VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ODSTRANĚNÍ VLIVU KANÁLU PRO IDENTIFIKACI JAZYKA A MLUVČÍHO

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. VALIANTSINA HUBEIKA

BRNO 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ODSTRANĚNÍ VLIVU KANÁLU PRO IDENTIFIKACI JAZYKA A MLUVČÍHO

INTERSESSION VARIABILITY