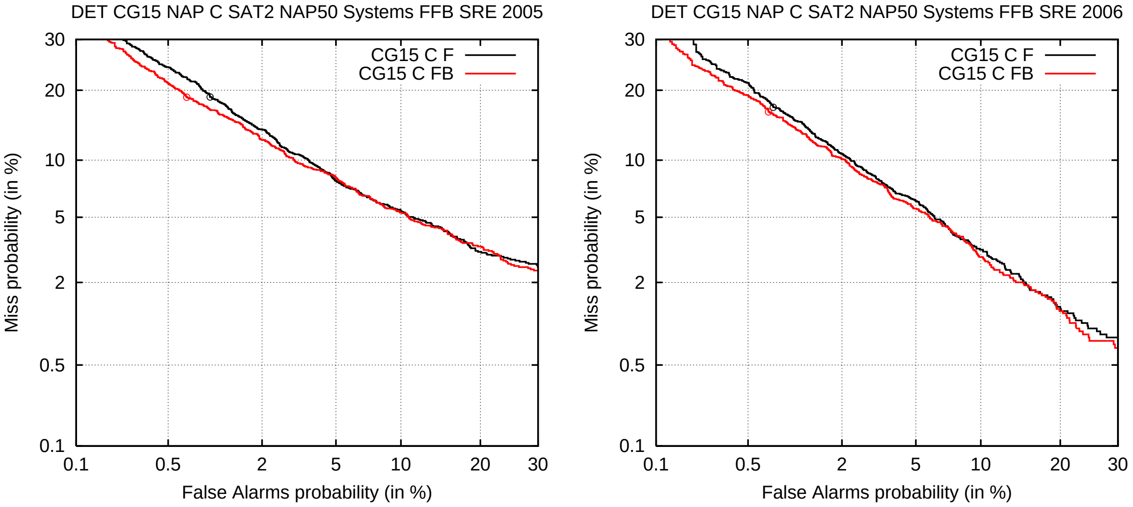

Figure 5.10 — DET curves of CG15 $\mathcal{C}$ systems using two SAT iterations and 50 dimensions for the session subspace and forward and forward-backward scoring for SRE 2005 (left, a) and SRE 2006 (right, b).

5.6.1 Relating Model-space and Feature-space CMLLR Transforms

A first thing to note is that feature-space and model-space CMLLR coefficients represent the same information, given that we can pass from one to the other back and forth just by operating on the transform coefficients themselves. GMM-UBM systems inherently manage the ambiguity in the direction of the adaptation process by dealing with transformed data and models only. For instance, these systems compute the likelihood ratio of a target adapted model and the UBM, but the way adaptation is performed to obtain these models is irrelevant as long as they are result in the right models. It does not matter whether they are obtained by adaptation of the UBM to the speaker data or by adaptation of a target speaker model to the UBM data. However, when dealing with transform coefficients themselves representation plays an important role, as each adaptation direction results in actual different features. In this case, how the issue is dealt with depends on the power of the model used. Linear kernel SVM use a regular dot product to decide in which side of the decision boundary transforms coefficients lie. It seems clear that a dot product is far less a powerful operation compared to matrix inversion and linear SVM cannot compensate for representation transformations of such complexity. This is probably a major reason that explains the difference in performance for CMLLR-SVM systems using feature-space or model-space coefficients.

From the discussion above, we envisage two approaches to cope with the issue. One is to use non-linear kernels that are capable of exploiting the underlying information more efficiently. In this sense, matrix kernels are better candidates than kernels working at the supervector level, given that ambiguity is encoded in matrix structure, not accessible in the supervector space. A second approach is transforming each CMLLR transform individually to obtain a more robust or performing representation regarding direction of adaptation. For the sake of simplicity, we focus on the latter approach.

Feature-space and model-space CMLLR transforms keep an inverse relationship. If a model-space transform has parameters $(\mathbf{A}, \mathbf{b})$ its feature-space counterpart has parameters $(\mathbf{A^{-1}}, -\mathbf{A^{-1}b})$. The building block for inverting such affine transforms is thus matrix inversion. Leaving offsets aside, most of the coefficients are affected by matrix inversion. Focusing on the matrices involved in these transforms, $\mathbf{A}$ and $\mathbf{A^{-1}}$ keep a tight relation although it may not be visible in the coefficients. If we factor both matrices using Singular Value Decomposition[9] [Golub & Loan, 1996] (SVD) we obtain

$$\begin{aligned} \mathbf{A} &= \mathbf{USV^T} && (5.14) \\ \mathbf{A^{-1}} &= \mathbf{VS^{-1}U^T} = (\mathbf{US^{-1}V^T})^T && (5.15) \end{aligned}$$

where the columns of $\mathbf{U} = (\mathbf{u_1}, \ldots, \mathbf{u_C})$ and $\mathbf{V} = (\mathbf{v_1}, \ldots, \mathbf{v_C})$ are two sets of C orthonormal vectors, so-called singular vectors, with C being the dimension of cepstral vectors. In the case that $\mathbf{A}$ is a square matrix, $\mathbf{S} = diag(\sigma_1, \ldots, \sigma_C)$ with $\sigma_1 \geq \ldots \geq \sigma_C$ is a diagonal matrix with C semi-positive values in decreasing order, so-called singular values. These singular values linearly weight left and right singular vectors so that the original matrix is recovered. From Equation 5.14 and right-hand side of Equation 5.15 we see that other than transposition, $\mathbf{A}$ and $\mathbf{A^{-1}}$ differ in their singular values only, i.e. $\mathbf{S}$ vs. $\mathbf{S^{-1}}$. Transposition causes a permutation of the transform coefficients in the supervectors but results in the same actual features and, hence, the same classification performance as well. Therefore, under a SVD-based factorization model, a major cause explaining the difference of performance for systems using feature-space and model-space CMLLR transform coefficients is their singular values. We have obviated here the effect of the offset coefficients of the affine transform as we assume that it represents a very small amount of features[10] in a supervector and experimentation has shown that error rates are very slightly affected by including or removing them.

[9] SVD assures proper factorization of any matrix regardless of its structure.

[10] Around 2%, $\frac{47}{47 \cdot 48}$ for 47 cepstral features in the front-end used in our experiments.

5.6.2 Alternative Representations of CMLLR Transforms

We have seen that singular values of SVD-factorized CMLLR transform matrices can explain the performance gap observed between model-space and feature-space CMLLR transform coefficients. In the following we explore alternative representations of these matrices so that robustness against direction of adaptation or performance are improved.

5.6.2.1 Singular Vectors

Given that singular value matrices $\mathbf{S}$ and $\mathbf{S^{-1}}$ can be deemed as the source of the observed difference in performance, we can think of using matrices $\mathbf{U}$ and $\mathbf{V}$ only to represent the adaptation process. SVD assures unique factorization of matrix $\mathbf{A}$ except for sign ambiguity[11] of pairs of singular vectors $\mathbf{u_i}$ and $\mathbf{v_i}$ and permutation of singular values and vectors[12]. If these two sources of ambiguity are resolved the adaptation process would possibly be more robustly represented by matrices $\mathbf{U}$ and $\mathbf{V}$ alone, hence avoiding the use of $\mathbf{S}$ or $\mathbf{S^{-1}}$ and focusing on a internal representation of matrix $\mathbf{A}$ independent of the direction of adaptation. Such an approach would stack all of the singular vectors of both matrices into a supervector of twice as features as the original supervectors and classify them using SVM. It should be noted at this point that although we are doubling the amount of features, coefficients of matrices $\mathbf{U}$ and $\mathbf{V}$ are very correlated for almost symmetric square matrices as is the case of CMLLR transform matrices. For a perfect symmetric matrix, $\mathbf{U}$ and $\mathbf{V}$ collapse into the same matrix[13].

To carry out such an approach two sources of ambiguity must be addressed in some form so that supervectors are comparable within and between speakers. First, SVD algorithms produce arbitrary signs for singular vector pairs $\mathbf{u_i}$ and $\mathbf{v_i}$. This kind of ambiguity that cannot be resolved by the algorithm itself but it can be easily resolved by switching the sign of singular vectors to follow a certain criterion. For instance, we can flip the sign of $\mathbf{u_i}$ and $\mathbf{v_i}$ so that the projection of $\mathbf{u_i}$ against a fixed reference vector, e.g. $\mathbf{1} = (1, \ldots, 1)^T$, is positive. A more sophisticate approach chooses the sign of singular vectors so that their projection against the original matrix $\mathbf{A}$ is maximized [Bro *et al.*, 2008]. Once sign ambiguity is resolved, we inquiry into a more intrincate ambiguity problem related to singular vector permutation. As mentioned in Section 5.6.1, singular vectors are arranged so that their corresponding singular values are in decreasing order. While this constraint resolves ambiguity for a single matrix decomposition, always resulting in the exact factorization even when different algorithms are used, note that the order in which singular vectors are arranged depends on the matrix being factored itself. Since there is no fixed ordering criterion beyond the context of the matrix being factored, singular vectors are bound to result in rather arbitrary ordering. Therefore, comparison of factorizations for two different CMLLR transforms, say from two different speakers, is not reliable. Experimentation further supported this idea, obtaining verification error rates close to 50%, i.e. almost random results.

5.6.2.2 Singular Value Transformation

Although characterization of CMLLR transform matrices using $\mathbf{U}$ and $\mathbf{V}$ factors as features is not feasible, at least in the form addressed above, SVD of matrix $\mathbf{A}$ still gives insight into

[11] Note that $\mathbf{u_i}\mathbf{v_i}^T = (-\mathbf{u_i})(-\mathbf{v_i})^T$.

[12] $\mathbf{USV^T}$ can be rewritten as the series $\sum_{i=1}^{D} \sigma_i \mathbf{u_i}\mathbf{v_i}^T$. It is clear that the order in which i is swept in the sum can be permuted arbitrarily.

[13] Note that symmetric matrices allow the eigenvalue decomposition $\mathbf{A}\mathbf{A}^T = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ or $\mathbf{A}^T\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, resulting in the same left and right singular vectors, which correspond to the eigenvectors.

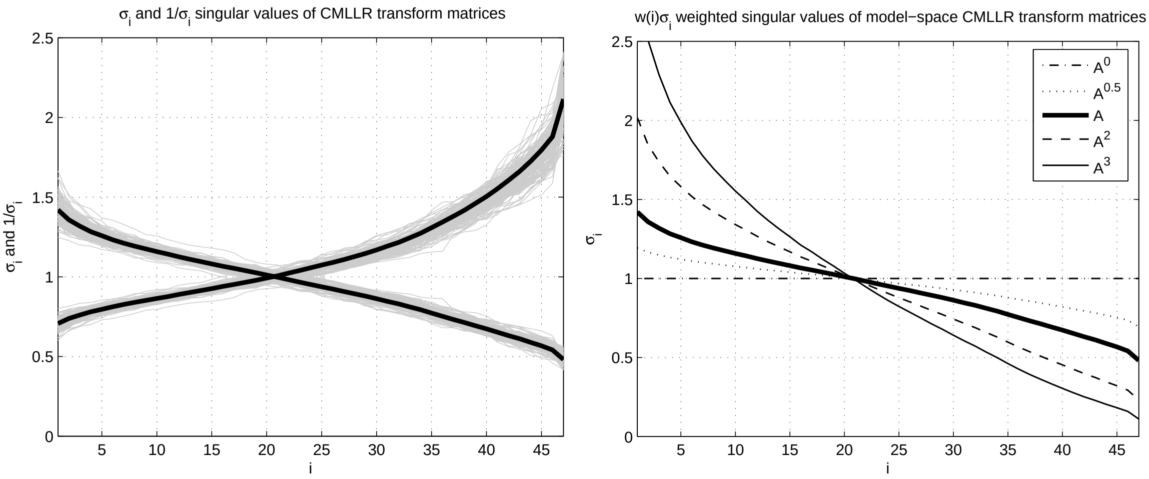

Figure 5.11 — (left,a): Analysis of singular values of CMLLR transform matrices for a random set of 150 speakers in the SRE 2005 training corpus. Decreasing curves correspond to model-space CMLLR transform matrices while increasing curves correspond to feature-space transforms. Grayed curves are per-segment curves. Thick black lines are mean singular values across speaker segments. (right, b): Weighted mean singular values of model-space CMLLR transform matrices across the same set of 150 speakers. We take powers 0, 0.5, 1, 2 and 3 of the singular values of matrix $\mathbf{A}$ corresponding to exponential weighting.

the feature-space and model-space CMLLR transforms. Using the SVD resynthesis ability, we can still modify $\mathbf{A}$ based on its SVD factorization, now working in the same supervector space as the original matrix, i.e. coefficients relating cepstral features, thus resulting in comparable matrices across speakers. Figure 5.11(a) shows a plot of singular values of $\mathbf{A}$ and $\mathbf{A^{-1}}$ for a random set of about 150 speakers in the SRE 2005 training corpus. Curves have been processed so that every singular value weights the same singular vectors, resulting in decreasing curves for $\mathbf{S}$ and increasing curves for $\mathbf{S^{-1}}$. Gray curves correspond to individual speakers and thick black lines correspond to average singular values across speakers. Note that for both groups of curves variability across speakers is very low, i.e. all curves for 150 speakers gathered around mean singular values. This implies that the speaker information must be in the singular vectors, natural basis for $\mathbf{A}$ and $\mathbf{A^{-1}}$. Also note that mean curves cross at $i \approx 21$ for which singular values and their reciprocal coincide, i.e. $\sigma_i = 1/\sigma_i = 1$. Their behavior starts to diverge around this point for model-space and feature-space transform matrices. Given that large singular values emphasize the corresponding singular vectors in the SVD-synthesized matrix, the region with larger singular values for model-space transforms compared to feature-space transforms is partly responsible for the classification performance improvement observed in the experiments of Section 5.4.3. Conversely, those singular values larger for feature-space transform matrices are responsible for performance loss. In this way, we can think of a continuum of alternative representations for matrix $\mathbf{A}$ by transforming its singular values. Since singular values are typically assumed to decrease exponentially it is reasonable to use an exponential weighting function so that the constant decay rate is preserved[14]. Such an function should evaluate to 1 for the point $(i = 21, \sigma_i = 1)$ then

[14] The product of two exponential functions is another exponential function.

$$w(i) = \mathrm{e}^{-\alpha(i-21)} \tag{5.16}$$

becoming a good candidate. Here, α is a parameter than controls the slope of new singular values and needs to be tuned. However, the same exponential effect can be obtained by directly taking a positive, possibly fractional, power of the singular values of $\mathbf{A}$. This does not allow to specify change of singular values in relative terms directly but it avoids calibration of the reference point $(i, \sigma_i = 1)$. Figure 5.11(b) shows singular value curves obtained by transformation of the singular values of $\mathbf{A}$ using powers 0, 0.5, 1, 2 and 3 functions. Note that power 0 results in the same forward and inverse matrix[15], power -1 in the transposed inverse matrix and positive powers move singular values in the direction of those of of matrix $\mathbf{A}$. Based on the results that motivated this transformation, we have no a priori knowledge about the optimal power value thus requiring to be explored experimentally.

Applying other functions onto the singular values leads to a relatively large family of transformations of matrix $\mathbf{A}$. In the same way as using the reciprocal of the singular values leads to $\mathbf{A^{-1}}$ and taking power $x \in \mathcal{R}$ to $\mathbf{A^{x}}$, logarithm and exponential functions result in log$\mathbf{A}$ and e$^{\mathbf{A}}$. These matrices can alternatively be obtained by Taylor power series expansion of the involved functions. A truncated SVD expansion can also be used, easily done by using the largest singular values only. However, note in Figure 5.11(a) that the singular value spread σ_1/σ_{47} is quite small, around 3, suggesting that the right-most basis in the graph also contribute significatively to system performance. In fact, we can think of the singular vectors corresponding to the first and last singular values as the best rank-1 approximations of matrices $\mathbf{A}$ and $\mathbf{A^{-1}}$, i.e. model-space and feature-space adaptation respectively. Under this assumption, left-most singular values of $\mathbf{A}$ and right-most singular values of $\mathbf{A^{-1}}$ should be preserved or emphasized. In this sense, a compound function which takes a positive power of model-space singular values and a negative power of feature-space singular values can be used. In this case, singular value transformation can be easily implemented as

$$\sigma'_i = \begin{cases} \sigma_i^x & \text{if } \sigma_i \geq 1 \\ \sigma_i^{-x} & \text{if } \sigma_i < 1 \end{cases} \tag{5.17}$$

where σ'_i and σ_i are the i-th transformed and original singular values of $\mathbf{A}$. Note that when $x = 1$ this function reduces to taking the maximum of the i-th singular values of $\mathbf{A}$ and $\mathbf{A^{-1}}$. We call the resulting matrix, $\mathbf{A^{x}_{max}}$ for this reason. We assume the presented power functions are flexible enough to explore the possibilities of singular value transformation.

5.6.2.3 Symmetric and Skew-Symmetric Decomposition

A third alternative for representing CMLLR transform coefficients is founded on the observation that $\mathbf{U}$ and $\mathbf{V}$ matrices are related to symmetry properties of $\mathbf{A}$. In fact, SVD requires left and right singular vectors so that non-symmetric matrices can be properly factorized. Both of these matrices collapse into a single eigenvector matrix when $\mathbf{A}$ is perfectly symmetric. Given that CMLLR adaptation of cepstra results in close to diagonal transform matrices, it may be convenient to decompose them into symmetric and skew-symmetric components, the latter behaving as a sort of residual matrix. Assuming that symmetric and skew-symmetric matrices, $\mathbf{A}_+$ and $\mathbf{A}_-$ respectively, combine additively to obtain the

[15] This is also an orthogonal matrix, as $\mathbf{A^{0}}^T = \mathbf{A^{0}}^{-1}$.

original matrix, they can be found as

$$\mathbf{A}_{+} = \frac{A + A^T}{2} \quad \text{and} \quad \mathbf{A}_{-} = \frac{A - A^T}{2} \tag{5.18}$$

Such a decomposition can be obtained for any square non-symmetric matrix $\mathbf{A}$ regardless of its structure. The skew-symmetric matrix $\mathbf{A}_{-}$ satisfies $\mathbf{A}_{-}{}^{T} = -\mathbf{A}_{-}$, has zero diagonal elements and, thus, null trace too. Also note that for any real $\mathbf{x}$ any quadratic form $\mathbf{x^T}\mathbf{A}_{-}\mathbf{x} = 0$ since $\mathbf{A}_{-}$ has imaginary eigenvalues and eigenvectors only. In fact, all real eigenvalues are in $\mathbf{A}_{+}$, a matrix with a non-null trace.

Based on the decomposition of Equation 5.18 we can represent matrix $\mathbf{A}$ using matrices $\mathbf{A}_{+}$ and $\mathbf{A}_{-}$. For C-dimensional cepstral vectors, the number of different elements for matrix $\mathbf{A}_{-}$ is $C(C-1)/2$ and $C(C+1)/2$ for $\mathbf{A}_{+}$ thus summing up to a total of C^2 different values, the same number involved in $\mathbf{A}$[16]. The coefficients of one of both matrices alone can be also used.

5.6.3 Experimental Results

In this section we explore two kinds of representations of CMLLR transform coefficients described in Section 5.6.2 for CMLLR-SVM systems using NAP session compensation. As a first alternative, we resynthesize model-space CMLLR transform matrices using a power of their singular values, eventually taking a positive power of the maximum of the singular values of model-space and feature-space transforms. Systems based on the latter use the notation CG15 $\mathbf{A^x_{max}}, \mathbf{b}$ where x is the power used while the former use simply CG15 $\mathbf{A^x}, \mathbf{b}$. We explored integer powers only in this set of experiments. The second alternative explored is decomposing matrix $\mathbf{A}$ into symmetric and skew-symmetric matrices, $\mathbf{A}_{+}$ and $\mathbf{A}_{-}$ respectively. We test three systems setups using $\mathbf{A}_{+}$, $\mathbf{A}_{-}$ or all non-redundant $\mathbf{A}_{+}, \mathbf{A}_{-}$ coefficients. Regarding CMLLR-SVM setup, we keep SAT and NAP parameters fixed this time, as we focus on exploration of representations only. We use 2 SAT iterations for UBM training and 50 dimensions for the session subspace as it resulted in the best overall performance observed in previous experiments.

We focus first on the upper block of Table 5.4 to observe the effect of using powers other than 1 for $\mathbf{A}$. A large variation of performance is observed with respect to system CG15 $\mathbf{A^1}, \mathbf{b}$, using model-space transform coefficients, which we set as reference system. The first system, using feature-space transform matrix coefficients $\mathbf{A}^{-\mathbf{1}}, \mathbf{b}$ [17], results in an important performance drop, about 10% in relative terms averaging MDC and EER for both corpora, but getting over 15% MDC for SRE 2005. SRE 2006 exhibits smaller performance loss, in the same line as observed in Section 5.4.3 for systems not using NAP. Using uniform weighting of all singular vectors, i.e. system CG15 $\mathbf{A^0}, \mathbf{b}$, obtains the worst performance amongst all experiments, eventually 20% below CG15 $\mathbf{A^1}, \mathbf{b}$. This verifies that certain singular vectors in the SVD decomposition of $\mathbf{A}$ have a strong effect on performance, namely first and last singular vectors, which correspond to major components responsible for model-space and feature-space adaptation respectively. On the other side, using $\mathbf{A^2}, \mathbf{b}$ slightly improves with respect to the reference system. We observe relative gains of nearly 4% MDC relative gain for SRE 2005 and 2% for SRE 2006 but none for EER. Powers larger than 2 seem

[16] We actually use the upper triangular part of $\mathbf{A}$, including the diagonal, to store $C(C+1)/2$ coefficients of $\mathbf{A}_{+}$ and the lower triangular part to store $C(C-1)/2$ coefficients of $\mathbf{A}_{-}$.

[17] Note that offset vector $\mathbf{b}$ is used instead of $-\mathbf{A}^{-\mathbf{1}}\mathbf{b}$, the latter corresponding to feature-space transforms. We have observed a minimal difference in performance by such replacement. This makes the first block of results more comparable, since only the matrix part of the transform coefficients changes from system to system.

		SRE 2005				**SRE 2006**			
		MDC		EER (%)		MDC		EER (%)	
System	Coeff.	F	FB	F	FB	F	FB	F	FB
CG15	$\mathbf{A^{-1}}, \mathbf{b}$	.0330	.0297	7.53	7.39	.0267	.0245	5.88	5.80
CG15	$\mathbf{A^{0}}, \mathbf{b}$	.0341	.0320	8.36	8.15	.0295	.0275	6.34	6.16
CG15	$\mathbf{A^{1}}, \mathbf{b}$	.0285	.0256	6.74	6.70	.0247	.0234	5.65	5.38
CG15	$\mathbf{A^{2}}, \mathbf{b}$	.0275	.0253	6.99	6.87	.0245	.0228	5.74	**5.37**
CG15	$\mathbf{A^{3}}, \mathbf{b}$	.0304	.0283	7.53	7.48	.0278	.0255	6.53	6.12
CG15	$\mathbf{A^{1}_{max}}, \mathbf{b}$	.0310	.0292	7.24	7.28	.0262	.0242	5.88	5.79
CG15	$\mathbf{A^{2}_{max}}, \mathbf{b}$	.0336	.0306	7.78	7.91	.0276	.0248	6.33	6.25
CG15	$\mathbf{A_{+}}, \mathbf{b}$	.0346	.0321	8.48	8.44	.0297	.0272	7.17	6.79
CG15	$\mathbf{A_{-}}, \mathbf{b}$	.0363	.0330	8.31	8.48	.0301	.0281	6.89	6.76
CG15	$\mathbf{A_{+}}, \mathbf{A_{-}}, \mathbf{b}$	**.0266**	**.0242**	**6.45**	**6.28**	**.0237**	**.0220**	**5.37**	5.48

Table 5.4 — MDC and EER of CMLLR-SVM systems using 2 SAT iterations and 50 dimensions for the NAP session subspace. The first block of systems uses matrix coefficients obtained by taking powers -1, 0, 0.5, 1, 2 and 3 of the singular values of $\mathbf{A}$. The second block uses matrix coefficients obtained by taking the first 45 and 46 out of 47 singular values of the SVD decomposition of $\mathbf{A}$ only. The third block uses the concatenation of coefficients of symmetric and skew-symmetric matrices $\mathbf{A}_{+}$ and $\mathbf{A}_{-}$. All of the systems use the offset vector corresponding to model-space transforms, i.e. $\mathbf{b}$, as well. Columns F and FB show forward and averaged forward-backward scores respectively. The best scores in each column are shown in boldface.

to result in further performance loss. Systems at the end of the first block of Table 5.4 use powers 1 and 2 of the maximum of model-space and feature-space singular values, as in Equation 5.17. These systems are clearly less performing than CG15 $\mathbf{A^{1}}, \mathbf{b}$, although CG15 $\mathbf{A^{1}_{max}}, \mathbf{b}$ is still slightly better than CG15 $\mathbf{A^{-1}}, \mathbf{b}$ using feature-space coefficients. The third block of results corresponds to the decomposition of $\mathbf{A}$ into symmetric and skew-symmetric components $\mathbf{A}_{+}$ and $\mathbf{A}_{-}$. Using forward scoring this system outperforms CG15 $\mathbf{A^{1}}, \mathbf{b}$ in 7% MDC and around 6% EER for SRE 2005 and 11% MDC for SRE 2006, although none in EER. Although this is the best performing of all systems, using either symmetric or skew-symmetric matrices alone results in the worst systems. It is also worth noting that the difference of performance from the worst to the best performing system easily goes beyond 20%, highlighting the relevance of representation issues of CMLLR transform coefficients in these kind of systems.

Forward-backward scoring brings additional improvements in general, reaching over 10% MDC although very small in EER average terms for SRE 2005, occasionally resulting in performance loss. MDC and EER gains are more balanced for SRE 2006, 6% MDC versus 4% EER in average.

The most significative systems using forward scoring are put together in the constellation plot of Figure 5.12. Relative placement of systems is rather consistent for both SRE 2005 and SRE 2006. CG15 $\mathbf{A^{0}}, \mathbf{b}$ using uniform weighting in the top-right corner, then $\mathbf{A^{1}_{max}}, \mathbf{b}$ and $\mathbf{A^{2}_{max}}$ and $\mathbf{A^{-1}}, \mathbf{b}$ using feature-space transforms, followed by CG15 $\mathbf{A^{1}}, \mathbf{b}$ and CG15 $\mathbf{A^{2}}, \mathbf{b}$ with similar overall performance and finally CG15 $\mathbf{A_{+}}, \mathbf{A_{-}}, \mathbf{b}$ in the bottom-left corner, outperforming all previous systems. Figures 5.12(a) and 5.12(b) show DET curves of some of these systems using forward scoring for SRE 2005 and SRE 2006. For SRE 2005,

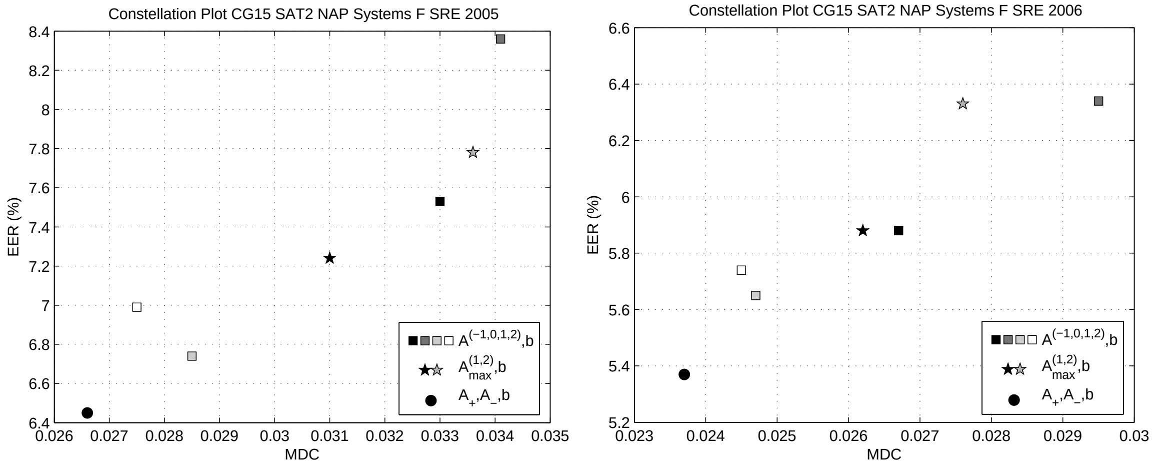

Figure 5.12 — Constellation plots of CG15 systems using forward scoring and alternative representations of CMLLR transform coefficients for SRE 2005 (left, a) and SRE 2006 (right, b). Systems using powers -1, 0, 1 and 2 of $\mathbf{A}$, powers 1 and 2 of the maximum of model-space and feature-space singular values of $\mathbf{A}$ and $\mathbf{A}_+$ and $\mathbf{A}_-$ matrices use symbols ■, $\star$ and • respectively. Gray levels correspond to different power values.

curves clearly respect the ordering observed in the corresponding constellation plot. CG15 $\mathbf{A}^2, \mathbf{b}$ results in a slight rotation of the DET curve of CG15 $\mathbf{A}^1, \mathbf{b}$, improving in MDC but not in EER and viceversa. CG15 $\mathbf{A}_+, \mathbf{A}_-, \mathbf{b}$ outperforms the rest of the systems at all operating points. For SRE 2006, curves show a similar behavior for false alarm rates below 10%, although curves differences are less clear in the graph.

5.7 Feature-level Inter-session Variability Compensation using CMLLR

We addressed NAP compensation of CMLLR transforms for CMLLR-SVM systems in Section 5.5. Experimental results seemed to confirm the improvement in robustness expected by modeling inter-session variability explicitly. In this section, we follow the discussion about session compensation of CMLLR transforms opened in Section 5.5.1 and we further develop a compensation technique working at the cepstral level, still being based on NAP compensation of CMLLR transforms.

5.7.1 CMLLR-based Session Compensation of Cepstra

The proposed approach to feature-level CMLLR session compensation uses CMLLR transforms in both supervector and affine transform domains together in a SAT framework. The core of the technique uses NAP to compensate CMLLR transform supervectors which, as opposed to CMLLR-SVM systems, are now applied at the cepstral level. The compensated cepstra can then be used in any other acoustic system in the same way feature mapping or feature-level Eigenchannel modeling already do.

We start noting that the same development in Section 5.5.1 decomposing speaker and

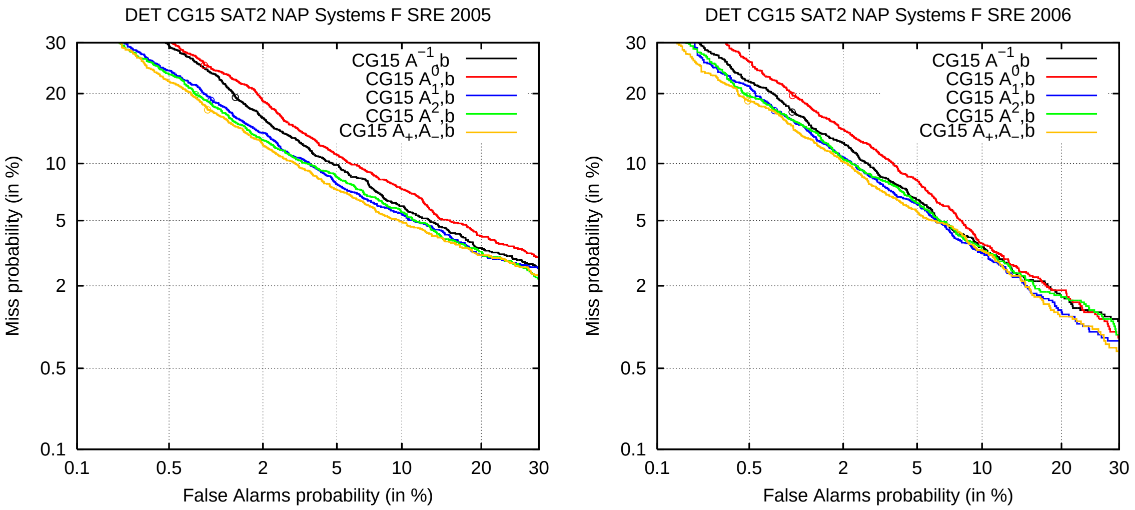

Figure 5.13 — DET curves of CG15 systems using two SAT iterations, alternative representations of CMLLR transform matrices, 50 dimensions for the NAP session subspace and forward scoring for SRE 2005 (left, a) and SRE 2006 (right, b). Only systems using powers of $\mathbf{A}$ and symmetric/skew-symmetric matrices are shown.

session components of GMM mean vectors can also be used for cepstral features, as both work in the same domain. In particular, speaker- and session-dependent cepstra $\mathbf{x_{sh}}$ can be decomposed following Equation 5.10 as

$$\begin{aligned} \mathbf{x_{sh}} &= \mathbf{x_s} + \mathbf{x_h} \\ &= \mathcal{C}_s(\mathbf{x}) + \mathcal{C}_h(\mathbf{x}) \qquad (5.19) \\ &= (\mathcal{C}_s + \mathcal{C}_h)(\mathbf{x}) \\ &= \mathcal{C}_{sh}(\mathbf{x}) \qquad (5.20) \end{aligned}$$

where $\mathbf{x_s}$ and $\mathbf{x_h}$ are speaker-only and session-only cepstral vectors and $\mathbf{x}$ a speaker- and session-independent vector. We identify the session-compensated vectors $\mathbf{x_s}$ as the left-hand side term of Equation 5.19, that is

$$\mathbf{x_s} = \mathcal{C}_s(\mathbf{x}) \tag{5.21}$$

$\mathbf{x}$ can be readily obtained[18] by inversion of Equation 5.20 as

$$\mathbf{x} = \mathcal{C}_{sh}^{-1}(\mathbf{x_{sh}}) \tag{5.22}$$

which combined with Equation 5.21 yields the estimator

$$\mathbf{x_s} = (\mathcal{C}_s \circ \mathcal{C}_{sh}^{-1})(\mathbf{x_{sh}}) \tag{5.23}$$

[18] Note that $\mathbf{x}$ can be also obtained directly from the UBM as $\mathbf{x} = \sum_{i=1}^{N} p(i|\mathbf{x_{sh}})\boldsymbol{\mu_i}$.

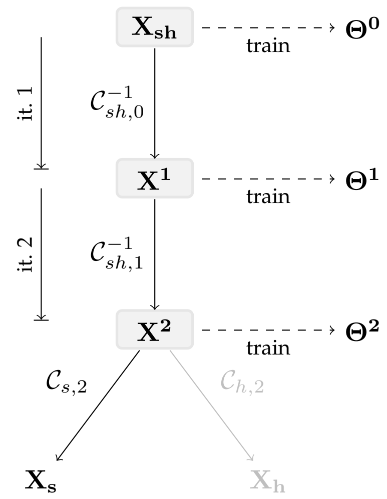
$\mathbf{X_{sh}}$
train
$\boldsymbol{\Theta}^{\mathbf{0}}$
it. 1
$\mathcal{C}_{sh,0}^{-1}$
$\mathbf{X^1}$
train
$\boldsymbol{\Theta}^{\mathbf{1}}$
it. 2
$\mathcal{C}_{sh,1}^{-1}$
$\mathbf{X^2}$
train
$\boldsymbol{\Theta}^{\mathbf{2}}$
$\mathcal{C}_{s,2}$
$\mathcal{C}_{h,2}$
$\mathbf{X_s}$
$\mathbf{X_h}$

Figure 5.14 — Diagram of cepstral-level session compensation using two SAT iterations for a speaker segment $\mathbf{X_{sh}}$.

Therefore, we first remove all speaker and session components from uncompensated cepstra $\mathbf{x_{sh}}$ by application of the feature-space CMLLR transform $\mathcal{C}_{sh}^{-1}$, obtaining a CMLLR estimate of $\mathbf{x}$. Second, we add the speaker component back to $\mathbf{x}$ by means of $\mathcal{C}_s$, the speaker component of the corresponding model-space CMLLR transform. Since both transforms are homogeneous, the composition of $\mathcal{C}_s$ and $\mathcal{C}_{sh}^{-1}$ results in a transform which is relatively close to the identity, involving a matrix $\approx \mathbf{I}$ and an offset vector $\approx \mathbf{0}$, if only a small part of CMLLR supervector variability is removed by NAP. This is observed in practice when few tens of dimensions are used for the session subspace compared to the thousands of features involved in CMLLR transforms.

All of the CMLLR transforms used are obtained using a UBM, which can be trained using SAT as well. Figure 5.14 illustrates a schema of how cepstra $\mathbf{X_{sh}} = (\mathbf{x_{sh,1}}, \ldots, \mathbf{x_{sh,T}})$ with T being the number of frames, are processed for session compensation. SAT uses transforms $\mathcal{C}_{sh}^{-1}$ in the same way as we did for the CMLLR-SVM system, yielding successive estimates of speaker- and session-independent cepstra $\mathbf{X}$, i.e. $\mathbf{X^1}$ and $\mathbf{X^2}$ for iterations one and two in the figure. The speaker component is added at the end of the processing chain, once the final UBM is ready. The session-compensated cepstra $\mathbf{X_s}$ is then obtained using the chain of transformations $\mathcal{C}_{s,2} \circ \mathcal{C}_{sh,1}^{-1} \circ \mathcal{C}_{sh,0}^{-1}$ in this case. Note that, as in standard SAT, the composition of transforms $\mathcal{C}_{sh,1}^{-1} \circ \mathcal{C}_{sh,0}^{-1}$ can be directly estimated from $\mathbf{X_{sh}}$ once $\boldsymbol{\Theta}^{\mathbf{2}}$ is available.

Speaker and session supervector components of $\mathcal{C}_{sh}$ obtained using NAP, $\mathbf{c_s}$ and $\mathbf{c_h}$ respectively, are orthogonal since they lie in complementary subspaces and hence their dot dot product evaluates to 0. Rewriting the dot product in terms of the rows of matrix $\mathbf{A}$ we obtain

$$
\begin{aligned}
\mathbf{c_s}^T\mathbf{c_h} &= \sum_{i}^{D} \mathbf{c_{s,i}c_{h,i}} \\
&= \sum_{r}^{C} \mathbf{A_{s,r}^T A_{h,r}} + \mathbf{b_s}^T\mathbf{b_h} = 0
\end{aligned}
$$

with C being is the number of rows and columns of $\mathbf{A}$, i.e. the dimension of cepstral vectors, and D is the dimension of supervectors $\mathbf{c_s}$ and $\mathbf{c_h}$. Assuming offset components not to affect orthogonality properties significatively, the dot product can be approximated as

$$
\begin{aligned}
\mathbf{c_s}^T\mathbf{c_h} &\approx \sum_{r}^{C} \mathbf{A_{s,r}^T A_{h,r}} \\
&\approx tr\left(\mathbf{A_s^T A_h}\right) \approx 0 \qquad (5.24)
\end{aligned}
$$

$$(5.25)$$

Note, however, that although Equation 5.24 is satisfied for transforms $\mathbf{A_s}$ and $\mathbf{A_h}$, nothing can be said about orthogonality of the corresponding speaker- and session-compensated cepstra

$$
\begin{aligned}
\left(\mathbf{A_s x}\right)^T \left(\mathbf{A_h x}\right) &= \mathbf{x}^T\mathbf{A_s}^T\mathbf{A_h x} \\
&= tr\left(\mathbf{x}^T\mathbf{A_s}^T\mathbf{A_h x}\right) \qquad (5.26)
\end{aligned}
$$

5.7.2 CMLLR-based Session Compensation of UBM

In the technique presented in the previous section, a speaker segment and a UBM are required to compute $\mathcal{C}_{sh}^{-1}$. We can use any UBM, trained using standard maximum likelihood estimation or using a SAT approach. We have also observed in Section 5.5.2 that the use of SAT for UBM training results in performance improvements of CMLLR-SVM systems, regardless of whether NAP is used or not. This can be explained by the fact that transform coefficients capture stronger speaker components if a more speaker-independent UBM is used. These systems removed session variability of uncompensated CMLLR transform supervectors, hence NAP being applied out of the SAT loop once the UBM is re-estimated. However, such an approach uses uncompensated transforms to re-train the UBM. In this section we discuss an approach to jointly compensate of UBM and CMLLR transforms in a SAT scheme.

The key idea is to include NAP in the SAT re-training loop of the UBM. Instead of using speaker- and session-compensated CMLLR transforms $\mathcal{C}_{sh}^{-1}$ to obtain cepstra used to retrain the UBM we can avoid speaker components only by using $\mathcal{C}_h \circ \mathcal{C}_{sh}^{-1}$[19]. In this way, we are actively targeting speaker components in final CMLLR transforms resulting from these speaker-compensated UBM. This approach is similar to a per-speaker SAT approach

[19] Note that $\mathcal{C}_h \circ \mathcal{C}_{sh}^{-1} = (\mathcal{C}_{sh} - \mathcal{C}_s) \circ \mathcal{C}_{sh}^{-1} = \mathcal{C}_{\mathbf{I,0}} - \mathcal{C}_s \circ \mathcal{C}_{sh}^{-1} \neq \mathcal{C}_s^{-1}$ in general. $\mathcal{C}_s^{-1}$ could also be obtained by NAP processing under the additive model $\tilde{\mathcal{C}}_{sh}^{-1} = \tilde{\mathcal{C}}_s^{-1} + \tilde{\mathcal{C}}_h^{-1}$ instead of $\mathcal{C}_{sh} = \mathcal{C}_s + \mathcal{C}_h$.

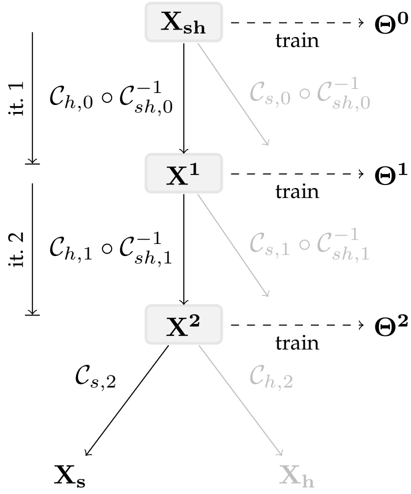
$\mathbf{X_{sh}}$
train
$\mathbf{\Theta^0}$
it. 1
$\mathcal{C}_{h,0} \circ \mathcal{C}^{-1}_{sh,0}$
$\mathcal{C}_{s,0} \circ \mathcal{C}^{-1}_{sh,0}$
$\mathbf{X^1}$
train
$\mathbf{\Theta^1}$
it. 2
$\mathcal{C}_{h,1} \circ \mathcal{C}^{-1}_{sh,1}$
$\mathcal{C}_{s,1} \circ \mathcal{C}^{-1}_{sh,1}$
$\mathbf{X^2}$
train
$\mathbf{\Theta^2}$
$\mathcal{C}_{s,2}$
$\mathcal{C}_{h,2}$
$\mathbf{X_s}$
$\mathbf{X_h}$

Figure 5.15 — Diagram of cepstral-level session compensation with two iterations of session-compensated SAT for a speaker segment $\mathbf{X_{sh}}$. Note that speaker components are not used to retrain models Θ^1 and Θ^2.

in which CMLLR transforms used for SAT are computed using speech from all sessions available for the speaker, with the difference that we use NAP to focus on speaker components of CMLLR transforms instead. To gain insight into how both approaches compare, we provide in Appendix A a proof that CMLLR transforms obtained using the concatenation of cepstra of multiple sessions for a speaker is equivalent to the weighted average of the CMLLR transforms obtained for each of the sessions individually. Under the constraints of NIST SRE campaigns this can be considered a uniform average. In this line, estimation of the inter-session covariance matrix of CMLLR transform supervectors in NAP indirectly involves the average of the CMLLR transform supervectors across sessions, the latter being equivalent to the CMLLR transform supervector obtained using concatenation of cepstra of these sessions. Thus, CMLLR transforms computed multiple sessions are inherently involved in NAP training as well.

The compensated UBM can be used in the session compensation estimator of the previous section or in any other system using an UBM. Figure 5.15 shows a diagram with the steps involved for compensating cepstra $\mathbf{X_{sh}}$ using two SAT iterations. At each iteration the speaker components of CMLLR transforms are removed in the CMLLR transform supervector space. The final estimate $\mathbf{X_s}$ is obtained as $(\mathcal{C}_{s,2} \circ \mathcal{C}_{h,1} \circ \mathcal{C}^{-1}_{sh,1} \circ \mathcal{C}_{h,0} \circ \mathcal{C}^{-1}_{sh,0})(\mathbf{X_{sh}})$. The composition of CMLLR transforms $\mathcal{C}_{h,1} \circ \mathcal{C}^{-1}_{sh,1} \circ \mathcal{C}_{h,0} \circ \mathcal{C}^{-1}_{sh,0}$ can still be obtained in a single adaptation step if model $\mathbf{\Theta^2}$ is available.

This session compensated SAT approach is possible because orthogonality in the CMLLR supervector space is not propagated down to the cepstral space, as shown at the end of the previous section. NAP actually uses new estimates of the inter-session covariance matrix at each SAT iteration because the CMLLR transforms used for this purpose are estimated from compensated cepstra, with correlated although complementary speaker and session components. Note that $\mathbf{c_s}$ and $\mathbf{c_h}$ in Equation 5.24 are supervectors resulting from application of $\mathbf{I} - \mathbf{E}\mathbf{E}^T$ and $\mathbf{E}\mathbf{E}^T$ low-rank transforms although matrices $\mathbf{A_s}$ and $\mathbf{A_h}$ for CMLLR transforms $\mathcal{C}_s$ and $\mathcal{C}_h$ are full-rank.

				SRE 2005				**SRE 2006**			
				MDC		EER (%)		MDC		EER (%)	
Method	SAT	$D_{\text{NAP}}^{\text{CMLLR}}$	D_{NAP}	F	FB	F	FB	F	FB	F	FB
No comp.	×	×	×	.0297	.0273	6.81	6.65	.0274	.0259	6.17	6.02
FM	×	×	×	.0279	**.0254**	**6.41**	**6.12**	.0270	**.0251**	**5.83**	**5.65**
CMLLR	×	25	×	.0285	.0269	6.74	6.58	.0268	.0260	6.20	6.11
CMLLR	×	50	×	**.0270**	.0263	6.86	6.49	**.0265**	.0254	6.34	6.16
CMLLR	×	75	×	.0330	.0314	8.40	8.19	.0309	.0290	7.48	7.03
CMLLR	1	25	×	.0289	.0269	6.74	6.53	.0294	.0287	7.90	7.86
CMLLR	1	50	×	.0296	.0279	7.53	7.28	.0313	.0293	8.31	8.41
CMLLR	1	75	×	.0347	.0335	9.52	8.98	.0342	.0329	9.78	9.19
No comp.	×	×	50	.0220	**.0203**	4.99	4.83	.0196	**.0185**	**4.23**	4.23
FM	×	×	50	.0211	.0204	**4.82**	**4.48**	.0198	.0189	4.41	**4.14**
CMLLR	×	25	50	.0214	.0205	**4.82**	4.74	**.0195**	.0186	4.50	4.45
CMLLR	×	50	50	**.0209**	**.0203**	4.99	4.95	.0202	.0193	4.32	4.36

Table 5.5 — MDC and EER of PLP-SVM systems using feature-level session compensation for SRE 2005 and SRE 2006. Systems using no compensation and feature mapping are shown in the first block of results. Systems based on NAP processing of CMLLR transforms using standard and SAT UBM are shown in the second block. The third block shows systems using compensated cepstra together with NAP compensation of polynomial feature supervectors. Note that FM × × 50 is the same system appearing as PLP-SVM in Table 4.1. Columns F and FB show forward and averaged forward-backward scores respectively. The best scores for systems using and not using NAP are shown in boldface.

5.7.3 Experimental Results

The following experiments exploring CMLLR-based session compensation of cepstra use PLP-SVM systems based on a GLDS kernel and SVM classification. This is the same system described in Section 4.1.4 but differing in the type of channel or session compensation applied to cepstra. It uses supervectors involving polynomial features built directly from cepstra, i.e. undergoing no modeling, which we expect to help make the effect of cepstra compensation more visible.

The baseline PLP-SVM system uses no channel or session compensation, i.e. no feature mapping or NAP applied to polynomial feature supervectors, while PLP-SVM FM uses feature mapping only. The other systems replace feature mapping with CMLLR-based session compensation of cepstra, described in Section 5.7.1, using either standard maximum likelihood or speaker adaptive training of the UBM. We include offset vectors $\mathbf{b}$ in the CMLLR transform supervectors and we use a several choices of session subspace dimensions. All PLP-SVM systems use the same speech/non-speech segmentation[20] and feature warping is applied at the end of the cepstra processing chain. We also provide results of the best performing systems using NAP compensation of polynomial supervectors.

[20] Our PLP-SVM systems are quite sensitive to proper speech activity detection, possibly due to poor modeling of cepstral features before classification. As a consequence of non-linear amplification of polynomial expansions, features issued from non-speech regions tagged as speech can disturb features issued from speech regions.

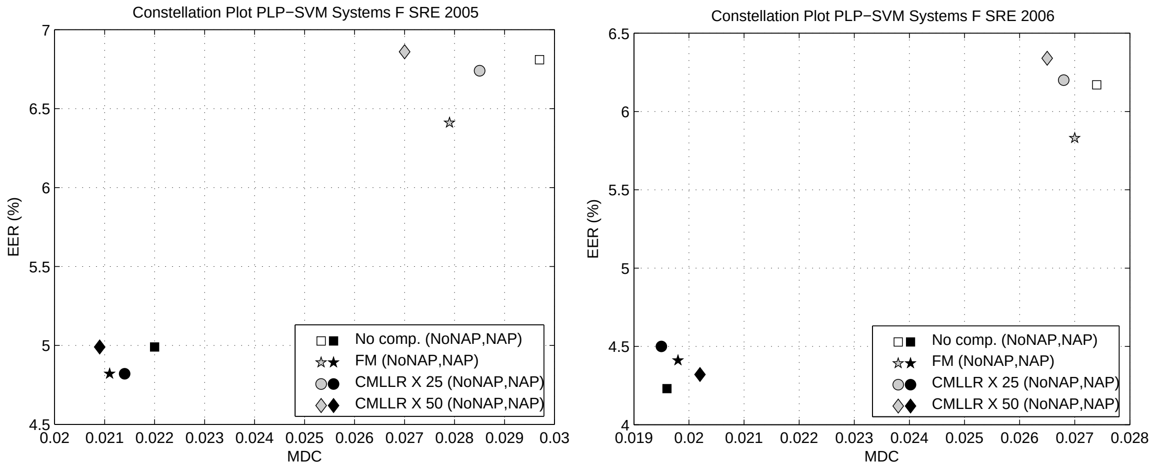

Figure 5.16 — Constellation plots of forward-scoring PLP-SVM systems using different session compensation algorithms for SRE 2005 (left, a) and SRE 2006 (right, b). Symbol shapes indicate the compensation method used on cepstra and gray levels the number of cascaded compensation algorithms used: white for no compensation, gray for FM or CMLLR and black for systems using NAP to compensate polynomial feature supervectors.

Table 5.5 shows MDC and EER results for all systems. Systems with no compensation and feature mapping only are found in the first block. PLP-SVM FM obtains systematic improvements over PLP-SVM over 6% MDC and EER for SRE 2005. Gains are slightly lower for SRE 2006, specially in MDC. The core condition of SRE 2006 includes additional channel conditions not observed in SRE 2005 or in the data used to train feature mapping models. Therefore, it is difficult for feature mapping to compensate for these new conditions.

The second block shows systems using CMLLR-based compensation with a standard UBM and a UBM trained using SAT as a function of the number of session subspace dimensions. These systems obtain improvements over PLP-SVM of about 9% and 3% MDC for SRE 2005 and SRE 2006 respectively, although degrading EER overall. Using 25 and 50 dimensions for the session subspace results in the best EER and MDC measures respectively. This behavior suggests that CMLLR systems also result in a slightly rotation of the DET curve with respect to PLP-SVM, visible in Figures 5.17(a) and 5.17(a). We can also observe that feature mapping outperforms CMLLR globally except for operating points in the MDC area, where a small improvement is obtained for CMLLR $\times$ 50 compared to FM. These results are highlighted in boldface style in Table 5.5, as CMLLR $\times$ 50 is the best performing system in MDC for forward-scoring systems. The optimal number of subspace dimensions is probably between 25 and 50. Using more dimensions clearly results in a too aggressive algorithm removing important speaker components. These systems are shown as white and gray symbols in the constellation plots of Figures 5.16(a) and 5.16(b). For SRE 2005, FM and CMLLR X 25 exhibit similar performance, although FM obtains more balanced MDC and EER improvements. For SRE 2006, CMLLR-based compensation obtains smaller gains than FM. Systems using CMLLR-based compensation with one SAT iteration of UBM training follow the same trend as systems using a regular UBM. However, the optimal subspace dimension seems to be lower, as performance loss is already visible for CMLLR 1 50, specially in EER. This can be explained by the fact that CMLLR transforms obtained using a SAT UBM capture stronger speaker and session components, requiring less

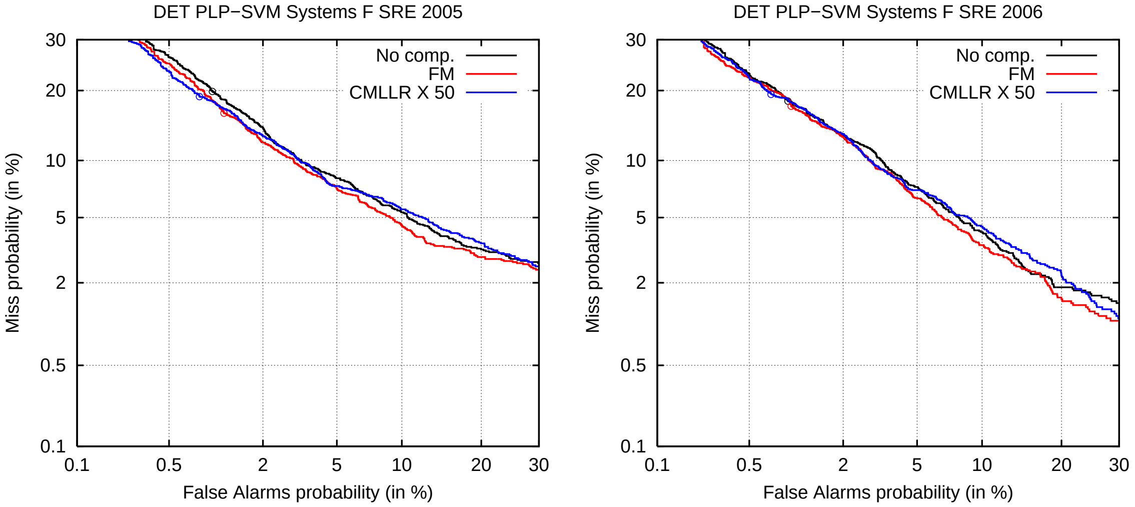

Figure 5.17 — DET curves of PLP-SVM systems using different techniques of session compensation of cepstra for SRE 2005 (left, a) and SRE 2006 (right, b). No compensation, feature mapping and CMLLR-based using a regular UBM and 50 dimensions for the session subspace are shown.

dimensions to capture the same amount of session variability. In any case, these systems never outperform feature mapping. Forward-backward scoring obtains improvements for almost all systems in Table 5.5, about 4% in average although they get slightly lower for systems using CMLLR-based compensation.

Important performance improvements are observed when using NAP compensation of polynomial feature supervectors. Systems in the third block of results in Table 5.5 obtain gains between 20% and 30% in both MDC and EER terms. This is visible in the constellation plots of Figures 5.16(a) and 5.16(b), where we can clearly distinguish two clusters of systems, one using NAP on the bottom-left corner and one not using NAP on the top-right corner. In such a scenario, gains due to cepstra compensation have reduced considerably. For SRE 2005, feature mapping and CMLLR are slightly outperforming PLP-SVM × × 50, the former preferring EER and the latter MDC. For SRE 2006, NAP alone does the best job and the rest of the systems perform similarly. If we look at the corresponding DET curves of Figures 5.18(a) and 5.18(b) we observe that feature mapping outperforms the other systems for most of the operating points for SRE 2005, although CMLLR takes over in the low false-alarm probability region beyond MDC. However, CMLLR becomes the less performing system in the rest of the curve, even below PLP-SVM X X 50. DET curves of these systems are much closer for SRE 2006, where feature mapping and non-compensated systems exhibit the same performance and CMLLR is consistently worse than them.

The previous results suggest that cepstral-level session compensation is not effective when combined with NAP working at the supervector level. This is not surprising since both techniques are aimed at the same goal, although in different domains. When NAP is not used at the supervector level, feature mapping is more efficient than CMLLR-based compensation. For SRE 2006, involving more session variability than SRE 2005, CMLLR fails to perform successful compensation while feature mapping still obtains a reasonable improvement. We argue that feature mapping performs channel compensation for each mean vector in the UBM independently, while our CMLLR-based technique uses a single affine compensation transform tied across all mean vectors in the model. We believe the

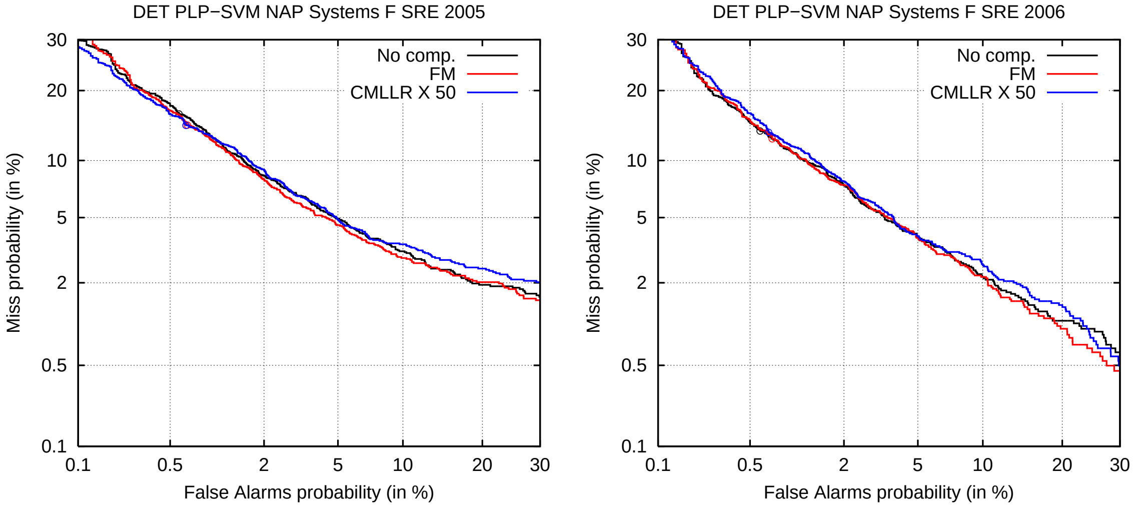

Figure 5.18 — DET curves of PLP-SVM systems using different techniques of session compensation of cepstra together with NAP compensation of polynomial feature supervectors for SRE 2005 (left, a) and SRE 2006 (right, b). No compensation, feature mapping and CMLLR-based using a regular UBM and 50 dimensions for the session subspace are shown. This dimension is also set to 50 for NAP compensation of polynomial feature supervectors.

limiting factor here is Gaussian tying rather than NAP compensation of CMLLR transform supervectors, as we saw in Section 5.5.1 that NAP compensation of GMM supervectors results in the Eigenchannel approach, which has proven to be effective. A simple way of improving the CMLLR-based compensation method could be estimating multiple CMLLR compensation transforms, the supervectors of which are jointly compensated using NAP. In this case, each of the CMLLR transforms must be carefully applied on the corresponding cepstra regions, resulting in sudden changes of the compensation transform in the time domain.

5.8 Summary and Conclusion

This chapter presented an alternative to the standard MLLR-SVM approach using CMLLR transform coefficients and speaker adaptive training. Being based on a GMM-UBM, this approach avoids the use of transcripts and is language-independent, in the sense that it can be used for any language without any structural change. Base CMLLR-SVM systems obtain reasonable performance which depends on two major factors: the choice of either model-space or feature-space CMLLR transform coefficients and the number of SAT iterations. Model-space coefficients significantly outperform feature-space counterparts, while performance for system using their concatenation lies in between. The number of SAT iterations is a second source of improvement, with performance increasing as more SAT iterations are performed, although at a progressively slower pace.

NAP compensation of CMLLR transforms and Eigenchannel modeling are two comparable approaches fitting a similar utterance variability model. CMLLR-SVM systems using NAP compensated CMLLR transform supervectors considerably outperform base CMLLR-

SVM systems. Gains are more visible in the SRE 2006 corpus, probably due to the presence of more channel variability in its core condition. Session subspace dimensions between 25 and 75 result in similar performance, which is overall optimal for 50 dimensions.

We have gained insight into several representation alternatives based on SVD decomposition of CMLLR transform matrices. Under this model, speaker-specific information is encoded in the singular vectors while singular values clearly exhibit almost no inter-speaker variance. Model-space and feature-space CMLLR transform matrices use the same singular vectors but reciprocal singular values. The most relevant model-space and feature-space components are gather each end of the singular value spectra repectively. We proposed transforming CMLLR transform matrices by operating on the singular values. Taking a power of singular values resulted in about the same performance obtained for model-space transforms. Decomposing model-space transform matrices into symmetric and skew-symmetric components resulted in considerable improvements over using model-space transforms. Symmetric and skew-symmetric matrices seem to be tightly coupled to each other, requiring both of them for good performance.

Based on the SAT framework used in the CMLLR-SVM approach we developed a technique that performs feature-level inter-session variability compensation. NAP-compensated CMLLR supervectors are converted back to matrix form and applied on cepstra to separate speaker-only from session-only components. Session compensation of cepstra showed rather modest improvements in performance, occasionally outperforming feature mapping, when using a PLP-SVM system based on the GLDS kernel and SVM classification. Feature mapping is more effective and stable across corpora. Rather marginal improvements are obtained by cepstral-level session compensation when used together with NAP compensation of polynomial feature supervectors.

Chapter 6

Multi-class (C)MLLR-SVM Systems

In the preceding chapter we have mainly focused on the use of mean-variance constrained MLLR adaptation in CMLLR-SVM systems and, particularly, on model-space and feature-space representations of the adaptation process. CMLLR allows adaptation of model parameters or cepstra indistinctively, although at the price of using a single transform only. Some issues, further discussed in this chapter, arise when trying to extend the UBM-based CMLLR-SVM system to multiple classes. However, it remains an interesting alternative to other MLLR-SVM approaches.

MLLR-SVM systems in the literature typically use the acoustic models of a HMM-based LVCSR system to base MLLR computation on. They usually require an orthographic transcription of the speech segment under adaptation, used to obtain a more precise alignment of speech against the phonemic acoustic models. By clustering tri-phone models, the acoustic space can be split into multiple regions, becoming straightforward to compute multiple MLLR transforms for each speech segment. This chapter presents a comprehensive experimental study of such systems, paying special attention to front-end normalization, type of model used, either GMM or HMM, as well as training scheme used, either maximum-likelihood estimation or speaker adaptive training.

Section 6.1 discusses the issue of estimating multiple (C)MLLR transforms using UBM and LVCSR acoustic models to eventually justify switching to the latter. All (C)MLLR-SVM, using either CMLLR or MLLR transform coefficients, system setups involved in the experimental study are described in Section 6.2 and we provide experimental results for both NIST SRE 2005 and SRE 2006 corpora. In Section 6.3 we focus on the structural differences of CMLLR and MLLR transform matrices to further support the obtained experimental results. Section 6.4 deals with the use of CMLLR and MLLR transforms together in a single (C)MLLR-SVM system. We compare and combine the explored (C)MLLR-SVM approach with other state-of-the-art acoustic systems in Section 6.5. A brief summary and conclusions are given in Section 6.6.

6.1 Multiple Regression Classes in (C)MLLR-SVM systems

(C)MLLR deliberately constrains adaptation by means of an affine transform which, in principle, represents a strong limitation compared to unconstrained bayesian approaches. Given that MLLR transform coefficients are averaged for each Gaussian in a regression class, tying can considerably affect the quality of the adapted mean vectors if a small amount of regression classes is used. The common way of overcoming the limitation of affine transforms in (C)MLLR is increasing the number of regression classes so that each of them specializes in a smaller region of the acoustic space. Such piece-wise linear adaptation typically results in better parameter estimates for the Gaussian components. We refer to Section 3.2.3.2 for more details on this subject.

HMM-based LVCSR systems have been using more or less intricate schemes to estimate multiple MLLR transforms [Leggetter & Woodl, 1995]. It is common to compute a single feature-space SAT transform that is directly applied on cepstra, which is then used to compute multiple MLLR transforms based on phonetic-class labeling of triphone models. Hierarchical clustering techniques can be further used to manage the amount of data assigned to each regression class. In fact, any scheme can be used to obtain MLLR transforms usable by MLLR-SVM systems. Transforms computed from two decoding passes of a LVCSR system, using a phone-loop model as reference first and then the word hypotheses obtained in the first pass, are used in [Stolcke *et al.*, 2005; Stolcke *et al.*, 2006]. The system computes a small number of transforms using acoustic-level hypotheses and a greater number of transforms are computed using word-level hypotheses, which are more reliable. When male and female speaker-independent models are used, MLLR transforms can be computed with respect to each of the models [Stolcke *et al.*, 2006], obtaining twice as transforms per regression class, so that gender mismatch issues are avoided. If hierarchical clustering is used to find the optimal amount of regression classes, we must make sure that the same amount of features, i.e. the same amount of transforms, is always presented to the SVM. This can be done by always using as many transforms as leaf nodes in the tree, duplicating transform coefficients in case of back-off.

Computing multiple MLLR transforms using a UBM is not straightforward, generally requiring a previous clustering stage. The arbitrary assignment of Gaussian components to indices in GMM makes the a priori identification of acoustic regions difficult. However, being Gaussian clustering the main goal, a data-driven approach such as K-means [Duda *et al.*, 2001] using any acoustic distance measure between Gaussians can be used instead. Beyond obtaining different cluster definitions at each run, resulting in structural noise in the performance measures, there is no other drawback in its implementation. Alternatively, a phonetic decoder can be used to train several sets of Gaussians using cepstra labeled for each phonetic class and then fusing the resulting models [Karam & Campbell, 2008]. This approach, however, is not fully acoustic-based as it relies on phonetic labeling.

Given that a LVCSR system was already available at LIMSI while developing (C)MLLR-SVM systems we decided to use its acoustic models to compute multi-class (C)MLLR transforms, leaving GMM aside in this chapter.

6.2 Experimental Setup

In this chapter we explore CMLLR-SVM and MLLR-SVM systems along multiple axes, namely the type of model used to compute the regression coefficients, the front-end, the method used for model training and the amount of regression classes. Many different setups are required so that system parameters are sampled finely enough to cover these four directions in a meaningful way.

We can consider all of the systems to be conceptually derived from the CMLLR-SVM approach presented in Chapter 5 and differing in the actual choices for computing (C)MLLR transforms. The latter is the only conceptual point that is further discussed in this chapter, given that we make use of some modules of a LVCSR system. Systems using multiple transforms also follow the diagram of Figure 5.2 with the exception that the UBM is replaced with phonemic HMM and that supervectors issued from each regression class are concatenated, resulting in a supervector of supervectors.

In the following sections, we give a description of the front-end and LVCSR setups to later detail all of the systems involved in the experiments in Section 6.2.3.

6.2.1 Front-end Setup

Training the acoustic models of a LVCSR system from scratch is a hard task requiring careful operation. In situations where a speech recognition system is ready to use, it may be interesting to stick to the optimizations and constraints of pre-trained acoustic models. In this way, MLLR-SVM systems often use the LVCSR system as a black box computing MLLR transform coefficients as only output. These acoustic models have been trained using a front-end setup optimized for speech transcription which is not necessarily optimal for speaker recognition. To assess to which extent such a choice is affecting speaker recognition performance we evaluate some of our systems using two different front-end setups, one optimized for speech recognition, PLP12, and one optimized for speaker recognition, PLP15N. Details for these front-ends are given next.

- **Speech Recognition (PLP12)**:
    - **30 ms** (240 samples) analysis window size, **10ms** (80 samples) shift period
    - **12 MEL-PLP** coefficients and log-energy along with their Δ and $\Delta\Delta$ derivatives, a total of **39** features
    - **0-3.8kHz** filterbank bandwidth
    - Utterance-level cepstral mean and variance normalization is applied to each segment of interest before taking Δ and $\Delta\Delta$ derivatives. Mean and variance are estimated from each utterance obtained via an algorithm of speaker clustering.
- **Speaker Recognition (PLP15N)**:
    - **30 ms** (240 samples) analysis window size, **10ms** (80 samples) shift period
    - **15 MEL-PLP** coefficients with Δ and $\Delta\Delta$ derivatives, and Δ and $\Delta\Delta$ energies, a total of **47** features
    - **0-3.8kHz** filterbank bandwidth
    - **Feature Mapping** using two gender- and three channel-dependent models (GSM, CDMA and landline handsets) trained on 6 hours/gender (24769 segments) of speech data taken from test speakers segments in NIST SRE 1997 to 2002 evaluation data
    - **Feature warping** using a 3 second sliding window

PLP15N is the same front-end setup used by previous systems in this manuscript, described in Section 4.1.1, with the exception of the speech activity detection method used.

Corpus	Hours	Sides	Duration/Side
Fisher	280	6127	2.8 min.
Switchboard I	286	4862	3.5 min.
Switchboard II	93	2348	2.3 min.
Callhome	3	240	0.7 min.

Table 6.1 — Corpora used for training LVCSR acoustic models. Columns are the name of the corpus, the total amount of hours of speech, the number of conversation sides and the average length of conversation sides for each corpus.

During alignment of cepstra against the acoustic models of the LVCSR system, all utterances in a speech segment are considered as whole units. Using a frame-by-frame approach to speech activity detection can result in broken utterance units, proper alignment becoming unfeasible. For this reason, we use two different speech activity detection methods, one used in MLLR-SVM systems based on LVCSR acoustic models and one for UBM-based systems. The former uses forced-alignment of cepstra against transcripts or hypotheses, obtaining homegeneous segmentation. The latter uses the method described in Section 4.1.1, considering voiced frames with a minimum energy, i.e. dropping unvoiced and low energy frames. Voicing is detected using the ESPS get_f0[1] pitch extraction algorithm. Note that filtering pitch contours using dynamic programming can force contiguous voiced segments with a minimum duration, but does not consider unvoiced regions of cepstra.

6.2.2 LVCSR Acoustic Models

As in UBM-based systems, estimation of MLLR transforms using a LVCSR system is performed in two steps. First, cepstra needs to be aligned against the acoustic models, obtaining occupation probabilities $p(i|\mathbf{x_t})$ for each frame $\mathbf{x_t}$ and state i. Using the derived alignments we can assign frames to regression classes for which MLLR transforms are actually computed by solving Equation 2.24.

There are several ways in which alignments can be obtained depending on either transcripts or decoded hypotheses are used. If transcripts are available, generally in the form of an orthographic transcription for each utterance, triphone-level left-to-right HMM can be derived with the help of a pronunciation dictionary. Cepstra are then aligned against these state sequences, obtaining the optimal mapping of each cepstral vector against the triphones of the utterance model. If transcripts are not available, the LVCSR system can still be used to decode its own hypotheses, now using the language models. Each decoding pass, e.g. at the acoustic or word levels, results in different alignments that can be used to adapt the acoustic models based on different MLLR schemes. Since all data involved in our MLLR-SVM systems, i.e. impostor, target and test speaker segments, has been carefully chosen to be either manually or automatically transcribed, we only need the acoustic models and a pronunciation dictionary for MLLR transform computation. We discuss the use of our own hypotheses for computing lattice-based MLLR transforms in Chapter 7.

In our experiments we use PLP12 and PLP15N front-ends that have to match the one used by the LVCSR acoustic models. We have trained two sets of speaker-independent acoustic models based on these front-ends, plus a third one using PLP15 features and SAT, using over 650 hours of spontaneous telephone speech from corpora in Table 6.1 and a

[1] KTH Sotfware, `http://www.speech.kth.se/software`.

maximum likelihood estimation criterion:

- **PLP12 AM** are based on the acoustic models used in the first decoding pass of the LIMSI Conversational Telephone Speech CTS speech-to-text system [Gauvain *et al.*, 2003]. They use PLP12 features, gender-independent continuous density HMM with Gaussian mixtures and context-dependent tied-state triphones based on a 48-symbol phone set. 6460 tied-states with 32 Gaussian components each were found by means of a decision tree.
- **PLP15N AM** are based on PLP12 AM except that they use PLP15N features, optimized for speaker recognition. We retrained the Gaussian mixtures of PLP12 AM, keeping the PLP12 AM alignments obtained for the training data. This required extracting PLP15N features with the exact segmentation and per-utterance length obtained in the PLP12 AM training process. Since only observation distributions are retrained, the number of tied states and Gaussian components per tied state are 6460 and 32 respectively, as in PLP12 AM.
- **PLP15NSAT AM** used the PLP15N AM as seed models to one iteration of SAT re-estimation. We compute one CMLLR transform per speaker using the concatenation all of its training data. The acoustic models are then retrained using the alignments obtained with the CMLLR-transformed cepstra. Triphones are clustered again resulting in 6106 tied states, a number still comparable with that obtained for PLP12 AM and PLP15N AM.

MLLR and CMLLR transforms are always computed using the alignments obtained with their respective acoustic models. This couples both steps involved in MLLR estimation, although one set of acoustic models is rather independent[2] from another. Thus, we are not able to address the effect of regression coefficients only.

6.2.3 System Description

Based on the CMLLR-SVM approach of Chapter 5, we use a bunch of system configurations that explore the following points of interest:

- Speech and speaker recognition front-ends (PLP12 vs. PLP15N)
- Model-space CMLLR and MLLR regression coefficients for single/multiple transforms
- UBM and LVCSR acoustic models (GMM vs. HMM)
- Standard maximum-likelihood estimation and speaker adaptive training (MLE vs. SAT)

Four basic system setups derive from CMLLR and MLLR transforms and GMM and HMM, only differing in the way transform supervectors are obtained. These base supervectors follow the processing described in Section 4.1.3. Inter-session variability is compensated using NAP for all systems. We use a fixed session subspace dimension of 50, which has been optimized a posteriori on several single-class and multiple-class systems.

[2] Strictly speaking they are not independent since the three sets of acoustic models used the same base alignments for training.

For the latter, note that NAP is actually applied after concatenation of supervectors for the involved regression classes. The inter-session covariance matrix used by NAP captures cross-transform session variability and, thus compensation is jointly performed within and across transforms. After NAP compensation, supervector components are scaled to fit minimum and maximum values, then classified with a SVM using a linear kernel.

We describe how base MLLR or CMLLR supervectors are obtained for each of these setups in the following sections.

6.2.3.1 HMM-based MLLR-SVM

This setup computes MLLR transforms using LVCSR acoustic models. We obtain one set of MLLR transforms for each conversation side, involving a fixed number of classes, either one, two or three, always excluding non-speech[3]. These classes are obtained using basic phonetic criteria, actually corresponding to triphone states having vowels or consonants as central phone. We use acronyms

- **1t**, one transform, speech only
- **2t**, two transforms, consonants and vowels
- **3t**, three transforms, consonants and two sets of vowels

to refer to them. Using up to three transforms assures a minimum amount of speech assigned to each regression class. Although rather conservative, this seems to be a reasonable approach for segments with controlled mean duration, as it is the case of NIST SRE campaigns. This assumption allows us to force estimation of full MLLR matrices, thus avoiding backing-off to diagonal matrices when a certain amount of data is not reached[4]. As mentioned in Section 6.2.2 MLLR transforms are obtained in a supervised way using transcripts generated automatically using a speech recognition system and provided by NIST for SRE 2004, SRE 2005 and SRE 2006 corpora. Many of the impostor speaker segments are taken from the Switchboard I corpus which has manually-derived transcripts available. The pronunciation dictionary is based on that used in the LIMSI RT'03 LVCSR system. The few missing entries were added manually.

The system uses either PLP12 or PLP15N features. The former result in 1560 coefficients per transform, including offset $\mathbf{b}$, and uses PLP12 acoustic models as described in Section 6.2.2. PLP15N features results in 2256 coefficients per transform, including $\mathbf{b}$ as well, and use PLP15N and PLP15SAT acoustic models. Base supervectors are obtained as the concatenation of MLLR transform coefficients for all regression classes.

6.2.3.2 HMM-based CMLLR-SVM

This is exactly the same setup used for the HMM-based MLLR-SVM system above except that it computes model-space CMLLR transforms. Cesptra assigned to each regression class are used to obtain feature-space CMLLR transforms with parameters $(\mathbf{A^{-1}}, -\mathbf{A^{-1}b})$ that are inverted to yield $(\mathbf{A}, \mathbf{b})$, thus becoming fully comparable with those obtained for MLLR transforms. All choices of regression classes, front-ends, acoustic models are kept the same as in HMM-based MLLR-SVM.

[3] The non-speech transform is dropped as it is assumed to carry no speaker information.

[4] According to experiments not included in this manuscript, backing off to diagonal MLLR matrices using the threshold optimized for speech recognition resulted in increased error rates for 2t and 3t classes. The back-off rate increased enormously when using 3t and PLP15N features.

CMLLR System	MLLR System	Front-end	Model	SAT	#Trans.
CG12	MG12	PLP12	GMM	×	1
CG12 SAT	MG12 SAT	PLP12	GMM	√	1
CG15	MG15	PLP15N	GMM	×	1
CG15 SAT	MG15 SAT	PLP15N	GMM	√	1
CH12 1t	MH12 1t	PLP12	HMM	×	1
CH12 2t	MH12 2t	PLP12	HMM	×	2
CH12 3t	MH12 3t	PLP12	HMM	×	3
CH15 1t	MH15 1t	PLP15N	HMM	×	1
CH15 1t SAT	MH15 1t SAT	PLP15N	HMM	√	1
CH15 2t	MH15 2t	PLP15N	HMM	×	2
CH15 2t SAT	MH15 2t SAT	PLP15N	HMM	√	2
CH15 3t	MH15 3t	PLP15N	HMM	×	3
CH15 3t SAT	MH15 3t SAT	PLP15N	HMM	√	3

Table 6.2 — System naming convention for (C)MLLR-SVM systems. Columns specify acronyms for systems using CMLLR and MLLR transforms, front-end used (PLP12 vs. PLP15N), type of model used to compute transforms (GMM vs. HMM), SAT (√) or regular (×) model training and number of transforms (1 to 3). The thick horizontal rule splits systems using GMM or HMM and the thin rule splits systems using PLP12 and PLP15N front-ends.

6.2.3.3 GMM-based CMLLR-SVM

This setup is fully based on the CMLLR-SVM system presented in Chapter 5. We use two gender-dependent UBM with 512 Gaussians each, trained using 5 iterations of maximum likelihood estimation and a variance floor of 1% of the total variance. The system can use either PLP12 or PLP15 features, for fair comparison against PLP12, PLP15N and PLP15N SAT acoustic models used in HMM-based MLLR-SVM and CMLLR-SVM systems. Single-class model-space CMLLR transforms with coefficients $(\mathbf{A}, \mathbf{b})$ are obtained by inversion of the corresponding feature-space transforms, resulting in supervectors with 1560 and 2256 features for PLP12 and PLP15N front-ends respectively. We limit the number of SAT iterations to one as used in PLP15NSAT acoustic models.

6.2.3.4 GMM-based MLLR-SVM

This is exactly the same setup used for the GMM-based CMLLR-SVM system above except that it computes MLLR transforms. UBM training and SAT and front-end variants are kept the same.

6.2.4 Experimental Results

Given the large number of configurations we have found necessary to give concise naming to each of them. System names use a short acronym summarizing:

- Type of transforms: **M** for MLLR, **C** for CMLLR

	SRE 2005				**SRE 2006**			
	MDC		EER (%)		MDC		EER (%)	
System	F	FB	F	FB	F	FB	F	FB
CG12	.0342	.0308	7.57	8.15	.0290	.0265	6.88	6.71
CG12 SAT	.0326	.0293	7.82	7.72	.0287	.0253	6.53	6.20
CG15	.0303	.0276	7.93	7.86	.0256	.0241	5.96	5.61
CG15 SAT	.0292	.0265	7.27	6.99	.0247	.0241	5.88	5.61
CH12 1t	.0329	.0298	7.53	7.24	.0292	.0258	6.70	6.39
CH12 2t	.0307	.0273	6.70	6.74	.0281	.0243	5.93	5.83
CH12 3t	.0310	.0267	6.61	6.66	.0284	.0241	6.20	6.11
CH15 1t	.0264	.0237	6.28	6.28	**.0230**	.0217	5.46	5.24
CH15 1t SAT	.0286	.0244	6.32	6.36	.0237	.0220	5.84	5.56
CH15 2t	.0251	.0233	**5.99**	6.19	.0232	**.0216**	5.27	5.33
CH15 2t SAT	.0258	.0228	6.19	6.24	.0235	.0217	5.37	5.37
CH15 3t	.0249	**.0222**	6.08	**5.85**	.0237	.0222	5.24	**5.10**
CH15 3t SAT	**.0243**	.0227	6.07	6.02	.0238	.0218	**5.18**	**5.10**

Table 6.3 — MDC and EER of CMLLR-SVM systems for SRE 2005 and SRE 2006 using multiple transforms, front-ends and model training schemes. All systems use NAP compensation of concatenated CMLLR transform supervectors with a subspace dimension of 50. The thicker rule corresponds splits systems using GMM or HMM and the thin rule splits systems using PLP12 and PLP15N front-ends. Columns F and FB show forward and averaged forward-backward scores respectively. The best scores of each column are shown in boldface.

- Model used to compute transforms: **G** for GMM, **H** for HMM
- Front-end: **12** for PLP12, **15** for PLP15N
- Number of transforms: **1t**, **2t** or **3t** for one, two or three transforms in LVCSR-based systems, nothing for UBM-based systems
- Training: **SAT** for speaker adaptive training, nothing if standard maximum likelihood estimation is used

A total of 26 system variants along with their acronyms are shown in Table 6.2. Note that each row corresponds to two systems, a MLLR-SVM and a CMLLR-SVM system. Tables 6.3 and 6.4 show MDC and EER measures for CMLLR-SVM and MLLR-SVM systems and we discuss them in the same order.

6.2.4.1 CMLLR-SVM System Results

CMLLR-SVM systems exhibit a considerable variation of performance across configurations, over 20% MDC and EER relative difference from worst to best performing systems. Figures 6.1(a) and 6.1(b) show constellation plots of forward scoring systems for SRE 2005 and SRE 2006. GMM-based and HMM-based systems roughly split these plots into two areas. Systems using PLP12 features with symbols ■ and ● tend to cluster in the top-right half

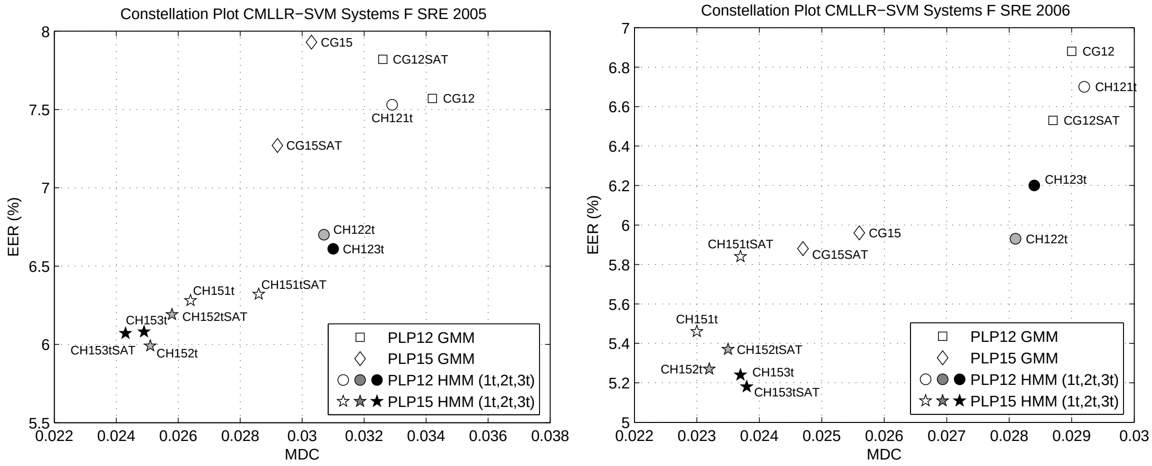

Figure 6.1 — Constellation plots of forward-scoring CMLLR-SVM systems using different models, training and number of transforms for SRE 2005 (left, a) and SRE 2006 (right, b). Symbol shapes indicate a front-end/model combination as indicated in the legend of the graph while gray levels indicate the number of transforms, the more transforms used the darker.

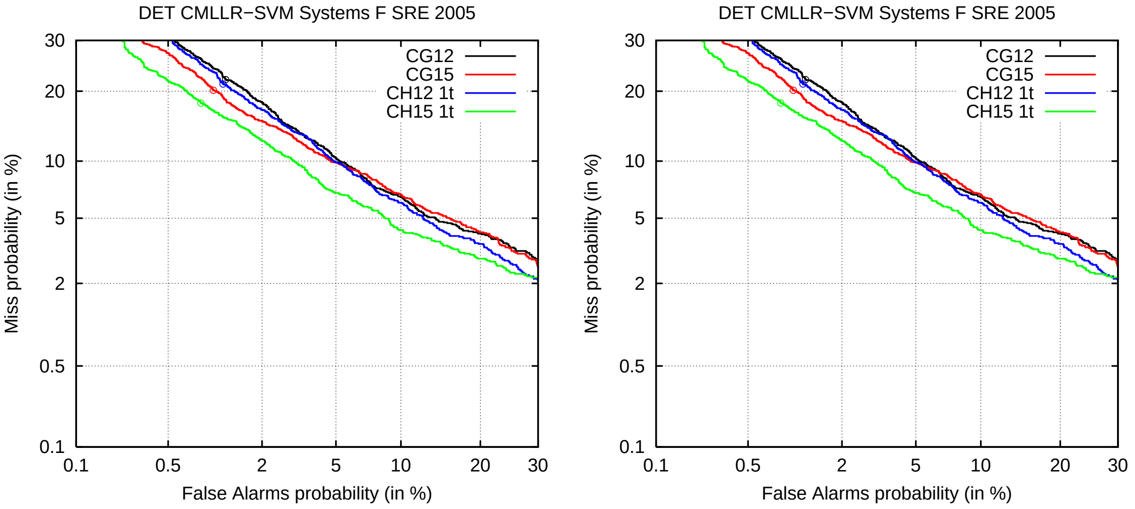

Figure 6.2 — DET curves of CMLLR-SVM systems using PLP12 and PLP15N features, one CMLLR transform and either GMM or HMM.

of the plot, while systems using PLP15N features with symbols ◇ and ⋆ lie in the bottom-left best-performing area. This is particularly visible for SRE 2006, where CG15 and CG15 SAT outperform multi-class systems CH12 1t, CH12 2t and CH12 3t. The latter use a multi-class CMLLR scheme and much more complex acoustic models compared to the former, suggesting the use of the PLP15N front-end as a major factor explaining the observed difference. This hypothesis is further supported by the fact that the phenomenon is more evident in SRE 2006, which involves stronger channel variability in the core condition compared to SRE 2005. Note that, for SRE 2006, NAP compensation of multi-class CMLLR supervectors in CH12 systems is not yet able to improve performance beyond CG15. Therefore, PLP15N

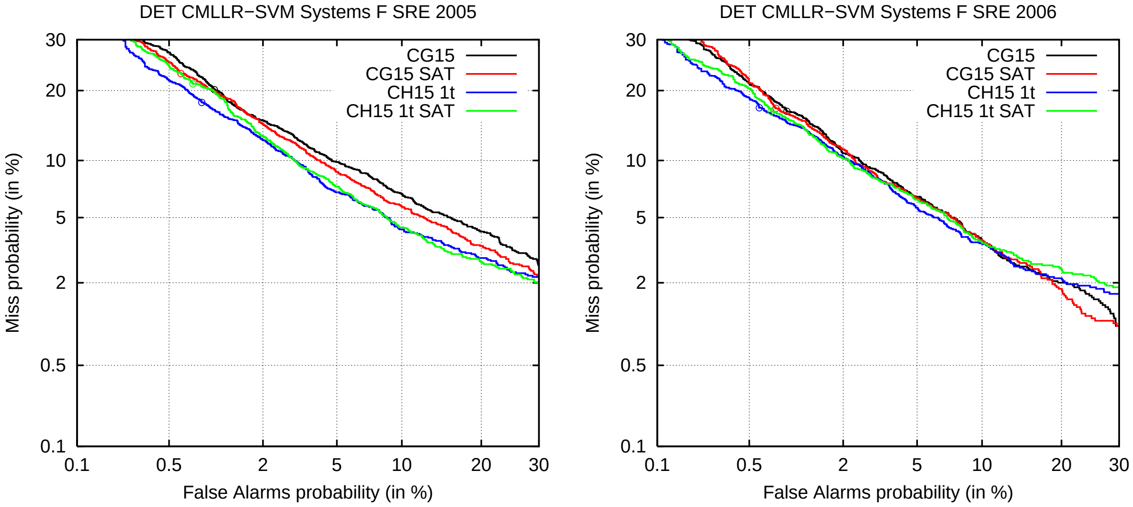

Figure 6.3 — DET curves of CMLLR-SVM systems using standard ML and speaker adaptive training of acoustic models. Systems use PLP15N features, one CMLLR transform and either GMM or HMM.

features are highly responsible for the difference of performance observed, although we do not know to which extent feature mapping and feature warping are affecting here[5]. By switching from PLP12 to PLP15N features we obtained averaged MDC and EER improvements for all systems of over 13%, although a smaller 8% EER for SRE 2005. Systems using HMM particularly benefit from it with an average gain of 16% MDC and 13% EER versus 12% MDC and 10% EER for GMM-based systems. These gains are stable within HMM-based and GMM-based categories. Figures 6.2(a) and 6.2(b) show DET curves for systems using PLP12 and PLP15N front-ends and either GMM or HMM. Note that gains for HMM-based systems are considerably larger than for GMM-based systems for most operating points.

In a similar way, results show that improvements obtained by switching from GMM to HMM depend on the front-end used. PLP12 features give small gains, 1.5% for both MDC and EER, while PLP15 obtain an averaged 5% when using SAT and 13% without it. GMM-based systems gather in top-right areas of constellation plots of Figures 6.1(a) and 6.1(b) and HMM-based occupy more performing areas.

The use of SAT results in no clear gains considering all systems overall, no larger than 2% MDC and EER. Overall, HMM-based systems obtain no improvement by using SAT while GMM-based systems get over 3% MDC and 4% EER relative gains. In the constellation plots of Figures 6.1(a) and 6.1(b) SAT shifts HMM-based systems in rather arbitrary directions while GMM-based systems are shifted in a top-right to left-bottom direction. Figures 6.3(a) and 6.3(b) show DET curves of systems PLP15N features, one CMLLR transform and either GMM or HMM. For SRE 2005, SAT does not help for most of operating points of CH15 1t but it does help for CG15 consistently. For SRE 2006, these differences are smaller particularly for CG15. Interaction of SAT together with the front-end type cannot be assessed as only one PLP12 SAT system was explored.

[5] Feature warping does not address linear distortion directly, although normalizing the feature distribution has an indirect impact on it. Matching a Gaussian distribution $\mathcal{N}(0, 1)$ has the effect of removing the mean of the processed features and, thus, it can be seen as performing cepstral mean substraction, used to remove linear distortion.

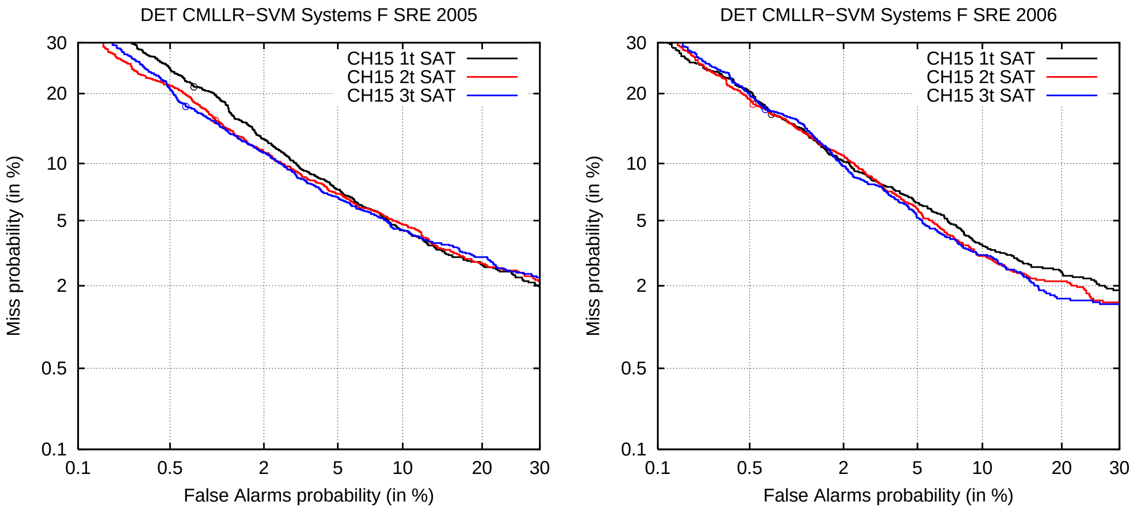

Figure 6.4 — DET curves of CMLLR-SVM systems using one, two and three transforms, PLP15N features and PLP15N SAT acoustic models.

CMLLR schemes using two or more transforms obtain improved performance compared to one-transform systems. Using PLP12 features, most of the gain is obtained by passing from one to two transforms, around 8% MDC and EER for SRE 2005 and 5% MDC and 10% EER for SRE 2006. Adding a third transform brings less than 1% MDC and EER for SRE 2005 and a considerable loss for SRE 2006. For systems using PLP15N features, adding more transforms always results in some kind of improvement, although sometimes gains in MDC are lost in EER, and viceversa. In general terms, using more transforms decreases MDC for SRE 2005 and EER for both SRE 2005 and SRE 2006. No improvement and eventually a considerable loss is observed for MDC and SRE 2006. This is visible, for instance, in the constellation plot of Figure 6.1(b), where CH15 systems follow a top-left to bottom-right trend as they use one, two and three transforms. For systems using SAT the difference of performance is more clear. DET curves for CH15 SAT systems are shown in Figures 6.4(a) and 6.4(b). For SRE 2005, a considerable improvement in the low false-alarm rate area is obtained by passing from one to two transforms, while adding a third transform is beneficial near MDC and EER operating points but not clearly for other regions of the curve. In fact, using multiple transform seems to rotate the DET curve as well, since we lose some performance in the high false-alarm rate region with respect to CH15 1t SAT. For SRE 2006, the three curves obtain rather similar performance in the low false-alarm area including MDC. For EER and operating points with higher false-alarm rates, using multiple transforms outperform the single-transform system CH15 1t SAT.

As for the scoring strategy, forward-backward score combination results in rather consistent gains across systems, which tend to be fairly large for the MDC operating point, around 9% averaging all systems, but small for EER, with only 1%. These improvements are similar across type of model and number of transforms used.

6.2.4.2 MLLR-SVM System Results

Table 6.4 shows results for MLLR-SVM systems. A large difference of performance, eventually reaching up to 50% in relative terms, is found across system configurations. Further-

	SRE 2005				**SRE 2006**			
	MDC		EER (%)		MDC		EER (%)	
System	F	FB	F	FB	F	FB	F	FB
MG12	.0384	.0363	9.81	9.36	.0310	.0290	7.40	7.13
MG12 SAT	.0326	.0310	8.07	7.94	.0272	.0248	6.16	5.79
MG15	.0367	.0341	8.90	8.86	.0304	.0282	7.72	7.58
MG15 SAT	.0278	.0264	7.20	7.28	.0244	.0226	5.70	5.43
MH12 1t	.0348	.0310	7.98	7.64	.0281	.0250	5.92	5.79
MH12 2t	.0310	.0268	6.94	6.65	.0246	.0207	5.37	5.05
MH12 3t	.0298	.0262	6.94	6.78	.0261	.0223	5.28	5.01
MH15 1t	.0254	.0232	5.74	5.70	.0207	.0189	5.10	4.82
MH15 1t SAT	.0198	.0180	4.86	4.57	.0183	.0164	4.32	4.31
MH15 2t	.0222	.0201	5.86	5.74	.0193	.0171	4.27	4.15
MH15 2t SAT	**.0180**	**.0159**	**4.53**	**4.33**	**.0155**	.0143	**3.45**	**3.57**
MH15 3t	.0219	.0201	5.45	5.41	.0191	.0172	4.27	4.23
MH15 3t SAT	.0183	.0165	4.91	4.82	**.0155**	**.0136**	3.77	3.68

Table 6.4 — MDC and EER of MLLR-SVM systems for SRE 2005 and SRE 2006 using multiple transforms, front-ends and model training schemes. All systems use NAP compensation of concatenated MLLR transform supervectors with a subspace dimension of 50. The thicker rule corresponds splits systems using GMM or HMM and the thin rule splits systems using PLP12 and PLP15N front-ends. Columns F and FB show forward and averaged forward-backward scores respectively. The best scores of each column are shown in boldface.

more, best performing systems, MH15 SAT series of systems, achieve much lower absolute MDC and EER measures than their CMLLR-SVM counterparts. Figures 6.5(a) and 6.5(b) present constellation plots for all systems for SRE 2005 and SRE 2006. Systems using PLP12 features gather in the top-right area while those using PLP15N features do in the bottom-left area. As in CMLLR-SVM, the front-end used can still be deemed as a major source of improvement. HMM-based systems obtain an average relative gain of around 20% for both MDC and EER, while it is smaller, 12% MDC and 8% EER for GMM-based systems using SAT, and even smaller for basic GMM-based systems, eventually resulting in a loss of performance. This suggests than MLLR takes considerable advantage of modeling complexity of LVCSR acoustic models, although CMLLR-SVM systems do to a lower extent too, with gains for HMM-based systems of 16% MDC and 13% EER. DET curves of Figures 6.6(a) and 6.6(b) show curves for systems using one transform only. The improvement obtained for HMM-based systems is consistently important at almost all operating points for both SRE 2005 and SRE 2006. GMM-based systems also obtain gains at all operating points. Note that their CMLLR-SVM counterparts, i.e. Figures 6.2(a) and 6.2(b) show smaller improvements, using HMM-based modeling in particular.

Another important source of performance improvement is the type of model used to compute MLLR transforms. Beyond front-end criteria, we can visually cluster systems by model in the constellation plots of Figures 6.5(a) and 6.5(b). GMM-based systems occupy the top-right less-performing part of the plots, with the corresponding HMM-based systems MH12 1t, MH15 1t and MH15 1t SAT lying far from them. Systems using PLP15N

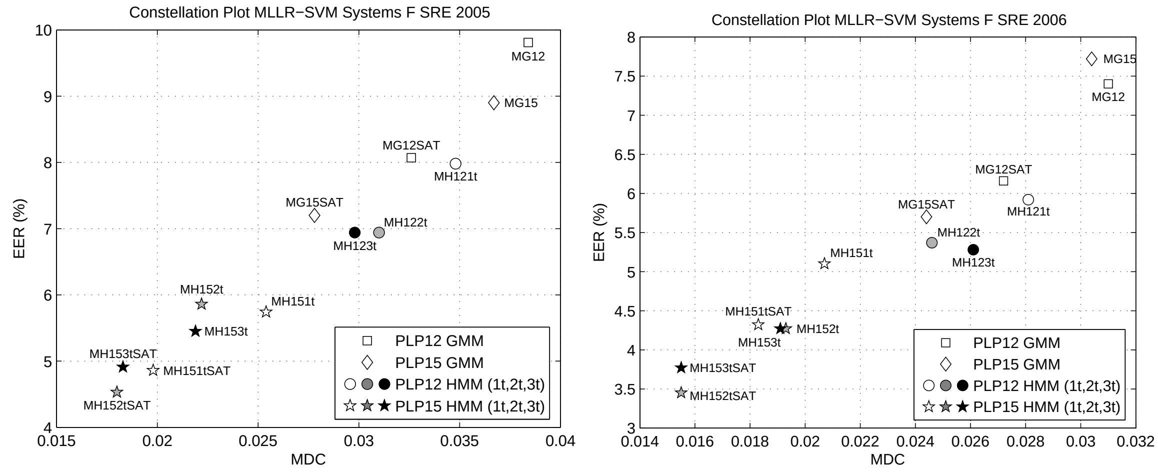

Figure 6.5 — Constellation plots of forward-scoring MLLR-SVM systems using different models, training and number of transforms for SRE 2005 (left, a) and SRE 2006 (right, b). Symbol shapes indicate a front-end/model combination as indicated in the legend of the graph while gray levels indicate the number of transforms, the more transforms used the darker.

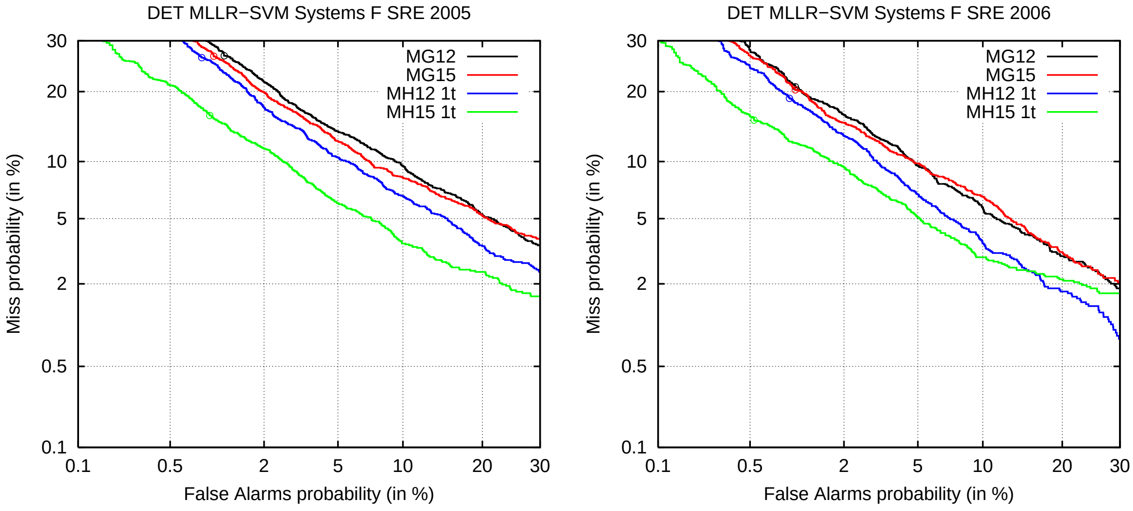

Figure 6.6 — DET curves of MLLR-SVM systems using PLP12 and PLP15N features, one MLLR transform and either GMM or HMM.

features result in enormous gains, around 30% MDC and EER in average, and those using PLP12 features stay around 15%. These gains are enormous compared to those obtained for CMLLR-SVM systems, i.e. 13% and 1.5% respectively.

SAT results in considerable gains regardless of the front-end setup[6], type of model and number of transforms in all setups involved . We obtain very stable improvements, 19% by averaging MDC and EER relative gains for all systems. SAT behavior is also clear in

[6] Although for PLP12 features, only systems CG12 SAT and CG12 can be compared.

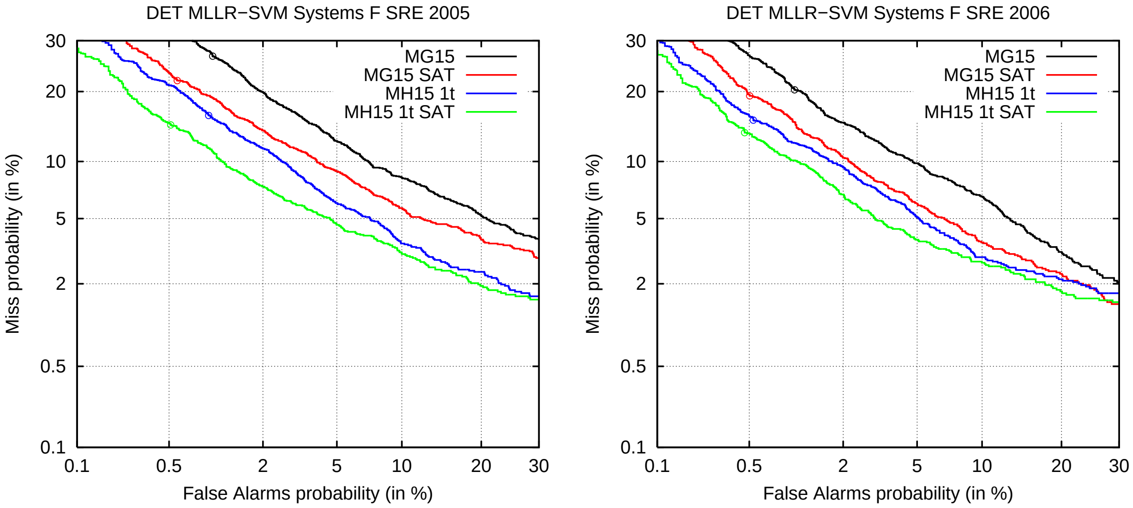

Figure 6.7 — DET curves of MLLR-SVM systems using standard ML and speaker adaptive training of acoustic models. Systems use PLP15N features, one MLLR transform and either GMM or HMM.

the constellation plots of Figures 6.5(a) and 6.5(b) where systems using SAT lie far from their counterparts in bottom-left direction. For PLP15N features, also note that the worst MH15 SAT system is already more performing than the best MH15 system regardless of the number of transforms used. For these systems, SAT is a major source of performance improvements, visually more relevant than the number of transforms. DET curves of single-transform GMM- and HMM-based systems in Figures 6.7(a) 6.7(b) show that the magnitude of SAT improvement is maintained along their respective curves. Gains for HMM-based systems is slightly reduced for SRE 2006, probably due to the improvement taken by NAP session compensation. Note that the corresponding curves for CMLLR-SVM systems of Figures 6.3(a) and 6.3(b) show a different scenario with much smaller gains, while absolute MDC and EER are still favorable to MLLR-SVM.

Regarding the number of transforms used, most of the gain is visible when passing from one to two transforms, whereas adding a third transform can eventually result in loss of performance. We can see in constellation plots of Figures 6.5(a) and 6.5(b) that black symbols are not necessarily south-east of gray symbols. This can be due to the number of transforms, but also to the choice of the regression classes used. The third class is obtained by splitting the vowel class, which was decided intuitively. We believe that other classes also based on phonological traits, e.g. obstruent or nasal classes, could have led to other conclusions. Using two transforms obtains averaged gains of 12% MDC and 10% EER for PLP12 and PLP15N systems. For the latter, SRE 2006 results in larger gains than SRE 2005, as can be seen in the DET curves of Figures 6.8(a) and 6.8(b) for systems using SAT. Given the larger variability in the core condition of SRE 2006, we suggest that using multiple-transform MLLR transform supervectors might have a positive effect on NAP compensation efficiency as well. Behavior of MLLR-SVM and CMLLR-SVM systems regarding the use of multiple transforms is quite similar for SRE 2005 but CMLLR-SVM systems fail to take advantage of it for SRE 2006 as seen when comparing Figures 6.8(b) and 6.4(b).

MLLR-SVM and CMLLR-SVM systems exhibit similar behavior regarding forward-backward scoring, i.e. considerable improvements overall, specially in MDC, with occasional performance loss. HMM-based systems obtain relative gains of 11% MDC and 2.5%

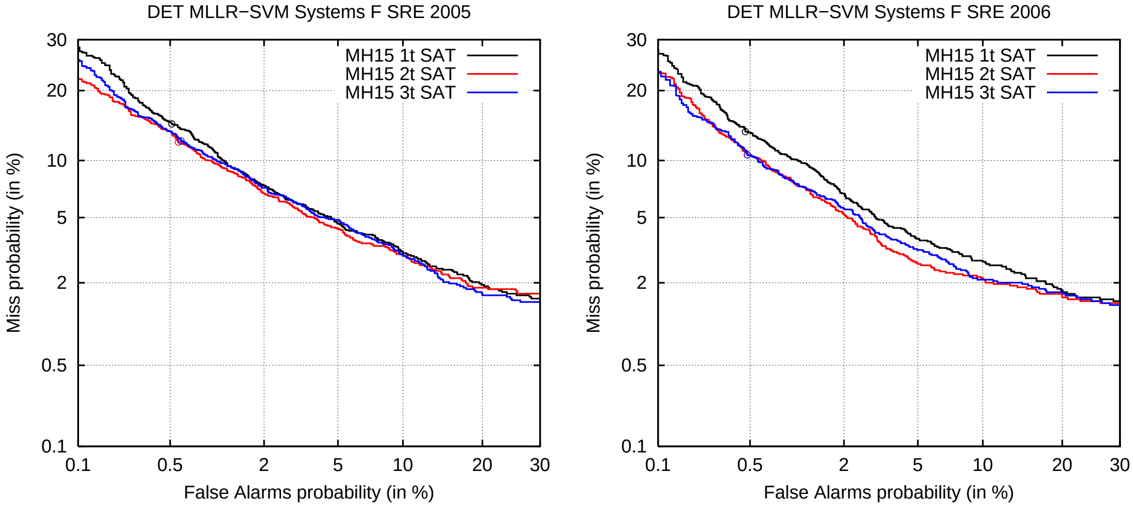

Figure 6.8 — DET curves of MLLR-SVM systems using one, two and three transforms, PLP15N features and PLP15N SAT acoustic models for SRE 2005 (left,a) and SRE 2006 (right,b).

EER, becoming a bit smaller in MDC, around 7%, for GMM-based systems.

6.3 CMLLR versus MLLR in (C)MLLR-SVM Systems

As we have seen in Section 5.1, CMLLR uses a single affine transform for mean and covariance adaptation while standard MLLR uses such transform for mean adaptation only, both using the same amount of parameters. The former ties mean and covariance parameters in such a way that the transform can be equivalently performed in the cepstral domain and, conversely, any feature-space affine transform results in adapted mean vector and covariance matrices. It is this duality between model-space and feature-space CMLLR adaptation that results in a tight relation between the respective transforms. In fact, both adaptation processes keep an inverse relationship, thus CMLLR transforms being required to be invertible while still resulting in proper adaptation. Making sure $\mathbf{A}$ is invertible assures that the speaker-dependent and speaker-independent covariance matrices, i.e. $\mathbf{A\Sigma A^T}$ and $\mathbf{A^{-1}\hat{\Sigma}A^{-T}}$, are invertible too if $\mathbf{\Sigma}$ already was. Note that such a property is required in Equation 2.24 for successful CMLLR or MLLR estimation, and it allows computing CMLLR transforms on CMLLR-transformed cepstra, enabling speaker adaptive training as well. MLLR adaptation focuses on model-space adaptation only, putting no constraints in this sense.

Invertibility of an affine transform with parameters $(\mathbf{A}, \mathbf{b})$ reduces to invertibility of its linear part, i.e. matrix $\mathbf{A}$. Therefore, behavior of MLLR and CMLLR must be visible in the inversion properties of this matrix. We saw while dealing with alternative representations of CMLLR transforms in Section 5.6.2, that SVD factorization of $\mathbf{A} = \mathbf{USV^T}$ reveals tying of its singular values across model- and feature-space transforms. Indeed, singular values of $\mathbf{A}$ correspond to the reciprocal of the singular values of $\mathbf{A^{-1}}$ while singular vectors of both matrices, i.e. matrices $\mathbf{U}$ and $\mathbf{V}$ using the decomposition of Equation 5.14, are the same except for a transposition operation. This means that, for instance, singular values

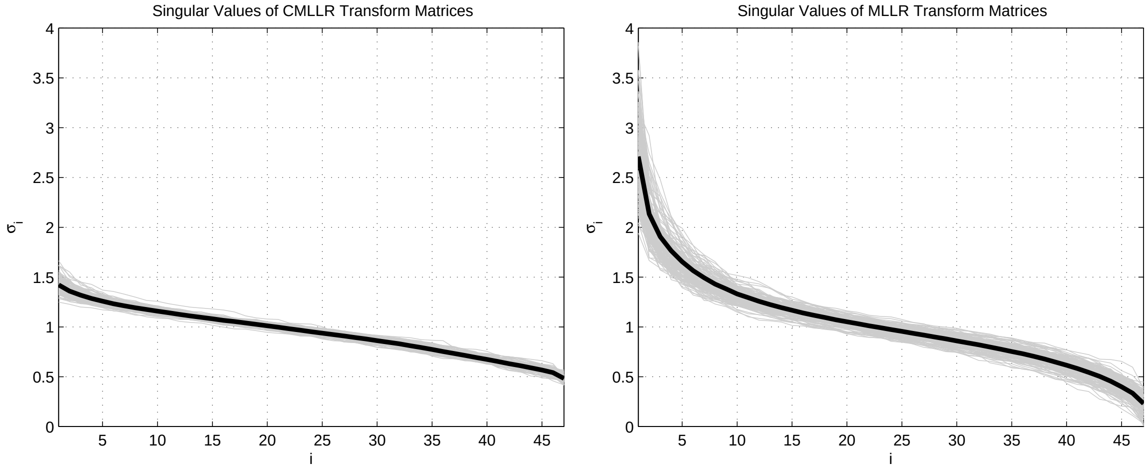

Figure 6.9 — Comparison of singular values of CMLLR (left,a) and MLLR (right,b) transform matrices for a random set of 150 speakers in the SRE 2005 training corpus. Both curves correspond to model-space adaptation of PLP15N cesptra and use the same vertical scale. Gray curves are per-segment curves. Thick black lines are averaged singular value curves.

of a model-space CMLLR transform matrix capture feature-space adaptation components as well, since its smallest singular values become the largest and most representative of its inverse. The reciprocal function is only defined for values different of zero, thus forcing singular values of $\mathbf{A}$ and $\mathbf{A^{-1}}$ to keep away from zero, or otherwise at least one of the matrices is not defined. None of these constraints applies to singular values of MLLR transform matrices.

Figures 6.9(a) and 6.9(b) show singular values of CMLLR and MLLR model-space transform matrices, $\mathbf{A}$ for a random set of 150 speakers in the SRE 2005 training data, using a UBM trained using standard maximum likelihood estimation. CMLLR transforms present rather smooth profiles whereas maximum and minimum singular values of MLLR transform matrices take more extreme values, some of them approaching 0. The average singular value spread of $\mathbf{A}$, i.e. average condition number, is around 3 for CMLLR transform matrices while it is 12 for MLLR transform matrices, 4 times larger[7]. Keeping a small singular value spread assures invertibility of CMLLR transform matrices, but pushes the matrix away from capturing all of the linear correlation present in the adaptation data. As an example of this, consider the images of Figures 6.10(a) and 6.10(b) corresponding to two sample CMLLR and MLLR transform matrices. Both show a strong diagonal corresponding to a direct feature-to-feature mapping. However, two other diagonal contributions, relating 15 PLP coefficients to their $\Delta\Delta$ features, are observable in the CMLLR matrix but not in the MLLR matrix. Keeping in mind that cepstra and $\Delta\Delta$ features are structurally correlated whether cepstra is adapted to a speaker or not, MLLR is capable of judging such components as non-relevant for adaptation while CMLLR does indeed. It seems clear that, in this example, CMLLR is sacrificing some adaptation power that could be used otherwise to capture speaker components. Small singular values of MLLR transform matrices are caused by rather redundant components of adaptation cepstra. The global inter-speaker variance of singular values turns out to be 0.002 for CMLLR versus 0.054 for MLLR, 25 times larger

[7] Similar values were found using PLP12 features and transforms from multiple acoustic classes.

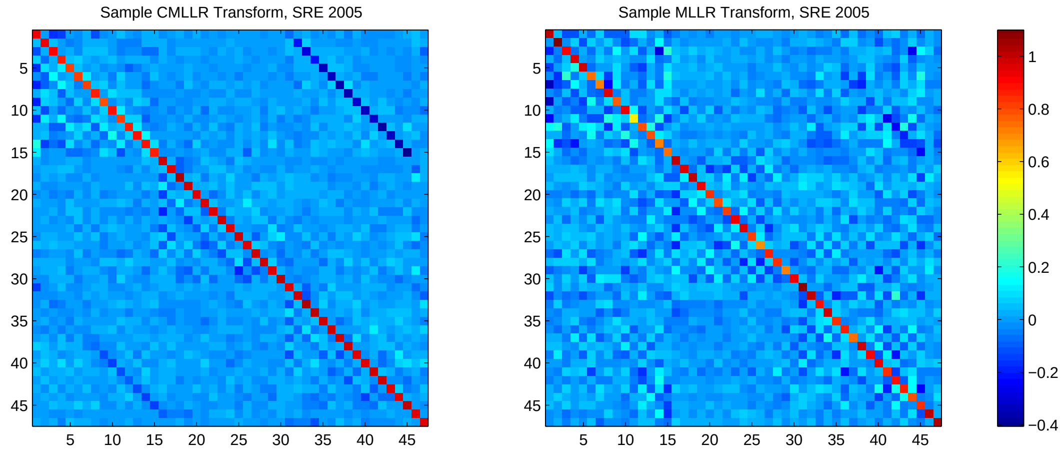

Figure 6.10 — Graphical representation of sample model-space CMLLR and MLLR transform matrices for the same speech segment computed using PLP15N features and a GMM-UBM trained via maximum-likelihood estimation.

deviation for MLLR transform matrices[8]. In such terms, MLLR transforms also seem to better adapt to speaker data compared to CMLLR transforms.

Symmetry properties of CMLLR and MLLR transform matrices are indirectly related to non-singularity by the concept of orthogonality. Consider reducing the singular value spread of a CMLLR transform matrix $\mathbf{A}$ down to 1, e.g. by setting all singular values to 1. The corresponding matrix $\mathbf{S}$ containing its singular values is then a diagonal matrix which is the inverse of itself, i.e. $\mathbf{S} = \mathbf{S^{-1}}$. In this case, straightforward manipulation of the SVD decomposition of $\mathbf{A}$ yields $\mathbf{A^{-1}} = \mathbf{A^T}$, that is, an orthogonal matrix. This very same condition can be rewritten as $\mathbf{AA^T} = \mathbf{I}$ or even $\mathbf{AA^T} - \mathbf{I} = 0$. The latter can be used to derive an non-orthogonality measure by quantifying the residual $||\mathbf{AA^T} - \mathbf{I}||_2$, where we use the induced 2-norm[9]. For the CMLLR and MLLR transform matrices analyzed before, we obtained non-orthogonality measures, averaged across the 150 speakers, of 1.04 and 7.01 respectively, that is, CMLLR matrices are much more orthogonal than MLLR counterparts. Therefore, $\mathbf{A^{-1}}$ can be considered to have similar coefficients to $\mathbf{A^T}$, hence to $\mathbf{A}$ as well. This suggests that both CMLLR adaptation directions capture relatively poor information, since they use rather similar coefficients although they focus on inverse processes.

From the discussion above, we assume that MLLR transforms result in a larger diversity of matrices compared to CMLLR, at least in terms of its singular values and, thus, their regression coefficients are capable of further specialization as well. Based on such rationale we expect to gain more insight into the experimental results presented in the previous sections. Comparing CMLLR-SVM and MLLR-SVM system results of Tables 6.3 and 6.4 in a setup-per-setup basis we observe that MLLR-based systems outperform CMLLR-based counterparts as more and more information is expected to be captured by the involved regression coefficients. Figures 6.11(a) and 6.11(b) show bar plots of MDC and EER relative improvement of MLLR-SVM vs. CMLLR-SVM systems using PLP15N features as a function of system setup for SRE 2005 and SRE 2006. The horizontal x-axis has been slightly

[8] $trace(\mathbf{A}) = trace(\mathbf{S})$ when $\mathbf{A} = \mathbf{USV^T}$.

[9] For any matrix $\mathbf{X}$ the induced 2-norm is obtained as $||\mathbf{X}||_2 = \sqrt{\lambda_{max}}$, where λ_{max} is the largest eigenvalue of $\mathbf{X^T X}$. Other norms could have been used, but $||\cdot||_2$ is consistent with the definition of condition number used previously, as $\text{cond}(\mathbf{X}) = ||\mathbf{X}||_2||\mathbf{X}||_2^{-1} = \sqrt{\lambda_{max}}/\sqrt{\lambda_{min}} = \sigma_{max}/\sigma_{min}$.

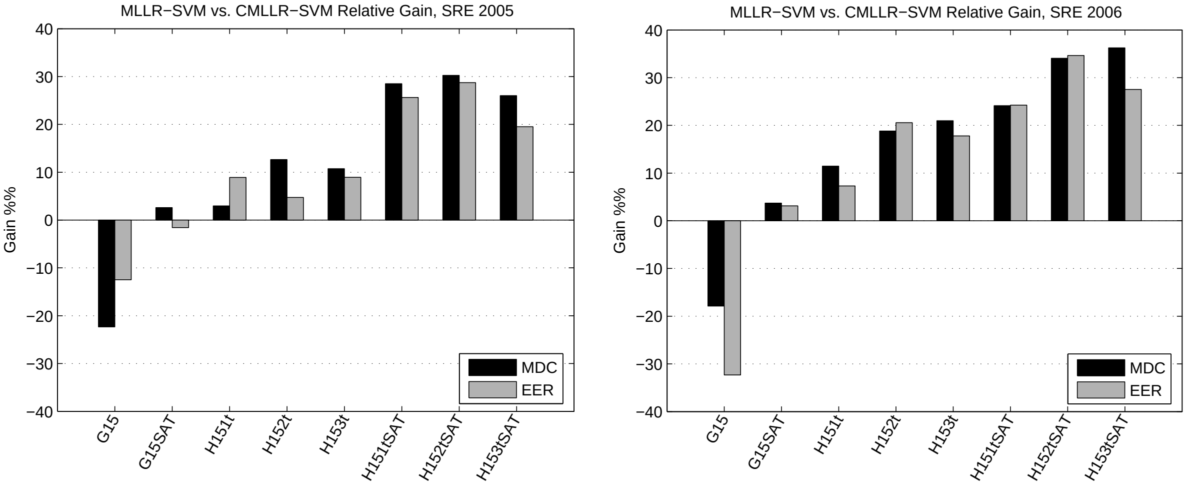

Figure 6.11 — Relative MDC and EER gain for MLLR-SVM versus CMLLR-SVM systems as a function of system setups of increasing complexity for SRE 2005 (left,a) and SRE 2006 (right,b). All systems use PLP15N features.

reordered to follow increasing complexity[10]. We can see that CMLLR-SVM systems largely outperform their MLLR counterparts for low-complexity systems, hence negative relative gains. This trend is reversed as transforms must capture more and more nuances as a result of further constraints imposed by the system. MLLR-SVM systems quickly outperform CMLLR-SVM by a larger and larger margin, eventually reaching gains over 30%. Gains become favorable to MLLR-SVM as soon as HMM are used, roughly increasing as more transforms are used, particularly when SAT is added on top of it.

Therefore, the constraint imposed by CMLLR seems advantageous for simple systems seeking global adaptation but it prevents detailed adaptation in more complex scenarios exploiting stricter restrictions. This agrees with the trend observed in MLLR adaptation of speech recognition systems, using a simple CMLLR scheme followed by a more detailed one using MLLR.

6.4 Combination of CMLLR and MLLR Transform Coefficients in (C)MLLR-SVM Systems

Given the considerably different behavior of CMLLR-SVM and MLLR-SVM systems we can think of taking advantage of both CMLLR and MLLR transforms simultaneously in a (C)MLLR-SVM system and still expect them to interact positively. We propose two main schemes for obtaining CMLLR and MLLR transform coefficients using SAT models.

Up to now, we used SAT only to render the acoustic models more speaker-independent. MLLR-SVM systems in previous sections compute MLLR transforms using cepstra unmatched to the speaker-independent SAT models. Feature-space CMLLR transforms obtained using these models map cepstra back to the same speaker-independent space used in the models. As a first alternative for CMLLR/MLLR combination, we propose to first

[10] The amount of parameters in the acoustic models, the total amount of parameters used for all transforms as well as the strength of components captured by transforms, e.g. using more speaker-independent SAT models, have been intuitively balanced in such criterion.

apply the corresponding feature-space CMLLR transform so that both cepstra and models work in a homogeneous acoustic space to later compute multiple MLLR transforms. In such a system, MLLR transforms are expected to capture weaker components as CMLLR-transformed cepstra and SAT models are brought closer. The remaining components are captured by the CMLLR transform. We use the concatenation of model-space CMLLR and MLLR transform coefficients as actual features used for classification, the former being obtained by inversion of its feature-space counterpart.

A second approach is to use CMLLR and MLLR transform coefficients, using unmatched cepstra and SAT models, directly for SVM classification. This approach allows the use any combination of multi-class CMLLR and MLLR transforms. We focus on setups using one CMLLR and multiple MLLR transforms for comparison against the preceding approach. We explore using multiple CMLLR and MLLR transforms as well.

6.4.1 Experimental Results

In the next series of experiments, we focus on the exploration of transform combination strategies only, thus using the best performing (C)MLLR-SVM systems observed in Sections 6.2.4.2. We use PLP15 features, PLP15N SAT acoustic models and multiple transforms. The rest of the system configuration is kept as in previous systems of this chapter, using 50 dimensions for the NAP session subspace with compensation applied after concatenation of individual transform supervectors. These are then scaled and classified using a SVM.

For convenience, we use CMH15 system names to denote that both CMLLR and MLLR transform coefficients are used. The amount of transforms is now given by pairs indicating number of CMLLR and MLLR transforms respectively, e.g. 1t/2t for one CMLLR and two MLLR transforms. Table 6.5 shows three blocks of results corresponding to best MLLR-SVM systems in Section 6.2.4.2, CMH15 SAT$_m$ systems using matched cepstra and models, and CMH15 SAT systems using unmatched cepstra and models.

At first sight, results are rather dependent on the type of system. CMH15 SAT$_m$ systems lose about 10% MDC and EER performance considering all three systems with respect to baseline MH15 SAT systems. This is clearly visible in the constellation plots of Figures 6.12(a) and 6.12(b), where we can identify symbols $\star$ in the top-right areas only. As already pointed out, transforming cepstra to match the SAT models prevents MLLR transforms from capturing a great deal of speaker variability. This residual variability is captured, though, in the CMLLR transform coefficients which are also used as features. The loss of performance can be explained by assuming that multiple MLLR transforms are able to capture more speaker variability than a single CMLLR transform, which is reasonable if we account for previous experimental results as well as the lines suggested in Section 6.3. Note that using multiple transforms in CMH15 SAT $_m$ systems obtains marginal gains in EER only for SRE 2006 and no other situations. This is somehow surprising since restricting MLLR to a regression class is expected to result in more precise coefficients for the corresponding class.

Regarding the rest of the systems, all using cepstra unmatched to the SAT models, we observe a rough trend for CMH15 SAT systems to outperform MH15 SAT counterparts, although very slightly and never systematically. Denoted as boldface in Table 6.5 and shown in the bottom-left corner of the constellation plots, CMLLR coefficients are used in systems exhibiting best MDC and EER for SRE 2005 but not necessarily for SRE 2006. Two out of the three best systems for both corpora are CMH15 SAT systems using two transforms at most, be it CMLLR, MLLR or both. Using one CMLLR transform in addition to MLLR transforms is only beneficial for CMH15 1t/1t SAT, obtaining relative gains of 5% MDC

	SRE 2005				SRE 2006			
	MDC		EER (%)		MDC		EER (%)	
System	F	FB	F	FB	F	FB	F	FB
MH15 1t SAT	.0198	.0180	4.86	4.57	.0183	.0164	4.32	4.31
MH15 2t SAT	.0180	.0159	4.53	4.33	.0155	.0143	**3.45**	3.57
MH15 3t SAT	.0183	.0165	4.91	4.82	.0155	**.0136**	3.77	3.68
CMH15 1t/1t SAT$_m$	.0211	.0187	5.03	4.99	.0172	.0161	4.32	4.18
CMH15 1t/2t SAT$_m$	.0214	.0199	5.08	5.03	.0181	.0159	4.23	4.23
CMH15 1t/3t SAT$_m$	.0223	.0200	5.41	5.24	.0181	.0162	4.23	4.12
CMH15 1t/1t SAT	.0188	.0167	4.66	4.67	.0166	.0156	4.27	3.90
CMH15 1t/2t SAT	.0170	**.0153**	**4.24**	**4.16**	.0173	.0160	3.81	3.81
CMH15 1t/3t SAT	.0191	.0174	4.62	4.70	.0174	.0153	3.77	3.71
CMH15 2t/2t SAT	**.0167**	.0155	4.53	4.49	**.0152**	.0140	3.58	3.53
CMH15 3t/3t SAT	.0216	.0189	4.94	4.79	.0160	.0146	3.58	**3.40**

Table 6.5 — MDC and EER of MH15 and CMH15 systems using CMLLR and MLLR transforms, PLP15N features and PLP15N SAT acoustic models for SRE 2005 and SRE 2006. All systems use NAP compensation of concatenated MLLR transform supervectors with a subspace dimension of 50. Columns F and FB show forward and averaged forward-backward scores respectively. The best scores of each column are shown in boldface.

and 7% EER for SRE 2005 and 9% MDC for SRE 2006. Using 2 CMLLR or MLLR transforms eventually results in the best systems for SRE 2005 and SRE 2006, although none in both corpora simultaneously. Systems using more transforms are well below these in terms of performance.

Forward-backward scoring obtains gains of 9% MDC and 1.5% EER considering the average for all systems. EER loss is occasionally observed for some of the systems, although never for both corpora at the same time.

6.5 Fusion with (C)MLLR-SVM Systems

In this section we explore score-level combination of CMLLR-SVM and MLLR-SVM systems together with the state-of-the-art systems described in Chapter 4. We use the logistic regression model of Equation 1.26 trained using SRE 2005 data. SRE 2006 is kept to test the fusion model, thus measuring performance of fused systems on the latter only. Logistic regression applies a scaling and an offset to each of the system scores which, after application of the logistic function, results in calibrated log-likelihood ratios as output scores as well. We used the FoCal toolkit[11] for this purpose.

6.5.1 Experimental Results

Fused-system experiments focus on combination of scores from baseline systems PLP-GMM, PLP-SVM and GSV-SVM with best performing of (C)MLLR-SVM systems using

[11] FoCal Toolkit, `http://www.dsp.sun.ac.za/~nbrummer/focal/index.htm`

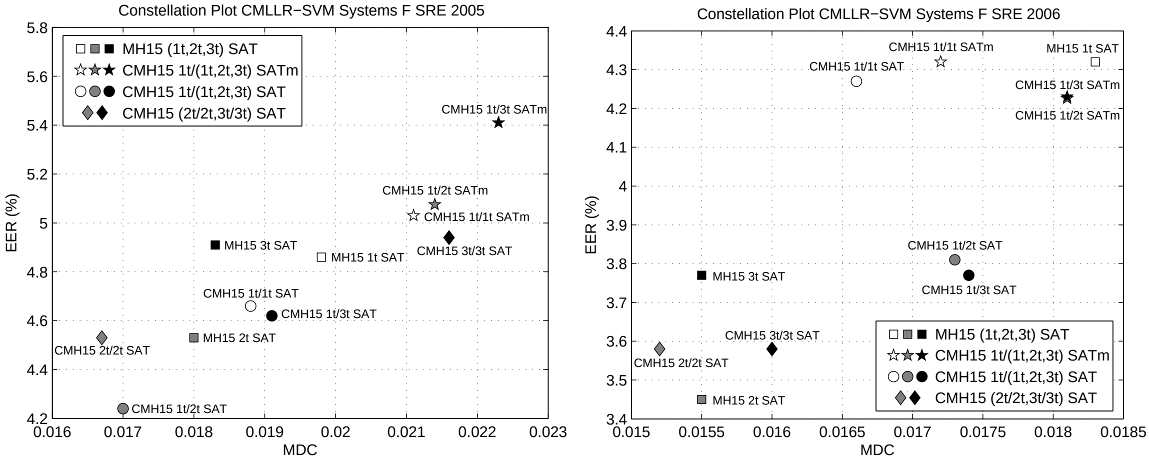

Figure 6.12 — Constellation plots of forward-scoring CMLLR-SVM systems using CMLLR and/or MLLR transform coefficients, PLP15N features and PLP15N acoustic models and different number of transforms for SRE 2005 (left, a) and SRE 2006 (right, b). Symbol shapes indicate type of combination of CMLLR and MLLR coefficients. Gray levels indicate the number of transforms involved, the more transforms used the darker.

either MLLR, CMLLR or both transform coefficients, i.e. MH15 2t SAT, CH15 2t SAT and CMH15 1t/2t SAT respectively. These are shown in the two first blocks of results of Table 6.6.

(C)MLLR-SVM individual systems obtain performance comparable to baseline systems in general. Constellation plots for SRE 2005 and SRE 2006 are shown in Figures 6.13(a) and 6.13(b). CH15 2t SAT and PLP-GMM systems lie in the top-right, less performing, area of both plots, with CH15 2t SAT considerably getting worse with respect to PLP-GMM for SRE 2006. Note that, although not shown in these plots, PLP-GMM obtains very important gains when using forward-backward scoring, whereas CH15 2t SAT stays about the same. MH15 2t SAT and CMH15 1t/2t SAT systems lie in the bottom-left area outperforming PLP-SVM and GSV-SVM, the latter in either MDC or EER but not simultaneously. CMH15 1t/2t SAT is the most performing system for SRE 2005 and MH15 2t SAT is for SRE 2006, as observed when comparing boldface numbers of first and second blocks of Table 6.6. DET curves of individual systems are shown in Figures 6.14(a) and 6.14(b). For SRE 2005, CMH15 1t/2t outperforms all systems at all operating points while MH15 2t SAT and GSV-SVM curves present a strong overlap. For SRE 2006, MH15 2t SAT is the most performing for low false-alarm rates while GSV-SVM is for high false-alarm rates.

Regarding fused systems, we explore the combination of one baseline system, either PLP-GMM or GSV-SVM, together with (C)MLLR-SVM systems first. PLP-GMM systems gain over 40% MDC and EER by combining with MH15 2t SAT (a+d), becoming slightly lower for the other (C)MLLR-SVM systems (a)+(e) and (a)+(f). GSV-SVM systems obtain considerable improvements too, around 30% MDC and EER by combining with MH15 2t SAT or CMH15 2t SAT. Fusion with CH15 2t SAT is beneficial as well, although gains are in the range 15%-30% for both MDC and EER. We have plotted these fused system along with the individual systems in the constellation plot of Figure 6.15(a). Systems including MH15 2t SAT (d) in the fusion are found in the bottom-left corner, followed by systems using CMH15 2t SAT (f) and then CH15 2t SAT (e). This suggests that MLLR transforms are

	SRE 2005				**SRE 2006**			
	MDC		EER (%)		MDC		EER (%)	
System	F	FB	F	FB	F	FB	F	FB
PLP-GMM (a)	.0287	.0202	5.82	4.74	.0218	.0177	4.69	**3.72**
PLP-SVM (b)	.0211	.0204	4.82	4.48	.0198	.0189	4.41	4.14
GSV-SVM (c)	**.0177**	**.0172**	**4.66**	**4.45**	**.0182**	**.0174**	**3.54**	3.26
MH15 2t SAT (d)	.0180	.0159	4.53	4.33	**.0155**	**.0143**	**3.45**	**3.57**
CH15 2t SAT (e)	.0258	.0228	6.19	6.24	.0235	.0217	5.37	5.37
CMH15 1t/2t SAT (f)	**.0170**	**.0153**	**4.24**	**4.16**	.0173	.0160	3.81	3.81
(a)+(d)	–	–	–	–	.0125	.0115	**2.30**	2.49
(a)+(e)	–	–	–	–	.0164	.0151	3.21	3.09
(a)+(f)	–	–	–	–	.0143	.0126	2.67	2.72
(c)+(d)	–	–	–	–	**.0117**	**.0112**	2.34	**2.43**
(c)+(e)	–	–	–	–	.0150	.0148	2.94	2.88
(c)+(f)	–	–	–	–	.0134	.0129	2.52	2.48
(a)+(b)+(c)	–	–	–	–	.0149	.0147	3.13	3.12
(a)+(b)+(c)+(d)	–	–	–	–	**.0114**	**.0114**	**2.25**	2.57
(a)+(b)+(c)+(e)	–	–	–	–	.0142	.0138	2.71	2.84
(a)+(b)+(c)+(f)	–	–	–	–	.0131	.0127	2.39	2.62
All	–	–	–	–	.0120	.0117	2.34	**2.57**

Table 6.6 — MDC and EER of baseline and fused systems involving MLLR-SVM for SRE 2005 and SRE 2006. Columns F and FB show forward and averaged forward-backward scores respectively, which also applies to system combination. The best scores in each column and block are shown in boldface.

preferable over CMLLR for system combination too besides performing well in a MLLR-SVM system alone. DET curves of Figures 6.16(a) 6.16(b) confirm this trend for virtually all operating points. Note that a large improvement is consistently obtained even with CH15 2t SAT, a system obtaining the highest error rates. This is probably explained by the fact that (C)MLLR transform coefficients involve complementary information not used in either PLP-GMM or GSV-SVM.

Fusing more than two systems results in rather slight further improvement. All fused systems are shown together in the constellation plot of Figure 6.15(b). Gray and black dots are spread across the plot, resulting in a large overlap of 2-system fusion and 3,4,5-system fusion areas. The baseline combination, i.e. (a)+(b)+(c), consisting of the three state-of-the-art systems described in Chapter 4 has become one of the less performing fused systems now. Note that MH15 2t SAT (d) still has a unique role as all four systems including it in the fusion are clustered in the bottom-left, most performing, corner of the plot. Interestingly, this is not the case for CMH15 1t/2t (f) SAT which achieves similar performance alone but does not combine as well. Figures 6.17(a) and 6.17(b) show DET curves of the baseline individual systems, the baseline combination system (a)+(b)+(c) and three fused systems involving MH15 2t SAT (d) using forward and forward-backward scoring respectively. The four top-left curves correspond to systems not using MLLR-SVM with the baseline combination systems effectively outperforming all baseline individual systems. As MH15 2t SAT (d) is introduced in a fused system a considerable improvement of performance is observed for both forward and forward-backward scoring, resulting in almost overlapped curves.

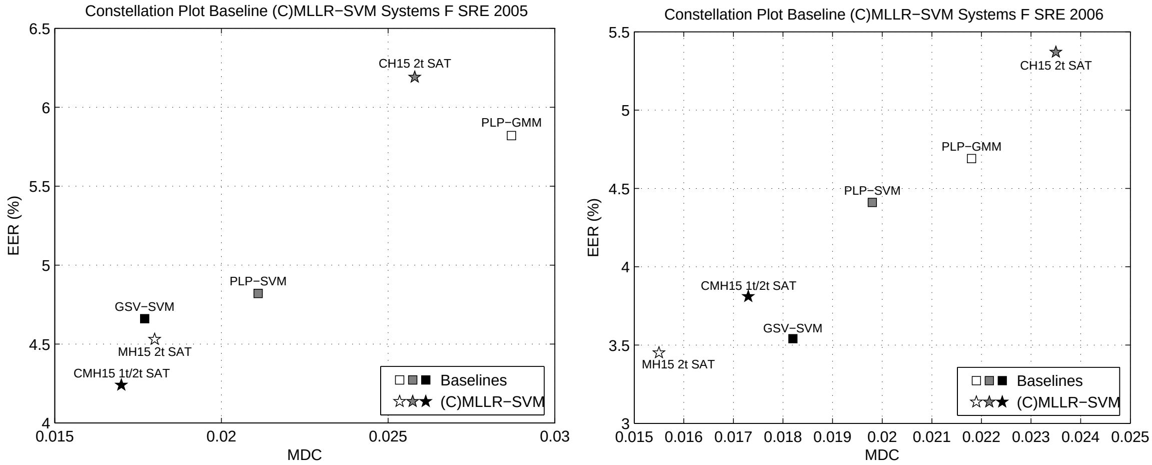

Figure 6.13 — Constellation plots of forward-scoring baseline and (C)MLLR-SVM systems for SRE 2005 (left, a) and SRE 2006 (right, b). Symbols ■ and ⋆ indicate baseline and (C)MLLR-SVM systems respectively. Gray levels indicate variations within system category.

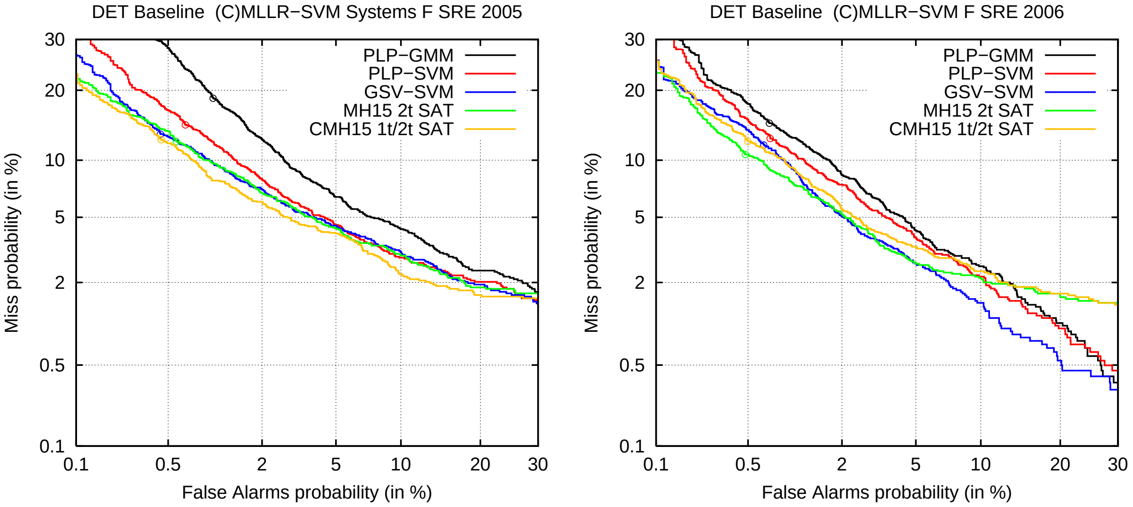

Figure 6.14 — DET curves of baseline and (C)MLLR-SVM systems for SRE 2005 (left,a) and SRE 2006 (right,b).

6.6 Summary and Conclusion

In this chapter we have explored the use of multi-class CMLLR and MLLR transform coefficients as features in (C)MLLR-SVM systems. The large amount of experiments conducted has allowed to gain insight into the sources affecting performance of these systems. CMLLR-SVM using a single transform and a GMM as acoustic model has shown to be a simple yet effective alternative to LVCSR-based systems when using a front-end optimized for speaker recognition purposes and SAT. LVCSR-based (C)MLLR-SVM systems typically obtain improved performance at the expense of the considerable cost of LVCSR

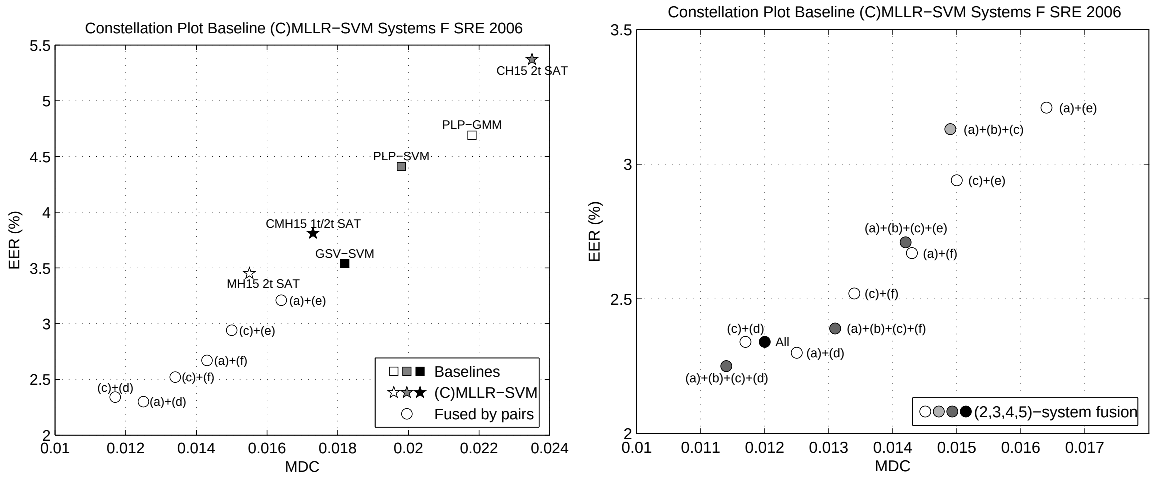

Figure 6.15 — Constellation plots of forward-scoring (C)MLLR-SVM fused systems for SRE 2006. (left, a) shows 2-system fusion and individual systems. (right, b) shows all fused systems alone.

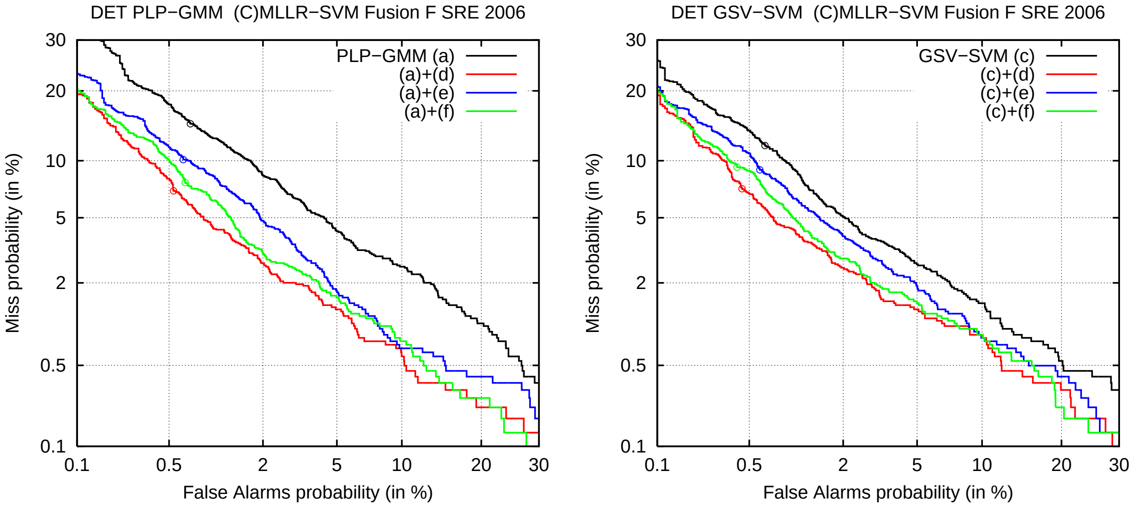

Figure 6.16 — DET curves of (C)MLLR-SVM systems combined with PLP-GMM (left,a) and GSV-SVM (right,b) for SRE 2006.

acoustic model training and of its dependence on transcripts. Using a front-end optimized for speaker recognition together with LVCSR acoustic models has shown to result in large performance improvement, even larger if speaker adaptive training is used during training. The amount of regression classes used for computing (C)MLLR transforms can result in considerable gains, although a performance loss has been found for two transforms. Forward-backward score combination systematically obtains further gains for individual (C)MLLR-SVM systems but it reduces considerably for fused systems.

Experimental results have shown that CMLLR and MLLR transform coefficients behave quite differently when classified using SVM, the latter eventually resulting in considerably

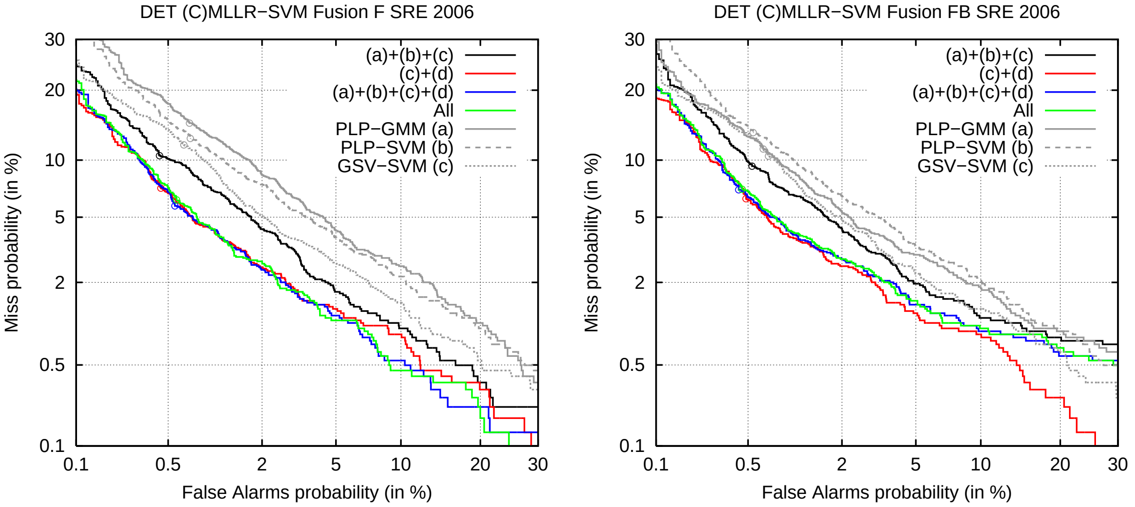

Figure 6.17 — DET curves of the baseline individual and fused systems and fused systems including MH15 2t SAT (d) using forward scoring (left,a) and forward-backward scoring (right,b) for SRE 2006.

more performing systems. As a global trend, CMLLR transform coefficients perform better in low-complexity systems whereas MLLR coefficients do in systems using more sophisticated speaker adaptation schemes. Non-singularity of CMLLR transform matrices could be one point explaining the observed difference of performance. Avoiding small singular values would prevent CMLLR transforms from properly capturing correlation of cepstra otherwise necessary for proper adaptation. CMLLR and MLLR transform matrices were found to considerably differ in terms of their non-orthogonality measures as well as singular value spread and variance.

We also explored the use of CMLLR and MLLR transform coefficients together in a (C)MLLR-SVM system. We found that in systems using SAT, matching cepstra to the SAT acoustic models was not beneficial compared to computing transforms directly on speaker-dependent cepstra. Using multiple CMLLR and MLLR transforms together did not bring clear improvements versus standard multi-class MLLR-SVM.

(C)MLLR-SVM systems outperform all three baseline systems for both SRE 2005 and SRE 2006. Furthermore, combining the baseline systems with one or more (C)MLLR-SVM by fusing their scores results in a large improvement of performance. MLLR-SVM was found to dominate score fusion, as two-system fusion obtained similar performance to three-, four- and five-system fusion.

Chapter 7

Lattice MLLR-SVM Systems

In Chapter 6 we performed a comprehensive study of multi-class (C)MLLR-SVM systems. Our implementation showed to be high performing, even more than other state-of-the-art acoustic systems. (C)MLLR-SVM using the acoustic models of a LVCSR system relies on available transcripts to obtain the actual features used for classification by means of supervised MLLR adaptation. Several sources of variability such as accent, speaking style or acoustic conditions can considerably degrade transcript quality, correspondingly affecting classification performance.

In this chapter, we propose the use of lattice-based MLLR, embedding word-lattices into MLLR estimation, to mitigate the effect of transcription errors on transform coefficients. The chapter is organized as follows: Section 7.2 introduces lattice-based MLLR adaptation and compares it to standard MLLR. Section 7.3 describes lattice-based MLLR-SVM systems and gives experimental results on the NIST SRE 2005 and SRE 2006 campaigns. Section 7.4 provides results for systems fused with lattice-based MLLR-SVM systems.

7.1 Transcription Errors in MLLR-SVM Systems

Supervised MLLR adaptation as used in MLLR-SVM systems of Chapter 6 assumes that the transcripts used contain no errors. Although we used manual transcripts for impostor speaker segments taken from the Switchboard I corpus, transcripts provided for most of the data, including NIST SRE 2004, SRE 2005 and SRE 2006 corpora, are actually hypotheses derived using a speech recognition system. Performance of such a system is not publicly available to our knowledge, but we can expect around 20% Word Error Rate (WER), i.e. 1 erroneous word out of 5 in average, for state-of-the-art LVCSR systems dealing with spontaneous conversational telephone speech.

Back to speaker recognition, MLLR-SVM systems use these transcripts as reference to constrain the triphone models used for alignment and estimation of MLLR transforms. Although the impact of transcription errors on system performance is not easily quantifiable in our situation, wrong models are likely to be chosen when these errors occur. LVCSR systems mainly rely on language model probabilities to decode hypotheses for a whole sequence of observation vectors. Given that language and acoustic models use distinct criteria to obtain the probabilities used during decoding, an erroneous word introduced by the language model can result in arbitrarily different acoustic models. The minimum unit hypothesized by the system is one word, thus forcing a full-word acoustic model at the least, even for partially-uttered or very short words. In a similar line, out-of-vocabulary (OOV) words not considered by the LVCSR system can result in random word models if not properly detected. We assume that the impact of choosing wrong models is observable in the MLLR regression coefficients obtained based on them and, therefore, in the classification performance obtained by MLLR-SVM systems as well.

In this chapter we introduce the use of lattice-based MLLR adaptation as a means of reducing the impact of transcription errors in the transform coefficients used for SVM classification.

7.2 Lattice-based MLLR

As discussed in Section 2.3.3, MLLR adapts mean vectors $\boldsymbol{\mu}$ of a HMM via the transform $\hat{\boldsymbol{\mu}} = \mathbf{A}\boldsymbol{\mu} + \mathbf{b}$ such that the likelihood of the adapted model is maximized. Assuming T feature vectors as adaptation data $\mathbf{X} = (\mathbf{x_1}, \ldots, \mathbf{x_T})$, the MLLR criterion can be compactly written as

$$\hat{\mathbf{\Theta}}^* = \arg\max_{\hat{\mathbf{\Theta}}} \log p(\mathbf{X}|\hat{\mathbf{\Theta}}) \tag{7.1}$$

where $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\mu_1}}, \mathbf{\Sigma_2}, \ldots, \hat{\boldsymbol{\mu_R}}, \mathbf{\Sigma_R})$ are mean and covariance parameters of the adapted model[1] for each of the R Gaussians considered in the regression problem. When using the acoustic models of a LVCSR system, the Gaussian components used for regression are chosen by first decomposing the transcripts into phones and then triphones. In a second stage, the triphones corresponding to the regression class of interest are taken, by static assignment or using a decision tree. In the expectation step of the EM algorithm used to solve Equation 7.1 the adaptation data $\mathbf{X}$ is aligned against the Gaussian components of the speaker-independent models $\mathbf{\Theta}$ obtaining $p(i|\mathbf{x_t})$ for each feature vector at time t and Gaussian i for each triphone in the HMM. These probabilities are then used in the maximization step to compute the regression parameters $\mathbf{A}$ and $\mathbf{b}$ by solving Equation 2.24.

In the process above only one transcript involving a single word-level alignment, e.g. $\mathbf{s_{1-best}} = (\omega_1, \ldots, \omega_N)$ for the 1-best hypothesis obtained by a speech recognition system, has been considered. In situations where the correct transcript is not known, we can introduce such an uncertainty in the MLLR formulation of Equation 7.1 by using the term $p(\mathrm{s}|\mathbf{X}, \mathbf{\Theta})$. This term designates the probability of obtaining word-alignment s given the adaptation data and the non-adapted model, and it allows conditioning of the log-likelihood as

$$\hat{\mathbf{\Theta}}^* = \arg\max_{\hat{\mathbf{\Theta}}} \sum_{\mathbf{s}\in\mathbf{S}} p(\mathbf{s}|\mathbf{X}, \mathbf{\Theta}) \log p(\mathbf{X}, \mathrm{s}|\hat{\mathbf{\Theta}}) \tag{7.2}$$

For standard MLLR this reduces to Equation 7.1, since $p(\mathrm{s}|\mathbf{X}, \mathbf{\Theta}) = 1$ for the word sequence in the transcript or 1-best hypothesis used and $p(\mathrm{s}|\mathbf{X}, \mathbf{\Theta}) = 0$ for all other possible word sequences in $\mathbf{S}$. However, the interest of such MLLR reformulation is that it can account for a large amount of word sequence in the estimation process.

MLLR estimation can involve multiple word sequences when considering several pronunciation variants per word in the lexicon. These variants are included as alternative arcs

[1] Note that covariance matrices $\mathbf{\Sigma}$ are not adapted.

with fixed probabilities in the transcript-conditioned HMM used to obtain the Gaussian-level alignment $p(i|\mathbf{x_t})$. Once these probabilities have been obtained, MLLR transforms are estimated in the standard way by solving Equation 2.24. Equation 7.2 is solved in an analogous way, with the difference that, besides pronunciation variants, different words are used indeed. LVCSR systems can output so-called word lattices that encode a set of word sequence hypotheses, generally more plausible hypotheses, as an acyclic graph structure. This variant of MLLR computing transforms using alignments derived from word lattices is called lattice-based MLLR (LMLLR) [Padmanabhan *et al.*, 2000]. We refer the reader to [Hetherington *et al.*, 1993; Murveit *et al.*, 1993; Ortmanns *et al.*, 1997] to gain insight on the construction of word lattices.

Considering multiple hypotheses for MLLR estimation can result in more robust transform coefficients. Assuming errors occurred during decoding of the most likely hypothesis, it is probable that one of the other hypotheses is correct, thus involving the correct acoustic models. However, assuming the most likely decoded hypothesis contains no errors, we are including erroneous models in the estimation. Therefore, robustness of lattice-based MLLR transform estimates is compromised by the quality of decoded hypotheses. For spontaneous conversational telephone speech, involving poor articulation, frequent disfluencies and fast variations of speech rate, we can assume current state-of-the-art systems to obtain considerable Word Error Rate (WER) around 15%-20%. In this context, we can expect word lattices to improve estimation over standard MLLR. Note, however, that a speech recognition system is required to generate word lattices, preventing lattice-based MLLR from taking advantage of high-quality transcripts obtained by human intervention. For these data, standard MLLR is probably better suited.

7.3 Lattice MLLR-SVM System Description

Lattice MLLR-SVM, otherwise LMLLR-SVM, systems are analogous to MLLR-SVM systems of Chapter 6 except that they use lattice-based MLLR transform coefficients as features instead of those obtained using standard MLLR. We focus on those setups that obtained best performance in the previous chapter, i.e multi-class systems using PLP15N features and PLP15N SAT acoustic models, with SAT used during training only. NAP is applied to reduce inter-session variability of LMLLR transform supervectors using 50 dimensions for the session subspace. We classify these supervectors using SVM with prior min-max scaling as described in Section 4.1.3.

MLLR transform coefficients for each regression class are computed using lattice-based MLLR adaptation as described in Section 7.2, thus requiring generation of word lattices for each speech segment in target, test or impostor speaker data. Several lattices per speech segment are generated using a speech recognition system based on the LIMSI CTS system [Gauvain *et al.*, 2003]. Word recognition is performed in two decoding passes using different acoustic models, both continuous density HMM, and language models, using N-gram statistics estimated on large corpora of text data. The front-end uses PLP12 features, described in Section 6.2.1, which have been warped using Vocal Tract Length Normalization (VTLN) with single-Gaussian gender-dependent GMM and ML estimation of the warping factor. Features are further transformed via Maximum Likelihood Linear Transform (MLLT) to minimize the likelihood loss resulting from the use of diagonal covariance Gaussian mixtures in the models. Acoustic models are tied-state context-dependent left-to-right HMM based on a 48-symbol phone set with 32-Gaussian mixture observation densities and are trained using Maximum Mutual Information Estimation (MMIE). The first recognition pass uses about 6500 tied-states along with a bi-gram language model to generate word

Corpus	#Lattices	#Nodes	#Arcs	Duration	#Arcs/second
Switchboard I	58.8	360.0	739.7	5.3 s	125.8
SRE 2004 Train	42.3	648.5	1532.0	4.6 s	307.7
SRE 2005 Train	40.9	452.7	1027.4	4.4 s	216.7
SRE 2005 Test	39.2	455.0	1022.1	4.3 s	235.2
SRE 2006 Train	45.0	655.3	1576.6	3.7 s	401.2
SRE 2006 Test	44.7	595.9	1421.4	3.7 s	361.6
All corpora	**45.1**	**527.9**	**1219.8**	**4.3** s	**274.7**

Table 7.1 — Statistics of lattice generation for speech segments in Switchboard I and NIST SRE corpora. Columns show the average number of lattices (and utterances) per segment and average number of nodes, arcs, duration (in seconds) and arc density (in arcs per second) per lattice.

lattices used to adapt the more detailed acoustic models of the second pass, with about 38000 tied-states, via single-class CMLLR and multiple-class MLLR adaptation. The second recognition pass generates more detailed word lattices using a tri-phone language model. First and second passes consider 10000 and 50000 hypotheses for decoding respectively and a beam width of 8 is used to prune non-promising paths during lattice generation. The final word lattices are obtained by rescoring second-pass lattices using a four-gram language model. These lattices are then used to compute lattice-based MLLR transform coefficients using PLP15N SAT acoustic models for fair comparison against those used in MH15 SAT systems. We do not consider those hypotheses with posterior probabilities below 0.01 in this step.

Table 7.1 provides basic statistics of the lattices generated for all corpora involved in our systems. The number of lattices corresponds to the number of utterances in a speech segments. The given values vary depending on the corpus, specially for Switchboard I. A larger average number of utterances and average utterance duration were obtained for this corpus with respect to NIST SRE corpora. The considerably smaller arc density observed in Switchboard I might reflect easier decoding due to significantly lower channel variability, given that it includes landline telephone data only. A similar effect is observed when comparing lattice arc density for SRE 2005 versus SRE 2006 for the same reason.

7.3.1 Experimental Results

We explore LMLLR-SVM systems using 1t, 2t and 3t regression classes and their MLLR-SVM counterparts. For the latter we use both transcripts provided by NIST as well as 1-best hypotheses obtained from the same lattices used for LMLLR-SVM. This makes both approaches fully comparable and the effect of considering multiple hypotheses in MLLR estimation can be properly assessed. LMLLR-SVM systems use acronyms LMH15 SAT whereas we use MH15 SAT 1-best and MH15 SAT for systems using 1-best hypotheses and NIST transcripts respectively.

Table 7.2 shows results for these systems in three blocks, the first corresponding to systems using NIST transcripts and the rest corresponding to systems using the LIMSI CTS speech recognizer to generate transcripts and lattices. MH15 SAT systems using NIST transcripts obtain the lowest performance overall as seen by comparing boldface numbers across blocks of results. The constellation plots of Figures 7.1(a) and 7.1(b) show MH15 2t

	SRE 2005				**SRE 2006**			
	MDC		EER (%)		MDC		EER (%)	
System	F	FB	F	FB	F	FB	F	FB
MH15 1t SAT	.0198	.0180	4.86	4.57	.0183	.0164	4.32	4.31
MH15 2t SAT	**.0180**	**.0159**	**4.53**	**4.33**	**.0155**	.0143	**3.45**	**3.57**
MH15 3t SAT	.0183	.0165	4.91	4.82	**.0155**	**.0136**	3.77	3.68
MH15 1t SAT 1-best	.0205	.0181	5.21	5.27	.0209	.0190	4.59	4.45
MH15 2t SAT 1-best	.0193	.0169	4.91	4.95	.0180	.0167	3.95	3.63
MH15 3t SAT 1-best	.0197	.0170	4.92	5.07	.0179	.0162	4.36	4.18
LMH15 1t SAT	.0204	.0178	4.78	4.82	.0200	.0174	4.45	4.23
LMH15 2t SAT	**.0183**	.0163	**4.70**	**4.62**	.0166	**.0152**	**3.67**	**3.63**
LMH15 3t SAT	.0184	**.0153**	4.78	4.66	**.0165**	.0154	3.95	3.68

Table 7.2 — MDC and EER of (L)MLLR-SVM systems using multiple one, two and three transforms, PLP15N features and PLP15N SAT acoustic models for SRE 2005 and SRE 2006. All systems use NAP compensation of concatenated MLLR transform supervectors with a subspace dimension of 50. Columns F and FB show forward and averaged forward-backward scores respectively. The best scores for system using NIST transcripts and LIMSI CTS hypotheses are shown for each column in boldface.

SAT systems in the bottom-left, more performing, corner for both SRE 2005 and SRE 2006. In global terms, MH15 SAT 1-best systems gather in the top-right area whereas LMH15 SAT are in a south-west area overall, obtaining 6% MDC and EER average gains eventually reaching 8% MDC and 9% EER for SRE 2006. Such a trend is clearly visible for SRE 2006 where, for any fixed number of transforms, systems become more performing as we go through symbols ■, ⋆ and •. However, the quality of the hypotheses seems to be a more important factor driving system performance, given that MH15 SAT systems using transcripts provided by NIST lie far away from their MH15 SAT 1-best counterparts using the LIMSI CTS system. Figures 7.2(a) and 7.2(b) show DET curves for 2t systems for SRE 2005 and SRE 2006. LMH15 2t SAT obtains improvements over MH15 SAT 1-best at almost all operating points in the plot whereas it rarely outperforms MH15 2t SAT.

Regarding the number of transforms LMLLR-SVM systems follow the same trend observed in MLLR-SVM systems. The constellation plot of Figures 7.1(a) and 7.1(b) show that using two transforms obtain optimal MDC and EER values regardless of whether lattices or transcripts are used. Passing from one to two transforms results in 10% MDC and EER relative gain for systems using the LIMSI CTS system compared to 12% for systems using NIST transcripts. Adding a third transform results in loss of performance for all systems, particularly for those using standard MLLR. MH15 3t SAT and MH15 3t SAT 1-best lose about 8% and 3.5% versus their respective 2t systems when averaging all MDC and EER measures and corpora. Such loss is considerably smaller for LMH15 SAT systems, less than 1%. The use of multiple alignments in lattice-based MLLR could account for this effect. Allowing cepstral vectors to be assigned to several triphones simultaneously increases the effective number of frames assigned to each of them and, thus, the average amount of data assigned to each regression class as well. Reliability of MLLR transform estimates should be improved when not enough data are available for a regression class. Such a situation naturally occurs as we increase the number of regression classes and we keep the num-

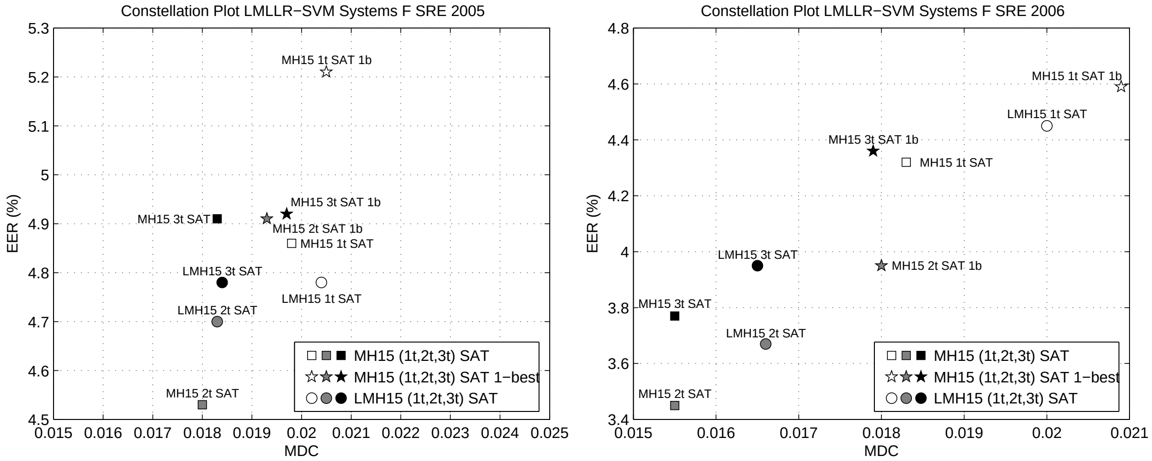

Figure 7.1 — Constellation plots of forward-scoring (L)MLLR-SVM systems using lattice-based MLLR and standard MLLR transform coefficients, PLP15N features, PLP15N acoustic models and one, two and three transforms for SRE 2005 (left, a) and SRE 2006 (right, b). Symbol shapes indicate the type of system. Gray levels indicate the number of transforms involved, the more transforms used the darker.

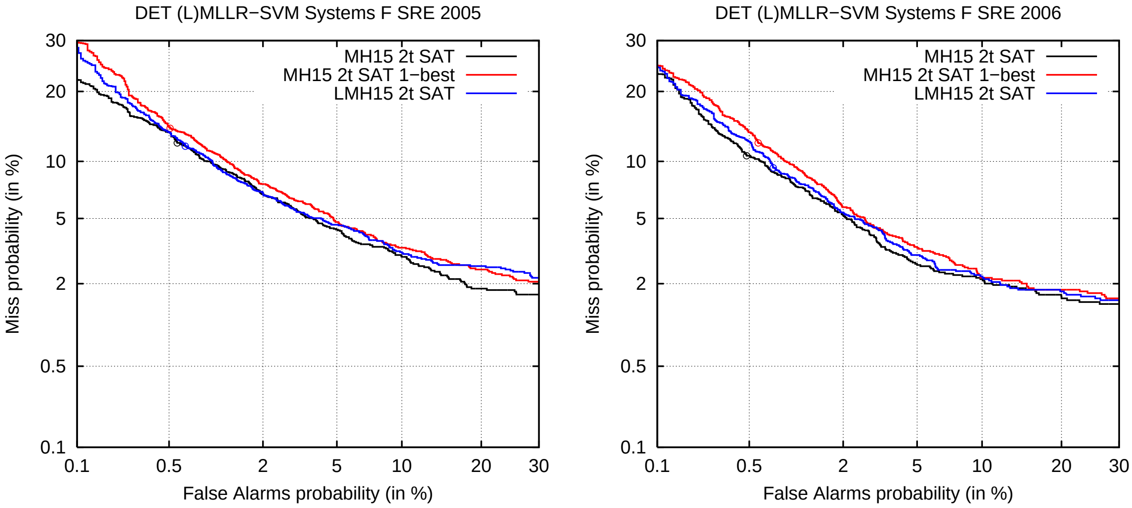

Figure 7.2 — DET curves of baseline and (L)MLLR-SVM systems using two transforms across different types of transcripts for SRE 2005 (left,a) and SRE 2006 (right,b).

ber of parameters fixed[2]. Comparing DET curves of LMH15 SAT systems of Figures 7.3(a) and 7.3(b) versus those of MH15 SAT systems of Figures 6.8(a) and 6.8(b) in the previous chapter, we can see that loss of 3t systems is slightly absorbed in LMH15 SAT systems, particularly for SRE 2005.

[2] We noticed a considerable rise of diagonal back-off rates when passing from 2t to 3t regression classes for MH15 SAT systems when using the threshold set for speech recognition use.

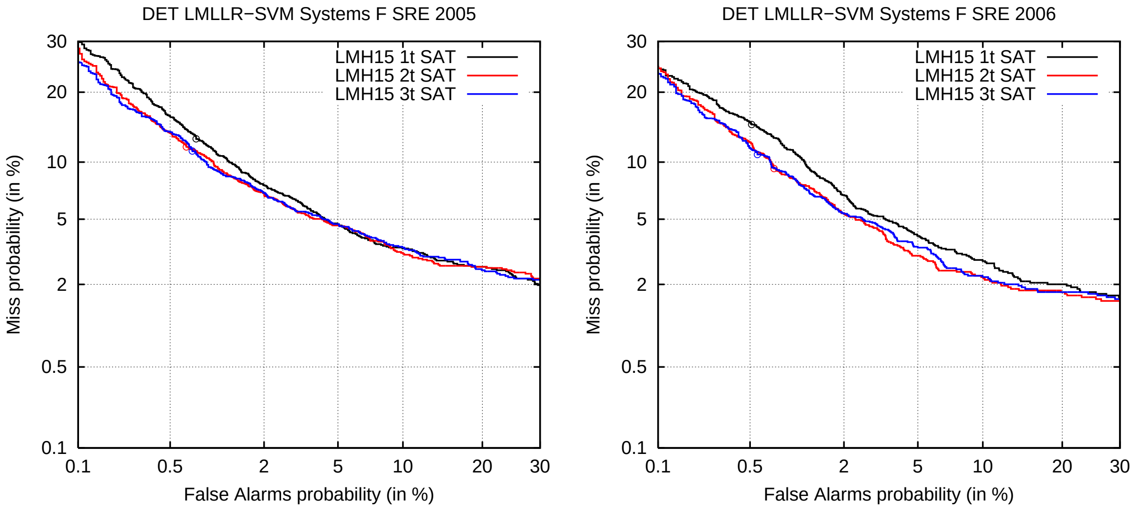

Figure 7.3 — DET curves of baseline and LMLLR-SVM systems using one, two and three transforms for SRE 2005 (left,a) and SRE 2006 (right,b).

Regarding the scoring approach, systems using forward-backward scoring obtain consistent gains over those using forward scoring only. These reach 7% in average for LMH15 SAT systems, only slightly larger than those obtained for MH15 SAT and MH15 SAT 1-best systems, about 6%.

7.4 Fusion with Lattice-based MLLR-SVM

In this section we perform a study of system combination for (L)MLLR-SVM systems similar to that of Section 6.5. We use the logistic regression model of Equation 1.26 trained using SRE 2005 score data. Fused scores are then obtained on SRE 2006 data only. Logistic regression applies a scaling and an offset to each of the system scores which, after application of the logistic function, results in calibrated log-likelihood ratios as output scores as before. We use the FoCal toolkit[3] for this purpose.

7.4.1 Experimental Results

We target score fusion of baseline systems PLP-GMM, PLP-SVM and GSV-SVM with (L)MLLR-SVM systems using two transforms, i.e. MH15 2t SAT, MH15 2t SAT 1-best and LMH15 2t SAT, using PLP15N features and PLP15N SAT acoustic models. Such setups obtain best absolute performance overall (see Tables 7.2 and 6.4). MDC and EER values for all individual and fused systems are shown in Table 7.3.

(L)MLLR-SVM individual systems obtain performance similar to that obtained baseline systems, namely GSV-SVM and PLP-SVM. Figures 7.4(a) and 7.4(a) show constellation plots of all considered individual systems for SRE 2005 and SRE 2006. All (L)MLLR-SVM systems lie in the most performing area, with GSV-SVM being the closest baseline system to them for both corpora and PLP-GMM the furthest one. However, note that these plots

[3] FoCal Toolkit, `http://www.dsp.sun.ac.za/~nbrummer/focal/index.htm`

	SRE 2005				SRE 2006			
	MDC		EER (%)		MDC		EER (%)	
System	F	FB	F	FB	F	FB	F	FB
PLP-GMM (a)	.0287	.0202	5.82	4.74	.0218	.0177	4.69	**3.72**
PLP-SVM (b)	.0211	.0204	4.82	4.48	.0198	.0189	4.41	4.14
GSV-SVM (c)	**.0177**	**.0172**	**4.66**	**4.45**	**.0182**	**.0174**	**3.54**	3.26
MH15 2t SAT (d)	**.0180**	**.0159**	**4.53**	**4.33**	**.0155**	**.0143**	**3.45**	**3.57**
MH15 2t SAT 1-best (e)	.0193	.0169	4.91	4.95	.0180	.0167	3.95	3.63
LMH15 2t SAT (f)	.0183	.0163	4.70	4.62	.0166	.0152	3.67	3.63
(a)+(d)	–	–	–	–	.0125	.0115	**2.30**	2.49
(a)+(e)	–	–	–	–	.0138	.0125	2.62	2.56
(a)+(f)	–	–	–	–	.0130	.0120	2.48	2.53
(c)+(d)	–	–	–	–	**.0117**	**.0112**	2.34	**2.43**
(c)+(e)	–	–	–	–	.0127	.0125	2.48	2.52
(c)+(f)	–	–	–	–	.0129	.0120	2.48	**2.43**
(a)+(b)+(c)	–	–	–	–	.0149	.0147	3.13	3.12
(a)+(b)+(c)+(d)	–	–	–	–	.0114	.0114	2.25	2.57
(a)+(b)+(c)+(e)	–	–	–	–	.0126	.0123	2.30	2.53
(a)+(b)+(c)+(f)	–	–	–	–	.0123	.0121	2.25	2.48
All	–	–	–	–	**.0113**	**.0113**	**2.20**	**2.34**

Table 7.3 — MDC and EER of individual and fused systems involving LMLLR-SVM and MLLR-SVM for SRE 2005 and SRE 2006. Columns F and FB show forward and averaged forward-backward scores respectively, which also applies to system combination. The best scores in each column and block are shown in boldface.

do not reflect the large gain obtained by PLP-GMM, which brings it closer to other systems when using forward-backward scoring. For SRE 2005, systems using the LIMSI CTS system are about 5% behind GSV-SVM, the most performing baseline system, in both MDC and EER terms. For SRE 2006, LMH15 2t SAT obtains over 10% MDC gains over GSV-SVM but loses about 7% in EER. Figures 7.5(a) and 7.5(b) show DET curves for individual systems. Curves of (L)MLLR-SVM systems are skewed with respect to those of baseline individual systems. For SRE 2005, MH15 2t SAT outperforms the other systems at most operating points, whereas GSV-SVM does for MH15 2t SAT 2-best and LMH15 2t SAT. For SRE 2006, LMH15 2t SAT lies between MH15 2t SAT and GSV-SVM for low false-alarm rates. All (L)MLLR-SVM systems lose considerable performance for high false-alarm rates in this corpora.

For two-system combination, we explore fusion of either PLP-GMM and GSV-SVM with one (L)MLLR-SVM system. Such schemes provide already considerable performance improvement for a minimum amount of systems involved. Based on PLP-GMM and GSV-SVM as baseline systems we obtain average gains around 38% and 30% MDC and EER respectively, becoming maximal when MH15 2t SAT is included and slightly lower for LMH15 2t SAT. DET curves of Figures 7.6(a) and 7.6(b) confirm such trend at all operating points. Figure 7.7(a) shows a constellation plot of individual and two-system fusion. Two clusters are clearly distinguishable, one corresponding to fused systems in the bottom-left corner and one for individual systems in the top-right area. Dispersion of fused systems is

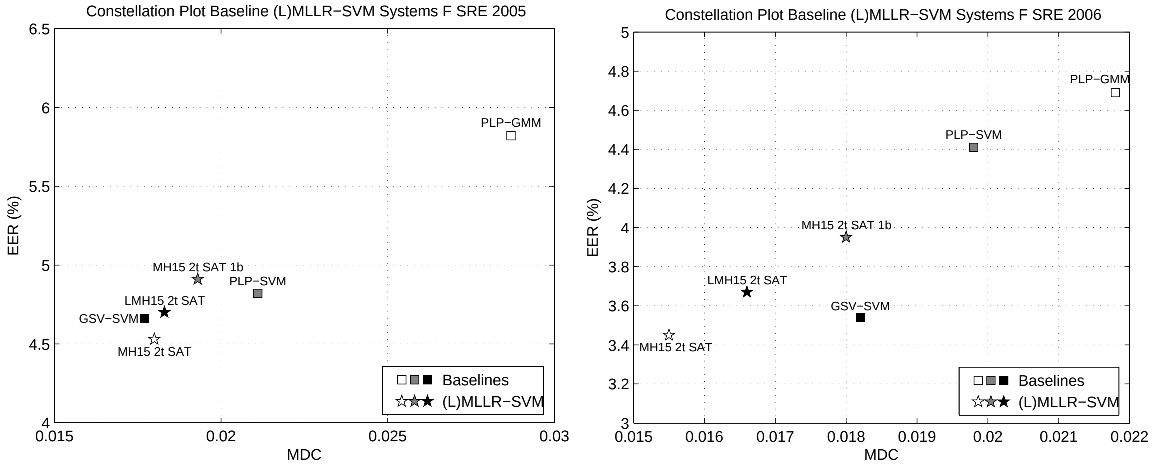

Figure 7.4 — Constellation plots of forward-scoring baseline and (L)MLLR-SVM systems for SRE 2005 (left, a) and SRE 2006 (right, b). Symbols ■ and ⋆ indicate baseline and (C)MLLR-SVM systems respectively. Gray levels indicate variations within system category.

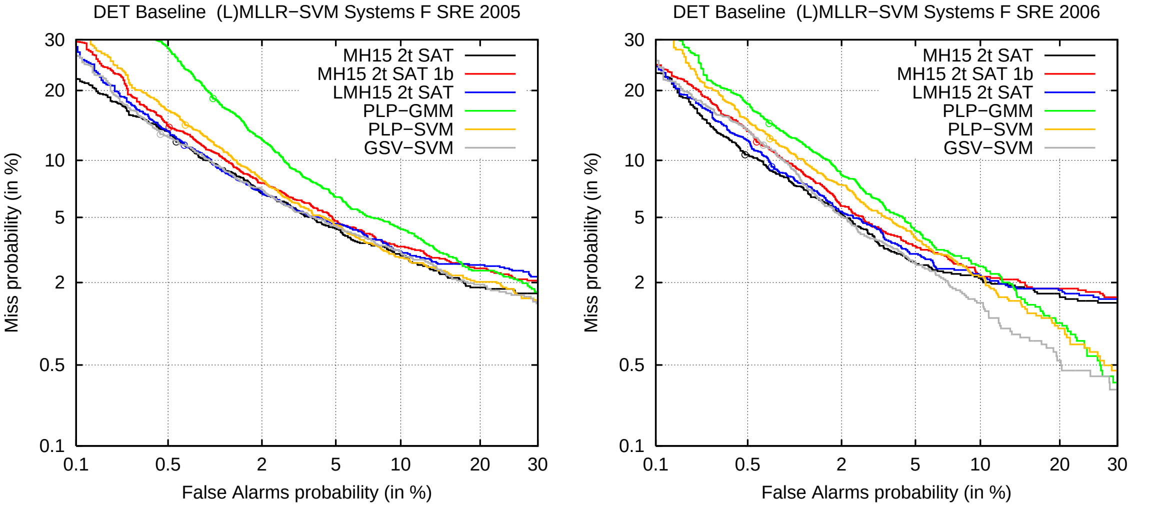

Figure 7.5 — DET curves of baseline and (L)MLLR-SVM systems for SRE 2005 (left,a) and SRE 2006 (right,b).

visibly lower than for individual systems.

A similar trend is observed when combining more than two systems. As found in Chapter 6, including MH15 2t SAT in the baseline combination, i.e. (a)+(b)+(c)+(d), dominates performance of four-system fusion. When forward scoring and the LIMSI CTS system are used, i.e. (a)+(b)+(c)+(e) and (a)+(b)+(c)+(f) systems, performance is a step below. However, forward-backward scoring seems to get performance of these three systems together, with (a)+(b)+(c)+(f) eventually outperforming (a)+(b)+(c)+(d) in EER terms. All fused systems are shown in the constellation plot of Figure 7.7(b). Two groups corresponding to fusion involving baseline systems only and (L)MLLR-SVM systems are identifiable. For

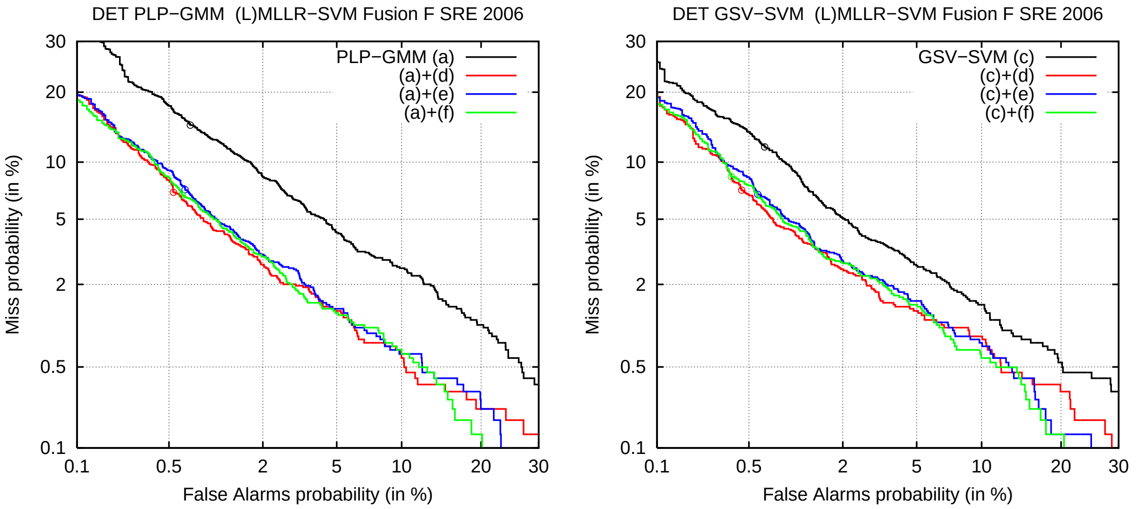

Figure 7.6 — DET curves of (L)MLLR-SVM systems combined with PLP-GMM (left,a) and GSV-SVM (right,b) for SRE 2006.

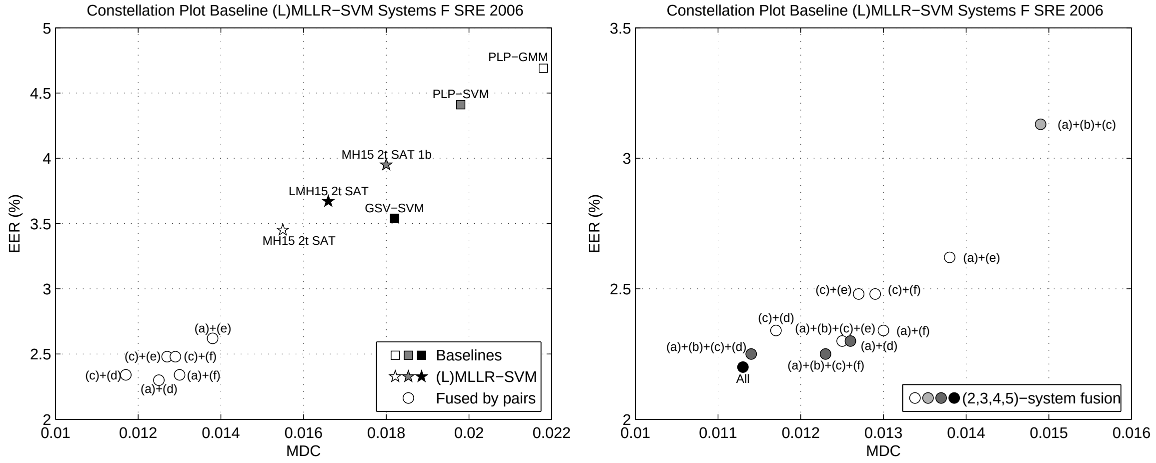

Figure 7.7 — Constellation plots of forward-scoring (L)MLLR-SVM fused systems for SRE 2006. (left, a) shows 2-system fusion and individual systems. (right, b) shows 3-, 4- and all-system fusion.

the latter, improved performance is obtained by adding more and more systems in the fusion, which is not the case for (C)MLLR-SVM systems, shown in the constellation plot of Figure 6.15(b). The use of heterogeneous transcription sources could have been responsible for this. Figures 7.8(a) and 7.8(b) show DET curves of baseline individual and several three-, four- and all-system combination. All-system combination and (a)+(b)+(c)+(d) outperform the rest of the systems for most operating points, specially in the low false-alarm rate region.

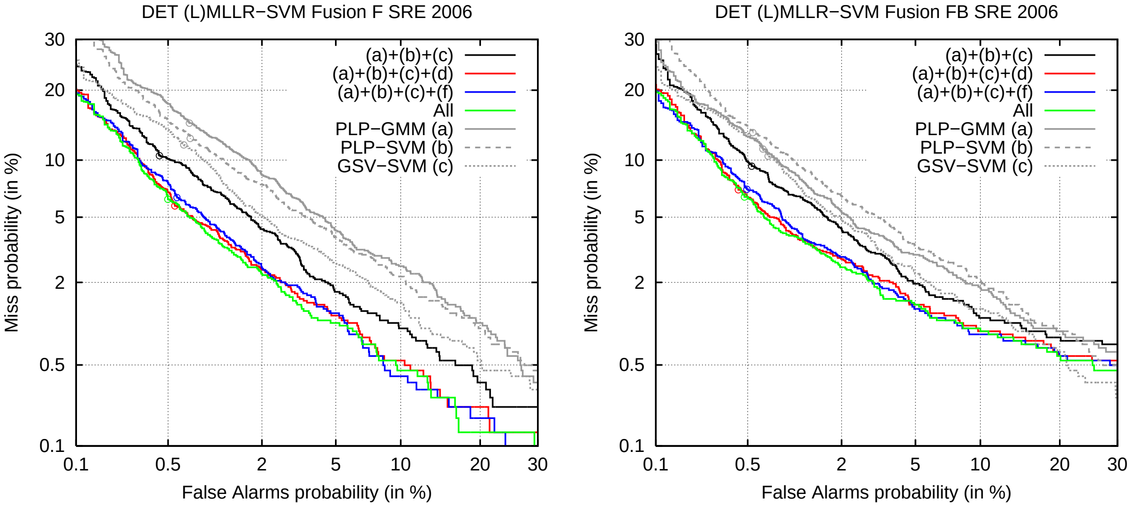

Figure 7.8 — DET curves of the baseline individual and fused systems including MH15 2t SAT (d) and LMH15 2t SAT (f) using forward scoring (left,a) and forward-backward scoring (right,b) for SRE 2006.

7.5 Summary and Conclusion

We have explored the use of lattice-based MLLR transform coefficients in MLLR-SVM systems. Lattice-based MLLR uses multiple hypotheses issued from a speech recognition system to derive soft-alignments that are used in the MLLR framework to derive regression coefficients in the usual way. We have seen that lattices obtained for different corpora significantly differ in complexity, namely arc density, possibly reflecting difficulty of transcription in the presence of more speaker and session variability in the database. LMLLR-SVM systems outperformed MLLR-SVM systems using the same transcription source, the LIMSI CTS system in our case. However, systems using 1-best transcripts provided by NIST turned out to be even more performing, which highlights the relevance of transcript quality in MLLR-SVM systems. LMLLR-SVM systems also seem to be more robust when facing data sparseness issues due to the use of a large amount of regression classes. This effect could be due to the multiple frame-to-state assignments which increases the effective number of frames per triphone state and, therefore, per regression class too.

Regarding individual system comparison, LMLLR-SVM obtained similar performance to GSV-SVM although still less performing than MLLR-SVM using transcripts provided by NIST. Fusing LMLLR-SVM with the baseline systems obtained slightly lower performance than that obtained with MLLR-SVM, which agrees with individual system performance. Including (L)MLLR-SVM systems using transcripts and lattices obtained with the LIMSI CTS system in the fusion further improved performance of the all-system combination, possibly due to using two independent sources of transcription.

Summary and Conclusion

Text-independent speaker recognition technology has relevant applications concerning indexing of audio content and intelligence services. For more than one decade, the GMM-UBM paradigm has prevailed as the ubiquitous solution to speaker verification. Of striking simplicity and performance, such an approach relies on a model of the speech signal uttered by a speaker in terms of spectral-envelope features. The success of the latter in other speech applications suggests that more specific features can be found for speaker recognition purposes. With the advent of Support Vector Machine (SVM) classifiers, the speaker recognition problem was reformulated under a new perspective which lent itself to the use of new features. One novel direction uses Maximum Likelihood Linear Regression (MLLR) coefficients obtained by adaptation of a speaker-independent model with speech data as features for speaker recognition. These coefficients, as opposed to cepstral coefficients, aim at capturing speaker-relevant cues explicitly and are classified using SVM.

Research of the MLLR-SVM approach is still in its early steps despite the high performance obtained by some systems, comparable to that of other state-of-the-art approaches. Current systems typically use standard MLLR adaptation schemes found on Large Vocabulary Continuous Speech Recognition (LVCSR) systems. We found it convenient to explore more unusual schemes with a potential on speaker recognition. This thesis makes some contributions to MLLR-SVM systems by exploring new alternatives to MLLR coefficients in a broad sense, covering both adaptation methods and representation of regression coefficients for SVM classification. We have validated the proposed approaches adhering to a international experimental evaluation protocol, the NIST Speaker Recognition Evaluation (SRE) 2005 and 2006 campaigns, enabling system comparison across laboratories.

A main reticence of the speaker recognition community on adopting the MLLR-SVM approach is the use of a Large Vocabulary Continuous Speech Recognition (LVCSR) system to compute MLLR transform coefficients. Such systems are not available to every laboratory working on speaker recognition and training its phonemic acoustic HMM requires considerable experience in the field besides being computationally expensive in both time and space resources. The high precision of these models using millions of parameters results in performing MLLR-SVM systems. However, although flexible MLLR adaptation schemes can be used, transcripts are often required to compute alignments used for MLLR estimation. Furthermore, the acoustic models are structurally dependent on a given language as a side effect of using phonemic HMM. As a consequence of the a priori information used, such MLLR-SVM approach is not directly usable or optimal in a variety of situations.

Still based on the MLLR-SVM paradigm but accounting for the limitations above we have reconsidered a fully acoustic version of it. We proposed a compact yet efficient system, CMLLR-SVM, which tackles the issues discussed above by using single-class CMLLR adaptation transforms together with Speaker Adaptive Training (SAT) of a Universal Background Model (UBM). The use of a UBM considerably reduces model training effort and eases the use of multiple SAT iterations for training. Given that training is now fully data-driven, i.e. not using any linguistic knowledge such as orthographic transcriptions, the UBM becomes structurally language-independent, thus only depending on the speech data used. We have seen that the use of CMLLR in lieu of MLLR leads to an inherent representation ambiguity of the regression coefficients issued from feature-space and model-space adaptation which compromises SVM classification accuracy. CMLLR-SVM systems using model-space CMLLR adaptation outperformed those using feature-space adaptation by over 10% MDC and EER, while the concatenation of both kinds of coefficients lies in between. Experimental results have also highlighted important gains due to using SAT for UBM training. A first SAT iteration obtains most of the gain, about 5% MDC and EER in average, while a second iteration halves it, suggesting exponential convergence of performance as more SAT iterations are performed.

Intrigued by the relatively large difference of performance between CMLLR-SVM systems using feature-space and model-space CMLLR transform coefficients, we developed a common framework in which both representations can be inscribed while giving birth to new representations that potentially lead to improved system performance. We observed that Singular Value Decomposition (SVD) of feature-space and model-space CMLLR transform matrices share the same natural basis and differ in their singular values, keeping a reciprocal relation indeed. We found that a set of transformations of CMLLR transform matrices naturally derived from its SVD decomposition by operating on their singular values only. A first proposal emphasized model-space singular vectors with respect to feature-space counterparts which was inspired from experimental results showing better performance for model-space adaptation. We used powers of the singular values of the matrix for this purpose. A second proposal emphasizes singular vectors corresponding to both feature-space and model-space components while keeping intermediate components unchanged. For both proposals, only marginal gains for systems transforming singular values with a power of 2 were observed over model-space CMLLR transform coefficients. We considered a third proposal for transforming CMLLR coefficients based on the decomposition of transform matrix coefficients into symmetric and skew-symmetric matrices. Using the coefficients derived from this decomposition we obtained gains up to 10% MDC compared to a system using model-space coefficients.

Following a popular trend in SVM-based speaker verification systems, we have used Nuisance Attribute Projection (NAP) to compensate CMLLR transform supervectors for inter-session variability. We have framed the compensation process into the same variability model used in Eigenchannel adaptation and we have compared the corresponding speaker and session estimators. Several structural limitations affecting the parameters of compensated models have been identified in our approach. Fortunately, such limitations do not apply in the CMLLR transform supervector space used for actual classification. The use of NAP in CMLLR-SVM systems obtained relative gains between 10% and 15% MDC and EER for SRE 2006 whereas these were found to be smaller for SRE 2005, eventually degrading EER. In this corpus, SAT obtained larger gains compared to NAP as opposed to what was observed in SRE 2006. We found this phenomena to be probably related to the amount of session variability, namely channel variability, included in the core condition of SRE 2005 and SRE 2006. These results tend to confirm the effectiveness of NAP in a CMLLR-SVM system. However, absolute performance of CMLLR-SVM is still one step below the state-of-the-art approaches considered in this thesis.

NAP compensation of CMLLR transform supervectors naturally lends itself to session compensation of cepstra. Considering that feature-space and model-space CMLLR transforms remove and add both speaker and session variability respectively, we have proposed to perform compensation by cascading two CMLLR transforms, one removing both components and one adding the speaker component only. The latter is obtained using NAP processing of model-space CMLLR transform supervectors and arranging its coefficients back into matrix-vector form. When evaluated on a SVM-based system using a GLDS kernel and polynomial features, our compensation scheme compares to feature mapping in terms of performance improvement. However, the latter is considerably more stable across operating points. When NAP is additionally used to compensate polynomial feature supervectors, feature-level compensation results in a marginal improvement of performance. We attribute the poor performance of our compensation approach to the underlying constraint of using a single affine transform as compensation operation, implying too strong Gaussian tying. We have observed that more successful approaches such as Eigenchannel modeling or Joint Factor Analysis (JFA) compensate each Gaussian independently which is likely to result in more powerful compensation.

Accounting for this limitation, we have addressed Gaussian tying flexibility of (C)MLLR adaptation by using multiple regression classes. Multi-class (C)MLLR-SVM systems use the concatenation of (C)MLLR transform coefficients issued from multiple regions of the acoustic space as features, as a sort of piece-wise adaptation. Adopting the more standard transcript-based approach, we use the acoustic models of a LVCSR system instead of a UBM to compute such transforms, given that Gaussian clustering in the former models is considerably more feasible and flexible. We conducted a comprehensive experimental study of adaptation schemes for (C)MLLR-SVM systems along multiple axes such as type of front-end, type of transform, number of transforms, type of model or training method. This study covered a large number of experiments for which we trained three sets of phonemic acoustic models based on two different front-ends. A relatively large amount of conclusions could be made.

(C)MLLR-SVM systems are more or less performing depending on whether CMLLR or MLLR adaptation are used. MLLR-SVM considerably outperforms CMLLR-SVM for most of the setups, especially for those obtaining best absolute performance. We found relative gains over to 30% MDC and EER for these systems. We have seen that relative performance of MLLR-SVM versus CMLLR-SVM is correlated with the degree of specialization required in the adaptation process. To further support this observation, we have analyzed CMLLR and MLLR transforms and we have observed significant differences that we believe agree with the observed difference of performance. MLLR transform matrices turned out to have 4 times larger singular value spread and 25 times larger singular value variance compared to CMLLR. Since singular values are highly related to the variance captured by transforms, we think it is fair to consider MLLR to have more adaptation flexibility, that is, more capable of adapting to different scenarios using the same amount of parameters. On the other hand, CMLLR coefficients focus on feature-space and model-space adaptation simultaneously at the price of performing poorer adaptation. Poor adaptation would be powerful enough for systems not requiring very specific adaptation, such as systems using GMM and one regression class. As more constraints are added to the system, e.g. more complex acoustic models or more regression classes, CMLLR would not be capable to keep the required adaptation precision while assuring the mean-variance constraint at the same time. For these systems, MLLR coefficients are clearly preferable. We have also observed that CMLLR transform matrices are much more orthogonal than MLLR matrices, which indicates that feature-space and model-space coefficients of the former are relatively close while belonging to inverse processes.

The use of a front-end optimized for speaker recognition tasks, involving channel compensation and normalization of feature statistics, was found to be one major source of improvement in (C)MLLR-SVM systems. Systems using (C)MLLR transform coefficients relating compensated cepstra outperformed those using a speech recognition front-end almost systematically. The improvement was found to be considerably larger for MLLR adaptation. HMM-based MLLR-SVM systems obtained an average gain around 20% MDC and EER which reduced down to 16% MDC and 13% EER for CMLLR-SVM. These further reduced for their respective GMM-based setups, less performing in absolute terms. One hypothesis explains this gain in terms of the amount of session variability captured by the regression coefficients. Compensating cepstra for channel effects prevents (C)MLLR adaptation coefficients to capture their components, thus resulting in more robust transforms. Assuming the variability that (C)MLLR coefficients are able to capture is limited, removing session or channel variability prior to adaptation may result in more adaptation power available for speaker components. In the same line, given that CMLLR transforms have less adaptation power than MLLR in order to assure the mean-variance constraint, the latter would be able to more efficiently exploit the gain resulting from using a compensated front-end, which agrees with the observed results.

The amount of parameters involved in the acoustic HMM of a LVCSR system and in a UBM can differ in several orders of magnitude. We have assessed the effect of the model used, radically changing model complexity, on single-class (C)MLLR-SVM system performance. In general terms, using HMM results in improved performance. CMLLR-SVM systems obtain gains up to 13% MDC and EER using the speaker recognition front-end but reduce to 1.5% using the speech recognition front-end. These gains rise up to 30% and 15% respectively for MLLR-SVM systems. This probably highlights the adaptation power of MLLR, being able to exploit model complexity far beyond CMLLR. Conversely, these results suggest that increasing model complexity can be unfruitful if a poor adaptation scheme is used.

Gaussian tying is plays an important role in (C)MLLR adaptation. Adaptation of a single Gaussian is equivalent to training from scratch whereas adaptation of all the Gaussians in a model can be too rough. We have assessed the effect of the number of transforms on (C)MLLR-SVM systems based on static definitions of up to three regression classes based on basic phonetic criterion. Experimental results showed that using more than one transform is always beneficial. Using two transforms corresponding to vowel and consonant classes, results in best performing systems. Gains are comparable between CMLLR-SVM and MLLR-SVM when the speech recognition front-end is used, around 8% and 10% MDC and EER respectively. When the speaker recognition front-end is used MLLR-SVM keeps the gain while CMLLR-SVM nearly fails to improve. Systems using three transforms rarely outperform two-transform counterparts. We consider two explanations accounting for it. One is that the amount of parameters estimated parameters for adaptation increases linearly with the number of transforms. The amount of data used for adaptation would not be enough to reliably estimate regression coefficients for three transforms. Another explanation is the choice of the regression classes. We arbitrarily chose the third transform to account for two subsets of vowels which might not be the best choice or not optimal at the least.

We also investigated the way acoustic models are trained under two formalisms, Maximum Likelihood Estimation (MLE) and Speaker Adaptive Training (SAT), the latter iteratively using MLE and CMLLR to obtain a more speaker-independent model. The rationale behind using SAT is that using a more speaker-independent model transforms are able to capture more speaker variability, which is interesting for speaker recognition purposes. However, in a scenario involving a large amount of channel or session variability, it is not clear whether SAT is beneficial or not since (C)MLLR transforms capture more session variability as well. Regarding CMLLR-SVM experiments, only GMM-based systems did benefit from using SAT, obtaining about a 4% MDC and EER improvement. MLLR-SVM obtained considerable gains, around 20% MDC and EER for both GMM-based and HMM-based systems. Such different behaviors still fit the idea that the constraint in CMLLR prevents proper adaptation while MLLR seem to capture more speaker variability under the SAT framework.

CMLLR and MLLR having exhibited fairly different behavior, we further explored the combination of transform coefficients in a (C)MLLR-SVM system in case further gains can be obtained due to complementarity of both adaptation variants. For this purpose we used the concatenation of model-space CMLLR and MLLR transform coefficients into a single supervector. Using multi-class CMLLR and MLLR coefficients resulted in slight gains over best MLLR-SVM setups, although not consistently for any specific setup. Among the configurations tested, those using no more than 2 CMLLR or MLLR transforms obtained best results. We also addressed matching SAT models with cepstra by applying feature-space CMLLR before computing MLLR transforms. This approach was systematically outperformed by our previous MLLR-SVM approach using unmatched cepstra and SAT models. If we think of SAT models to lie in a more speaker-independent space, MLLR coefficients capture less speaker variance if we use matched cepstra and viceversa. The adaptation components due to matching cepstra act as a sort of residual that is captured by CMLLR transform coefficients. Such an approach seems to be less effective than using straight MLLR adaptation, probably because of poor adaptation of CMLLR transforms.

In our experiments, multi-class MLLR-SVM using a speaker recognition front-end, LVCSR acoustic models trained with SAT and two transforms obtain the best absolute performance. In our experiments, this system outperformed other acoustic approaches overall including a PLP-GMM system based on the GMM-UBM paradigm, a PLP-SVM system based on the GLDS kernel and a GSV-SVM system using Gaussian mean supervectors as features. However, MLLR-SVM still relies on available transcripts for all of the data used in the system. Based on erroneous transcripts, which it is typical for spontaneous telephonic speech and transcribing a different language, the wrong acoustic models are likely to be used for both alignment and estimation of MLLR transforms. We proposed the use of lattice-based MLLR, weighting multiple hypotheses to estimate MLLR transforms, as a way to improve robustness of MLLR-SVM systems in the presence of transcription errors. Using our LVCSR system to generate word lattices, this approach consistently outperformed MLLR-SVM using 1-best hypotheses by 6% MDC and EER in average. However, it did not outperform MLLR-SVM using transcripts provided by NIST, which still evidences the impact of transcript quality on system performance. Lattice MLLR-SVM also showed smaller loss of performance when using three transforms compared to standard MLLR-SVM systems. We believe this is probably due to assigning feature vectors to multiple states in the model, which results in an increase of the effective amount of speech data used to estimate each transform.

Best performing speaker recognition systems are currently obtained by combination of several subsystems, each exploiting a different feature extraction or modeling approach. Given the novelty of MLLR coefficients as features, we thought it necessary to explore the combination of MLLR-SVM with other state-of-the-art systems. We considered score fusion using a logistic regression model of PLP-GMM, PLP-SVM and GSV-SVM acoustic systems and MLLR-SVM systems. Results showed that MLLR-SVM brings gains up to 40% and 30% MDC and EER when fused with PLP-GMM and GSV-SVM respectively. These reduce down to 15%-30% for less performing CMLLR-SVM systems, but still, including systems using (C)MLLR coefficients as features in the fusion is an efficient and simple way to improve performance. We believe this is due to the use of complementary information, given that acoustic systems use features derived from adapted models but not from the adaptation process itself. MLLR-SVM is brings major improvements of performance when fused with two or more systems as well. Lattice MLLR-SVM obtains comparable gains to MLLR-

SVM in a fused system scenario, although it is individually less performing. Including both approaches in 5-system fusion resulted in further though very slight gains.

Further Work

Despite the amount of work done for this thesis, we consider each of the explored research lines as a starting point susceptible of further study. In the following, we discuss several enhancements which we believe would be worth investigating.

Although multi-class LVCSR-based (C)MLLR-SVM systems have shown to be far more performing than UBM-based counterparts, we still stick to the idea of a fully acoustic approach based on a UBM due to simplicity and flexibility it brings. In Chapter 5 we have thoroughly used 512 Gaussian components for the UBM. This was a reasonable number accounting for the amount used for GSV-SVM systems. Given that no other point supports such a choice we believe more complex, or simpler, UBM should be tested as well to fully explore the approach. The expected improvement is probably bounded by that obtained using phonemic HMM, which involve several hundreds of times more parameters.

Using multiple transforms in a UBM-based system is another direction to pursue. Although using data-driven clustering of Gaussian components poses additional difficulties such as using variable cluster definitions, we still believe performance can be improved by performing piece-wise adaptation based on different regions of the acoustic space. Rather than CMLLR, for which multiple classes did not seem very appealing in terms of performance, we think of MLLR as effectively exploiting such adaptation schemes.

Regarding alternative representations of CMLLR adaptation, we feel SVD factorization of transform matrices is a framework to further exploit. In this line, we think a subband spectral decomposition of transform matrices is an interesting direction. Conceptually analogous to filterbank filtering, this approach would synthesize several matrices each issued from singular values in each subband of the spectra. These matrices are generated in the same space as the original transform matrices and are thus comparable between and within speakers. Moreover, since the sum of all subband matrices yields the original matrix, a SVM classifier using a linear kernel is able to reconstruct the coefficients of the latter, thus generalizing current approaches.

The interest of finding alternative representations of (C)MLLR transforms is to use them in the most performing systems and not in the basic setups, as we did. In this sense, we have not tried to use the symmetric and skew-symmetric decomposition of MLLR transform matrices. Further gains could be obtained if the improvement found on CMLLR-SVM is confirmed in multi-class MLLR-SVM systems. Alternatively, instead of finding new representation of transforms it could be interesting to investigate new kernels using the adaptation information in new ways. Kernels exploiting the matrix structure of (C)MLLR transforms could be well-suited to this purpose.

The proposed CMLLR-based feature-level session compensation scheme has been found not to be efficient although based on NAP compensation. Using more regression classes could be a starting point to see whether such a framework has a true potential or not. More complex Gaussian tying results in more precise adaptation and more precise compensation as well, given that the total amount of parameters devoted to compensation increases. We also believe that MLLR is probably better suited for multi-class adaptation. In this line, multi-class MLLR could be used to add speaker components estimating $\mathcal{C}_s$ of Equation 5.23 while feature-space CMLLR would still be focused on removing speaker and session components via $\mathcal{C}_{sh}^{-1}$.

Regarding multi-class LVCSR-based MLLR-SVM systems, removing the need for transcripts is still an important point to consider. Performing speaker recognition using cross-lingual access trials poses important structural problems to such systems. As already proposed by one site participating in the NIST SRE campaigns relying on a phone-loop model to perform CMLLR/MLLR adaptation is one line to pursue. It would also be worth considering multi-lingual phone models as well as several language-dependent acoustic models.

Conclusion

From the discussion above, (C)MLLR-SVM systems using (C)MLLR transform coefficients together with a SVM classifier stand out as a serious alternative to other state-of-the-art approaches to speaker recognition. They exhibit considerable flexibility via a large variety of adaptation schemes and combine well with other state-of-the-art acoustic approaches. In a broader sense, (C)MLLR coefficients have shown to be highly relevant features addressing speaker characterization.

Part III

Appendices

Appendix A

MLLR Transforms using Data from Multiple Acoustic Conditions

We show here that MLLR transforms obtained using adaptation data pooled from multiple acoustic conditions, e.g. concatenation of cepstra from several recording sessions for a particular speaker, can be approximated by the arithmetic mean of the MLLR transforms obtained for each of the conditions separately under particular constraints. The interest of such proof is that NAP processing of MLLR transform supervectors indirectly uses session-wise mean MLLR supervectors for every speaker to compute the inter-session covariance or kernel matrix and it could be replaced by true maximum-likelihood estimates using the concatenated data.

Although the following proof is applicable to hidden Markov models and an aribtrary number of acoustic conditions in general, we assume two acoustic conditions a and b and a Gaussian mixture model with parameters $\mathbf{\Theta}$ for the sake of simplicity. We switch to a different notation of MLLR to that in Section 2.3.3 to compactly represent MLLR estimation equations. For a set of R Gaussians in a regression class, we define the MLLR mean adaptation affine transform as

$$\hat{\boldsymbol{\mu}} = \mathbf{W}\boldsymbol{\xi} \tag{A.1}$$

where ξ is the offset-augmented mean vector

$$\boldsymbol{\xi} = (1, \mu_1, \ldots, \mu_D)^T \tag{A.2}$$

for vectors with D components.

Given the adaptation data $\mathbf{X} = (\mathbf{x_1}, \ldots, \mathbf{x_t}, \ldots, \mathbf{x_T})^T$ for $0 \leq t \leq T$, the maximization step in MLLR estimation [Leggetter & Woodland, 1994; Gales & Woodland, 1996] reduces to the solving

$$\sum_{t=0}^{T}\sum_{r=1}^{R} p(r|\mathbf{x_t})\boldsymbol{\Sigma_r^{-1}}\mathbf{x_t}\boldsymbol{\xi_r^T} = \sum_{t=0}^{T}\sum_{r=1}^{R} p(r|\mathbf{x_t})\boldsymbol{\Sigma_r^{-1}}\mathbf{W}\boldsymbol{\xi_r}\boldsymbol{\xi_r^T} \tag{A.3}$$

which is a set of D equations that need to be solved simultaneously. $p(r|\mathbf{x_t})$ is the occupation probability as defined in equation 2.10, $\boldsymbol{\Sigma_r}$ is the covariance matrix for Gaussian r, $\boldsymbol{\xi}$ its corresponding augmented mean vector as defined in Equation A.2 and $\mathbf{W}$ the MLLR affine transform we are seeking for.

If we assume segment $\mathbf{X} = (\mathbf{X^a}, \mathbf{X^b})$ to contain concatenated data from two different acoustic conditions, say condition a for $0 \leq t \leq t_{ab}$ and condition b for $t_{ab}+1 \leq t \leq T$ we can split the right hand side of equation A.3 to account for each of the conditions separately as

$$\begin{gathered}\sum_{t=0}^{T}\sum_{r=1}^{R} p(r|\mathbf{x_t})\boldsymbol{\Sigma_r^{-1}}\mathbf{x_t}\boldsymbol{\xi_r^T} = \\ \sum_{t=0}^{t_{ab}}\sum_{r=1}^{R} p(r|\mathbf{x_t})\boldsymbol{\Sigma_r^{-1}}\mathbf{W^a}\boldsymbol{\xi_r}\boldsymbol{\xi_r^T} + \sum_{t=t_{ab}+1}^{T}\sum_{r=1}^{R} p(r|\mathbf{x_t})\boldsymbol{\Sigma_r^{-1}}\mathbf{W^b}\boldsymbol{\xi_r}\boldsymbol{\xi_r^T}\end{gathered} \tag{A.4}$$

We can now develop the right hand side of equation A.4 by reversing the order of the sums and gathering $\mathbf{W^a}$ and $\mathbf{W^b}$ matrices together. Defining $n_r^a = \sum_{t=0}^{t_{ab}} p(r|\mathbf{x_t})$, $n_r^b = \sum_{t=t_{ab}+1}^{T} p(r|\mathbf{x_t})$ then

$$\begin{aligned}\sum_{r=1}^{R}\sum_{t=0}^{t_{ab}} p(r|\mathbf{x_t})\boldsymbol{\Sigma_r^{-1}}\mathbf{W^a}\boldsymbol{\xi_r}\boldsymbol{\xi_r^T} + \sum_{r=1}^{R}\sum_{t=t_{ab}+1}^{T} p(r|\mathbf{x_t})\boldsymbol{\Sigma_r^{-1}}\mathbf{W^b}\boldsymbol{\xi_r}\boldsymbol{\xi_r^T} &= \\ = \sum_{r=1}^{R}\boldsymbol{\Sigma_r^{-1}}\left(\sum_{t=0}^{t_{ab}} p(r|\mathbf{x_t})\mathbf{W^a} + \sum_{t=t_{ab}+1}^{T} p(r|\mathbf{x_t})\mathbf{W^b}\right)\boldsymbol{\xi_r}\boldsymbol{\xi_r^T} &= \\ = \sum_{r=1}^{R}\boldsymbol{\Sigma_r^{-1}}(n_r^a\mathbf{W^a} + n_r^b\mathbf{W^b})\boldsymbol{\xi_r}\boldsymbol{\xi_r^T}\end{aligned} \tag{A.5}$$

which, multiplying and dividing over $n_r = n_r^a + n_r^b = \sum_{t=0}^{T} p(r|\mathbf{x_t})$ becomes

$$\begin{aligned} &= \sum_{r=1}^{R} n_r\boldsymbol{\Sigma_r^{-1}}(\frac{n_r^a}{n_r}\mathbf{W^a} + \frac{n_r^b}{n_r}\mathbf{W^b})\boldsymbol{\xi_r}\boldsymbol{\xi_r^T} = \\ &= \sum_{r=1}^{R}\sum_{t=0}^{T} p(r|\mathbf{x_t})\boldsymbol{\Sigma_r^{-1}}(\frac{n_r^a}{n_r}\mathbf{W^a} + \frac{n_r^b}{n_r}\mathbf{W_b})\boldsymbol{\xi_r}\boldsymbol{\xi_r^T} = \\ &= \sum_{t=0}^{T}\sum_{r=1}^{R} p(r|\mathbf{x_t})\boldsymbol{\Sigma_r^{-1}}(\frac{n_r^a}{n_r}\mathbf{W^a} + \frac{n_r^b}{n_r}\mathbf{W^b})\boldsymbol{\xi_r}\boldsymbol{\xi_r^T}\end{aligned} \tag{A.6}$$

Combining equations A.4 and A.6 yields

$$\sum_{t=0}^{T}\sum_{r=1}^{R} p(r|\mathbf{x_t})\boldsymbol{\Sigma_r^{-1}}\mathbf{x_t}\boldsymbol{\xi_r^T} = \sum_{t=0}^{T}\sum_{r=1}^{R} p(r|\mathbf{x_t})\boldsymbol{\Sigma_r^{-1}}(\frac{n_r^a}{n_r}\mathbf{W^a} + \frac{n_r^b}{n_r}\mathbf{W^b})\boldsymbol{\xi_r}\boldsymbol{\xi_r^T} \tag{A.7}$$

Therefore, a MLLR transform obtained using the concatenation of the adaptation data of acoustic conditions a and b results in the weighted sum of individual MLLR transforms $\mathbf{W_a}$ and $\mathbf{W_b}$

$$\mathbf{W_{ab}} = \sum_{r=1}^{R} \left(\frac{n_r^a}{n_r} \mathbf{W^a} + \frac{n_r^b}{n_r} \mathbf{W^b} \right) \tag{A.8}$$

which can be rearranged as

$$\mathbf{W_{ab}} = \frac{n^a}{n} \mathbf{W^a} + \frac{n^b}{n} \mathbf{W^b} \tag{A.9}$$

if we let $n^a = \sum_{r=1}^{R} n_r^a$, $n^b = \sum_{r=1}^{R} n_r^b$ and $n = n_a + n_b = \sum_{r=1}^{R} n_r$.

In the context of NIST SRE campaigns, where segments have the same mean duration, and NAP compensation, where segments belong to the same speaker thus sharing strong similarities, we can make the assumption that $n_a \approx n_b$, which yields

$$\mathbf{W_{ab}} \approx \frac{1}{2} \mathbf{W^a} + \frac{1}{2} \mathbf{W^b} \tag{A.10}$$

corresponding to the arithmetic mean of MLLR transforms.

List of Figures

List of Tables

List of Acronyms

ASD Automatic Speaker Diarization

ASI Automatic Speaker Identification

ASR Automatic Speech Recognition

ASV Automatic Speaker Verification

CAT Cluster Adaptive Training

CDMA Code Division Multiple Access

CMLLR Constrained Maximum Likelihood Linear Regression

CMS Cepstral Mean Substraction

CMVN Cepstral Mean and Variance Normalization

CSR Continuous Speech Recognition

CTS Conversational Telephone Speech

DCF Decision Cost Function

DCT Discrete Cosine Transform

DET Detection Error Trade-off

DNAP Discriminant Nuisance Attribute Projection

DTW Dynamic Time Warping

EM Expectation Maximization

EMAP Extended Maximum A Posteriori

EER Equal Error Rate

F Forward scoring

FB Forward and Backward scoring

FA Factor Analysis

FIR Finite Impluse Response

FM Feature Mapping

FW Feature Warping

GLDS Generalized Linear Discriminant Sequence

GMM Gaussian Mixture Model

GSM Global System for Mobile Communications

GSV Gaussian Mean Supervector

GSVD Generalized Singular Value Decomposition

HMM Hidden Markov Model

HTER Half Total Error Rate

IDIAP Institut Dalle Molle d'Intelligence Artificielle Perceptive

ISV Inter-session Variability

JFA Joint Factor Analysis

KL Kullback-Leibler

KPCA Kernel Principal Component Analysis

KTH Kungliga Tekniska Högskolan

LDA Linear Discriminant Analysis

LDC Linguistic Data Consortium

LIMSI Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur

LMLLR Lattice-based Maximum Likelihood Linear Regression

LP Linear Prediction

LPCC Linear Prediction Cepstral Coefficient

LVCSR Large Vocabulary Continuous Speech Recognition

MAP Maximum A Posteriori

MDC Minimum Detection Cost

MEL-PLP Mel-scaled Perceptual Linear Prediction

MFCC Mel-Frequency Cepstral Coefficient

ML(E) Maximum Likelihood (Estimation)

MLED Maximum Likelihood Eigen-Decomposition

MLLR Maximum Likelihood Linear Regression

MLLT Maximum Likelihood Linear Transform

MMI(E) Maximum Mutual Information (Estimation)

MSE Mean Square Error

NAP Nuisance Attribute Projection

NIST National Institute of Standards and Technology

PLP Perceptual Linear Prediction

PCA Principal Component Analysis

PPCA Probabilistic Principal Component Analysis

RASTA Relative Spectral Transform

ROC Receiver Operating Characteristics

SAD Speech Activity Detection

SAT Speaker Adaptive Training

SDA Scatter Difference Analysis

SMO Sequential Minimal Optimization

SNR Signal to Noise Ratio

SRE Speaker Recognition Evaluation

STK Speech Toolkit

SVD Singular Value Decomposition

SVM Support Vector Machines

SWB1 Switchboard I

TFLOG Term Frequency Log Likelihood Ratio Generalized

TFLLR Term Frequency Log Likelihood Ratio

UBM Universal Background Model

VAD Voice Activity Detection

VTLN Vocal Tract Length Normalization

WCCN Within-class Covariance Normalization

WER Word Error Rate

References

[Adami *et al.*, 2003] A. Adami, R. Mihaescu, D. A. Reynolds, & J. Godfrey. Modeling prosodic dynamics for speaker recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 788–791, Hong Kong, April 2003.

[Anastasakos *et al.*, 1996] T. Anastasakos, J. McDonough, R. Schwartz, & J. Makhoul. A compact model for speaker-adaptive training. In *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, volume 2, pages 1137–1140, Philadelphia, PA, 1996.

[Atal & Hanauer, 1971] B. S. Atal & S. L. Hanauer. Speech analysis and synthesis by linear prediction of the speech wave. *Journal of the Acoustical Society of America*, 50(2B):637–655, 1971.

[Atal, 1971] B. S. Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *Journal of the Acoustical Society of America*, 55:1304–1312, 1971.

[Atal, 1976] B. S. Atal. Automatic recognition of speakers from their voices. *Proceedings of the IEEE*, 64(4):460–475, 1976.

[Auckenthaler *et al.*, 2000] P. Auckenthaler, M. Carey, & H. Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10:42–54, 2000.

[Bahl *et al.*, 1989] L. R. Bahl, P. F. Brown, P. V. de Souza, & R. L. Mercer. A tree-based statistical language model for natural language speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(7):1001–1008, 1989.

[Basu, 2003] S. Basu. A linked-hmm model for robust voicing and speech detection. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 816–819, Hong Kong, China, April 2003.

[Baum & Ergon, 1967] L. E. Baum & J. A. Ergon. An inequality with applications to statistical estimation for probabilistic functions of a a markov process and to a model for ecology. *Bulletin of the American Mathematical Society*, 73(3):360–363, 1967.

[Bilmes, 1997] J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *Technical Report ICSI-TR-97-021, International Computer Science Institute (ICSI), Berkeley, CA*, 1997.

[Bimbot *et al.*, 2004] F. Bimbot, J-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meigner, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, & D. A. Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Applied Signal Processing*, 4:430–451, 2004.

[Black *et al.*, 1992] J. R. Black, F. Jelinek, J. Lafferty, D. M. Magerman, R. Mercer, & S. Roukos. Towards history-based grammars : Using richer models for probabilistic parsing. In *Proceedings of the DARPA Spoken Language Systems Workshop*, February 1992.

[Bocchieri, 1993] E. Bocchieri. Vector quantization for efficient computation of continous density likelihoods. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 692–695, Minneapolis, 1993.

[Boser *et al.*, 1992] B. E. Boser, I. M. Guyon, & V. Vapnik. A training algorithm for optimal margin classifiers. In *Fifth Annual Workshop on Computational Learning Theory*, Pittsburgh, 1992.

[Bourlard & Morgan, 1994] H. Bourlard & N. Morgan. *Connectionist Speech Recognition*. Springer, 1994.

[Bro *et al.*, 2008] R. Bro, E. Acar, & T. G. Kolda. Resolving the sign ambiguity in the singular value decomposition. *Journal of Chemometrics*, 22(2):135–140, 2008.

[Brummer & du Preez, 2006] N. Brummer & J. du Preez. Şapplication-independent evaluation of speaker detection. *Computer Speech and Language*, 20:42–54, 2006.

[Brummer *et al.*, 2007] N. Brummer, L. Burget, J.H. Cernocky, O. Glembek, F. Grezl, M. Karafiat, D.A. van Leeuwen, P. Matejka, P. Schwarz, & A. Strasheim. Fusion of heterogeneous speaker recognition systems in the stbu submission for the nist speaker recognition evaluation 2006. *IEEE Transactions on Audio, Speech, and Language Processing*, 15:2072 – 2084, September 2007.

[Brummer, 2004] N. Brummer. Application-independent evaluation of speaker detection. In *Proceedings of the IEEE Speaker Odyssey Workshop*, pages 33–40, Toledo, Spain, 2004.

[Campbell *et al.*, 2000] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, & T. R. Leek. Phonetic speaker recognition with support vector machines. *Advances in Neural Information Processing Systems, MIT press*, 16:1377Ű1384, 2000.

[Campbell *et al.*, 2006] W. M. Campbell, D. E. Sturim, D. A. Reynolds, & A. Solomonoff. Svm based speaker verification using a gmm supervector kernel and nap variability compensation. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, 2006.

[Campbell *et al.*, 2007] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, & T. R. Leek. High-level speaker verification with support vector machines. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, page 73Ű76, Montreal, Canada, 2007.

[Campbell, 2002] W. M. Campbell. Generalized linear discriminant sequence kernels for speaker recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 161–164, 2002.

[Canseco-Rodriguez *et al.*, 2004] L. Canseco-Rodriguez, L. F. Lamel, & J. L. Gauvain. Speaker diarization from speech transcripts. In *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pages 1272–1975, Jeju, Korea, 2004.

[Chan *et al.*, 2004] A. Chan, J. Sherwani, R. Mosur, & A. Rudnicky. Four-layer categorization scheme of fast gmm computation techniques in large vocabulary continuous speech recognition systems. In *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pages 689–692, Jeju, Korea, October 2004.

[Chen *et al.*, 2000] K. T. Chen, W. W. Liau, H. M. Wang, & L. S. Lee. Fast speaker adaptation using eigenspace-based maximum likelihood linear regression. In *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, volume 3, pages 742–745, 2000.

[Chou & Juang, 2003] W. Chou & B. H. Juang. *Pattern Recognition in Speech and Language Processing*. CRC Press, 2003.

[Cieri *et al.*, 2004] C. Cieri, J.P. Campbell, H. Nakasone, D. Miller, & K. Walker. The mixer corpus of multilingual, multichannel speaker recognition data. In *Proceedings of the International Conference on Language Resources and Evaluation*, Lisbon, Portugal, 2004.

[Cieri *et al.*, 2006] C. Cieri, W. Andrews, J.P. Campbell, G. Doddington, J. Godfrey, S. Huang, M. Liberman, A. Martin, H. Nakasone, M. Przybocki, & K. Walker. The mixer and transcript reading corpora: Resources for multilingual, crosschannel speaker recognition research. In *Proceedings of the International Conference on Language Resources and Evaluation*, 2006.

[Cieri *et al.*, 2007] C. Cieri, L. Corson, D. Graff, & K. Walker. Resources for new research directions in speaker recognition: The mixer 3, 4 and 5 corpora. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, Antwerp, Belgium, August 2007.

[Cohen, 2002] I. Cohen. Optimal speech enhancement under signal presence uncertainty using log-spectral amplitude estimator. *IEEE Signal Processing Letters*, 9(4):113–116, 2002.

[Dehak *et al.*, 2008] R. Dehak, N. Dehak, P. Kenny, & P. Dumouchel. Kernel combination for svm speaker verification. In *Proceedings of the IEEE Speaker Odyssey Workshop*, Stellenbosch, South Africa, 2008.

[Dempster *et al.*, 1977] A. P. Dempster, N. M. Laird, & D. B. Durbin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[Digalakis *et al.*, 1995] V. V. Digalakis, D. Rtischev, & L. G. Neumeyer. Speaker adaptation using constrained estimation of gaussian mixtures. *IEEE Transactions on Speech and Audio Processing*, 3:357–366, September 1995.

[Digalakis *et al.*, 1996] V. Digalakis, V. Monaco, & P. Murveit. Genones: Generalized mixture tying in continuous hidden markov model-based speech recognizers. *IEEE Transactions on Speech and Audio Processing*, 4(4):281–289, 1996.

[Do, 2003] M. N. Do. Fast approximation of kullback-leibler distance for dependence trees and hidden markov models. *IEEE Signal Processing Letters*, 10:115Ű118, 2003.

[Doddington *et al.*, 2000] G. R. Doddington, M. A. Przybocki, A. F. Martin, & D. A. Reynolds. The nist speaker recognition evaluation - overview, methodology, systems, results, prespective. *Speech Communication*, 31:225Ű254, 2000.

[Doddington, 2001] G. Doddington. Speaker recognition based on idiolectal differences between speakers. In *Proceedings of European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 2521–2524, Aalborg, Denmark, 2001.

[Duda *et al.*, 2001] R. O. Duda, P. E. Hart, & D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2001.

[Ephraim & Malah, 1984] Y. Ephraim & D. Malah. Speech enhancement using optimal nonlinear spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(6):1109–1121, 1984.

[Ephraim & Malah, 1985] Y. Ephraim & D. Malah. Speech enhancement using a minimum min-square error log-spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(2):443–445, 1985.

[Fant, 1970] Gunnar Fant. *Acoustic Theory of Speech Production*. Mouton De Gruyter, 1970.

[Ferrer *et al.*, 2008a] L. Ferrer, M. Graciarena, A. Zymnis, & E. Shriberg. System combination using auxiliary information for speaker verification. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, Nevada, 2008.

[Ferrer *et al.*, 2008b] L. Ferrer, K. Sommez, & E. Shriberg. An anticorrelation kernel for improved system combination in speaker verification. In *Proceedings of the IEEE Speaker Odyssey Workshop*, Stellenbosch, South Africa, 2008.

[Furui, 1981] S. Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 292(15):254–272, 1981.

[Gales & Woodland, 1996] M. J. F. Gales & P. C. Woodland. Mean and Variance Adaptation within the MLLR Framework. *Computer Speech and Language*, 10(4):249–264, October 1996.

[Gales, 1998] M. J. F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12:75–98, 1998.

[Gales, 2000] M. J. F. Gales. Cluster adaptive training of hidden markov models. *IEEE Transactions on Speech and Audio Processing*, 8(4):417–428, 2000.

[Gauvain & Lee, 1994] J. L. Gauvain & C. H Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298, 1994.

[Gauvain *et al.*, 2003] J. L. Gauvain, L.Lamel, H. Schwenk, G. Adda, L.Chen, & F. Lefevre. Conversational telephone speech recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2003.

[Gerhard, 2003] D. Gerhard. Pitch extraction and fundamental frequency: History and current techniques. *Technical Report, Department of Computer Science, University of Regina*, 2003.

[Golub & Loan, 1996] G. H. Golub & C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

[Gorsuch, 1983] R. L. Gorsuch. *Factor Analysis*. Lawrence Erlbaum Associates, 1983.

[Haigh & Mason, 1993] J.A. Haigh & J.S. Mason. Robust voice activity detection using cepstral features. In *TENCON*, pages 321–324, Beijing, China, October 1993.

[Hatch *et al.*, 2006] A. O. Hatch, S. Kajarekar, & A. Stolcke. Within-class covariance normalization for svm-based speaker recognition. In *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, 2006.

[Hermansky & Morgan, 1994] H. Hermansky & N. Morgan. Rasta processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589, 1994.

[Hermasky, 1990] H. Hermasky. Perceptual linear prediction analysis of speech. *Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.

[Hetherington *et al.*, 1993] I. L. Hetherington, M. S. Phillips, J. R. Glass, & V. W. Zue. A* word network search for continuous speech recognition. In *EUROSPEECH*, pages 1533–1536, Berlin, Germany, September 1993.

[Huang *et al.*, 2001] X. Huang, A. Acero, & H. W. Hon. *Spoken Language Processing - A Guide to Theory Algorithm and System Development*. Prentice Hall, 2001.

[Huang *et al.*, 2007] Y. A. Huang, J. Chen, & J. Benesty. *Source Separation and Speech Dereverberation*. Springer, 2007.

[Hwang & Huang, 1993] M-Y Hwang & X. Huang. Shared distribution hidden markov models for speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1(4):414–420, 1993.

[Jaakkola & Haussler, 1999] T. Jaakkola & D. Haussler. Exploiting generative models in discriminative classifiers. *In M. S. Kearns, S. A. Solla and D. A. Cohn, editors, Advances in Neural Information Processing Systems 11, MIT Press*, 1999.

[Jain & Hermansky, 2001] P. Jain & H. Hermansky. Improved mean and variance normalization for robust speech recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Salt Lake City, Utah, USA, May 2001.

[Jelinek, 1976] F. Jelinek. Continuous speech recognition by statisical methods. *Proceedings of the IEEE*, 64(4):532–556, 1976.

[Jolliffe, 1986] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 1986.

[Jon *et al.*, 2001] E. Jon, D. K. Kim, & N. S. Kim. Emap-based speaker adaptation with robust correlation estimation. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 321–324, Salt Lake City, Utah, December 2001.

[Junqua *et al.*, 1994] J.C. Junqua, B. Mak, & B. Reaves. A robust algorithm for word boundary detection in the presence of noise. *IEEE Transactions on Speech and Audio Processing*, 2(3):406–412, 1994.

[Kajarekar & Stolcke, 2007] S. Kajarekar & A. Stolcke. Nap and wccn: Comparison of approaches using mllr-svm speaker verification system. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 249–252, Honolulu, Hawaii, 2007.

[Karam & Campbell, 2007] Z. N. Karam & W. M. Campbell. A new kernel for svm mllr based speaker recognition. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 290–293, 2007.

[Karam & Campbell, 2008] Z. N. Karam & W. M. Campbell. A multi-class mllr kernel for svm speaker recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4117–4120, Las Vegas, Nevada, March 2008.

[Katz, 1987] S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recogniser. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3):400–401, 1987.

[Kenny & Dumouchel, 2004] P. Kenny & P. Dumouchel. Experiments in speaker verification using factor analysis likelihood ratios. In *Proceedings of the IEEE Speaker Odyssey Workshop*, pages 219–226, Toledo, Spain, June 2004.

[Kenny *et al.*, 2004] P. Kenny, G. Boulianne, & P. Dumouchel. Disentangling speaker and channel effects in speaker verification. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 37–40, Montreal, Canada, May 2004.

[Kenny *et al.*, 2005] P. Kenny, G. Boulianne, & P. Dumouchel. Eigenvoice modeling with sparse training data. *IEEE Transactions on Speech and Audio Processing*, 13(3):345–354, 2005.

[Kenny *et al.*, 2007] P. Kenny, G. Boulianne, P. Ouellet, & P. Dumouchel. Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 15(4):1435–1447, 2007.

[Kristjansson *et al.*, 2005] T. Kristjansson, S. Deligne, & P. Olsen. Voicing features for robust speech detection. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 369–372, Lisbon, Portugal, September 2005.

[Kuhn *et al.*, 1998] R. Kuhn, P. Nguyen, J-C. Junqua, & L. Goldwasser. Eigenvoices for speaker adaptation. In *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pages 1771–1774, Sydney, Australia, December 1998.

[Lamel *et al.*, 1981] L. Lamel, L. Rabiner, A. Rosenberg, & J. Wilpon. An improved endpoint detection for isolated word recognition. *IEEE Acoustics, Speech, and Signal Processing Magazine*, 29:777–785, 1981.

[Lee *et al.*, 1990] H. Lee, L. R. Rabiner, R. Pieraccini, & J. G. Wilpon. Acoustic modeling for large vocabulary speech recognition. *Computer Speech and Language*, 4(2):127–165, 1990.

[Leggetter & Woodl, 1995] C. J. Leggetter & P. C. Woodl. Flexible speaker adaptation using maximum likelihood linear regression. In *Proc. ARPA Spoken Language Technology Workshop*, pages 110–115, 1995.

[Leggetter & Woodland, 1994] C. J. Leggetter & P. C. Woodland. Speaker adaptation of continuous densityt hmms using linear regression. In *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pages 451–454, Yokohama, Japan, 1994.

[Leggetter & Woodland, 1995] C. J. Leggetter & P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech and Language*, 9:171–185, 1995.

[Levinson, 1947] N. Levinson. The wiener rms error criterion in filter design and prediction. *Journal of Mathematical Physics*, 25:261–278, 1947.

[Li & Porter, 1988] K-P. Li & J. Porter. Normalization and selection of speech segments for speaker recognition scoring. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 595–598, 1988.

[Li *et al.*, 2002] Q. Li, J. Zheng, A. Tsai, & Q. Zhou. Robust endpoint detection and energy normalization for real-time speech and speaker recognition. *IEEE Transactions on Speech and Audio Processing*, 10(3):146,157, 2002.

[Li *et al.*, 2008] B. Li, J. Hu, & K. Hirasawa. Support vector machine classifier with whm offset for unbalanced data. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 12(1):94–101, 2008.

[Longworth & Gales, 2006] C. Longworth & M. J. F. Gales. Discriminative adaptation for speaker verification. In *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, Pittsburgh, Pennsylvania, September 2006.

[Louradour *et al.*, 2006] J. Louradour, K. Daoudi, & F. Bach. Svm speaker verification using an imcomplete choleski decomposition sequence kernel. In *Proceedings of the IEEE Speaker Odyssey Workshop*, pages 1–5, San Juan, Puerto Rico, 2006.

[Luo *et al.*, 2008] J. Luo, C. C. Leung, M. Ferras, & C. Barras. Parallelized factor analysis and feature normalization for automatic speaker verification. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1409–1412, Brisbane, Australia, September 2008.

[Mak *et al.*, 2006] M-W. Mak, R. Hsiao, & B. Mak. A comparison of various adaptation methods for speaker verification with limited enrolment data. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 929–932, Toulouse, France, May 2006.

[Marino *et al.*, 2000] J. B. Marino, A. Nogueiras, P. Paches-Leal, & A. Bonafonte. The demiphone: An efficient contextual subword unit for continuous speech recognition. *Speech Communication*, 32(3):187–197, 2000.

[Martin *et al.*, 1997] A. Martin, G. Doddington, T. Kamm, M. Ordowski, & M. Przybocki. The det curve in assessment of detection task performance. In *Proceedings of European Conference on Speech Communication and Technology (EUROSPEECH)*, Rhodes, Greece, September 1997.

[Martin, 1982] P. Martin. Comparison of pitch detection by cepstrum and spectral comb analysis. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 180–183, 1982.

[Martin, 2009] A. Martin. Speaker databases and evaluation. *Encyclopedia of Biometrics*, 2009.

[Mason *et al.*, 2005] M. Mason, R. Vogt, B. Baker, & S. Sridharan. Data-driven clustering for blind feature mapping in speaker verification. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, Lisbon, Portugal, September 2005.

[Matrouf *et al.*, 2007] D. Matrouf, N. Scheffer, B. Fauve, & F. Bonastre. A straightforward and efficient implementation of the factor analysis model for speaker verification. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1242–1245, 2007.

[Melin *et al.*, 1998] H. Melin, J. Koolwaaij, J. Lindberg, & F. Bimbot. A comparative evaluation of variance flooring techniques in hmm-based speaker verification. In *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pages 2379–2382, Sydney, Australia, November 1998.

[Moré & Toraldo, 1998] J. J. Moré & G. Toraldo. On the soltuion of large quadratic programming problems with bound constraints. *SIAM Journal on Optimization*, 1:93–113, 1998.

[Murveit *et al.*, 1993] H. Murveit, J. Butzberger, V. Digalakis, & M. Weintraub. Large-vocabulary dictation using sri's decipher speech recognitionsystem: progressive search techniques. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 319–322, Minneapolis, MN, USA, November 1993.

[Ney *et al.*, 1994] H. Ney, U. Essen, & R. Knseser. On structuring probabilistic dependencies in stochastic language modeling. *Computer Speech and Language*, 8(1):1–38, 1994.

[Nguyen *et al.*, 1999] P. Nguyen, C. Wellekens, & J.-C. Junqua. Maximum likelihood eigenspace and mllr for speech recognition in noisy environments. In *Proceedings of European Conference on Speech Communication and Technology (EUROSPEECH)*, 1999.

[Noll, 1969] M. Noll. Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate. In *Proceedings of the Symposium on Computer Processing ing Communications*, 1969.

[Ortmanns *et al.*, 1997] S. Ortmanns, H. Ney, & X. Aubert. A word graph algorithm for large vocabulary continuous speech recognition. *Computer Speech and Language*, 11(1):43–72, 1997.

[O'Shaughnessy, 1986] D. O'Shaughnessy. Speaker recognition. *IEEE Acoustics, Speech, and Signal Processing Magazine*, 8:4–16, 1986.

[O'Shaughnessy, 1995] D. O'Shaughnessy. What's in a number?: moving beyond the equal error rate. *Speech Communication*, 171-2:193–209, 1995.

[Osuna *et al.*, 1997] E. Osuna, R.Freund, & F.Girosi. An improved training algorithm for support vector machines. *Neural Networks for Singal Processing VII - Proceedings of the 1997 IEEE Workshop*, pages 276–285, 1997.

[Padmanabhan *et al.*, 2000] M. Padmanabhan, G. Saon, & G. Zweig. Lattice-based unsupervised mllr for speaker adaptation. In *Proceedings of the ISCA ITRW ASR2000*, pages 128–131, 2000.

[Padrell *et al.*, 2005] J. Padrell, D. Macho, & C. Nadeu. Robust speech activity detection using lda applied to ff parameters. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 557–560, Philadelphia, March 2005.

[Pelecanos & Sridharan, 2001] J. Pelecanos & S. Sridharan. Speaker recognition based on idiolectal differences between speakers. In *Proceedings of the IEEE Speaker Odyssey Workshop*, pages 213–218, Crete, Greece, 2001.

[Pigeon *et al.*, 2000] S. Pigeon, P. Druyts, & P. Verlinde. Applying logistic regression to the fusion of the nist'99 1-speaker submissions. *Digital Signal Processing*, 2000.

[Platt, 1999] J. Platt. Fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods - Support Vector Learning, B. Schölkopf, C. Burges, and A. Smola, eds., MIT Press*, pages 185–208, 1999.

[Povey *et al.*, 2003] D. Povey, M. J. F. Gales, D. Y. Kim, & P. C. Woodland. Mmi-map and mpe-map for acoustic model adaptation. In *Proceedings of European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1981–1984, Geneva, Switzerland, September 2003.

[Rabiner & Juang, 1986] L. R. Rabiner & B. H. Juang. An introduction to hidden markov models. *IEEE Acoustics, Speech, and Signal Processing Magazine*, 3(1):4–16, 1986.

[Rabiner & Juang, 1993] L. R. Rabiner & B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[Rabiner & Sambur, 1977] L. R. Rabiner & M. Sambur. Application of an lpc distance measure to the voiced-unvoiced-silence detection problem. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(4):338–343, 1977.

[Rabiner, 1989] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[Renevey & Drygajlo, 2001] P. Renevey & A. Drygajlo. Entropy based voice activity detection in very noisy conditions. In *Proceedings of European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1887–1890, Aalborg, October 2001.

[Reynolds, 1994] D. A. Reynolds. Experimental evaluation of features for robust speaker identification. *IEEE Transactions on Speech and Audio Processing*, 2(15):639–643, 1994.

[Reynolds, 1995] D. A. Reynolds. Speaker identification and verification using gaussian mixture speaker models. *Speech Communication*, 171(2):91–108, 1995.

[Reynolds, 1997] D. A. Reynolds. Comparison of background normalization methods for text-independent speaker verification. In *Proceedings of European Conference on Speech Communication and Technology (EUROSPEECH)*, 1997.

[Reynolds, 2000] D. A. Reynolds. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10:19–41, 2000.

[Reynolds, 2003] D. A. Reynolds. Channel robust speaker verification via feature mapping. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages 6–10, 2003.

[Scholkopf & Smola, 2002] B. Scholkopf & A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.

[Shriberg *et al.*, 2004] E. Shriberg, L. Ferrer, A. Venkataraman, & S. Kajarekar. SVM modeling of SNERF-grams for speaker recognition. In *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, page 1409Ű1412, Jeju, Korea, October 2004.

[Smith & Gales, 2002] N. Smith & M. Gales. Speech recognition using svms. In *Advances in Neural Information Processing Systems 14*, pages 1197–1204. MIT Press, 2002.

[Sohn *et al.*, 1999] J. Sohn, N. S. Kim, & W. Sung. A statistical model-based voice activity detection. *IEEE Signal Processing Letters*, 6(1):1–3, 1999.

[Solomonoff *et al.*, 2004] A. Solomonoff, C. Quillen, & W. M. Campbell. Channel compensation for svm speaker recognition. In *Proceedings of the IEEE Speaker Odyssey Workshop*, pages 57–62, Toledo, Spain, 2004.

[Solomonoff *et al.*, 2005] A. Solomonoff, W. M. Campbell, & I.Boardman. Advances in channel compensation for svm speaker recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 629–632, Philadelphia, PA, 2005.

[Stolcke *et al.*, 2005] A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, & A. Venkataraman. Mllr transforms as features in speaker recognition. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2425 – 2428, Lisbon, Portugal, 2005.

[Stolcke *et al.*, 2006] A. Stolcke, L. Ferrer, & S. Kajarekar. Improvements in mllr-transform-based speaker recognition. In *Proceedings of the IEEE Speaker Odyssey Workshop*, pages 1–6, San Juan, Puerto Rico, 2006.

[Stolcke *et al.*, 2007] A. Stolcke, S. S. Kajarekar, L. Ferrer, & E. Shriberg. Speaker recognition with session variability normalization based on mllr adaptation transforms. *IEEE Transactions on Audio, Speech and Language Processing*, 15:1987–1998, 2007.

[Stolcke *et al.*, 2008] A. Stolcke, S. Kajarekar, & L. Ferrer. Nonparametric feature normalization for svm-based speaker verification. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1577 – 1580, 2008.

[Swets, 1964] J. A. Swets. *Signal Detection and Recognition by Human Listeners*. Wiley & Sons, Inc., 1964.

[Tao *et al.*, 2005] Q. Tao, G-W. Wu, F-Y. Wang, & J. Wang. Posterior probability support vector machines for unbalanced data. *IEEE Transaction on Neural Networks*, 16(6):1561–1573, 2005.

[Tipping & Bishop, 1999] M. Tipping & C. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11:435–474, 1999.

[Tucker, 1992] R. Tucker. Voice activity detection using a periodicity measure. *IEE Proceedings on Comunications, Speech and Vision*, 139(4):277–280, 1992.

[Vair *et al.*, 2006] C. Vair, D. Colibro, F. Castaldo, E. Dalmasso, & P. Laface. Channel factors compensation in model and feature domain for speaker recognition. In *Proceedings of the IEEE Speaker Odyssey Workshop*, pages 1–6, San Juan, Puerto Rico, 2006.

[Vapnik, 1998] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[Viterbi, 1967] A. J. Viterbi. Error bounds for convolutional codes and an asympotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

[Vogt *et al.*, 2005] R. Vogt, B. Baker, & S. Sridharan. Modeling session variability in text-independent speaker verification. In *Proceedings of Conference of the International Speech Communication Association (INTERSPEECH)*, pages 3117–3120, Lisbon, Portugal, 2005.

[Vogt *et al.*, 2008] R. Vogt, S. Kajarekar, & S. Sridharan. Discriminant nap for svm speaker recognition. In *Proceedings of the IEEE Speaker Odyssey Workshop*, Stellenbosch, South Africa, 2008.

[Vuuren, 1996] S. Van Vuuren. Comparison of text-independent speaker recognition methods on telephone speech with acoustic mismatch. *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, 3:1788–1791, 1996.

[Waegner & Young, 1992] N. P. Waegner & S. J. Young. A trellis-based language model for speech recognition. In *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pages 245–248, Banff, Canada, October 1992.

[Wan, 2003] V. Wan. *Speaker Verification using Support Vector Machines*. PhD thesis, University of Sheffield, 2003.

[Woszczyna, 1998] M. Woszczyna. *Fast Speaker Independent Large Vocabulary Continuous Speech Recognition*. PhD thesis, Universitat Karlsruhe; Institut fur Logik, Komplexitat und Deduktionssysteme, 1998.

[Young, 1996] S. J. Young. A review of large-vocabulary continuous speech recognition. *IEEE Signal Processing Magazine*, 13(5):45–57, 1996.

[Zavaliagkos *et al.*, 1995] G. Zavaliagkos, R. Schwartz, & J. Makhoul. Batch, incremental and instantaneous adaptation techniques for speech recognition. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 245–248, Detroit, Michigan, May 1995.

[Zhao *et al.*, 2008] X. Zhao, Y. Dong, H. Yang, J. Zhao, L. Lu, & H. Wang. Nonlinear kernel nuisance attribute projection for speaker verification. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4125 – 4128, Las Vegas, Nevada, 2008.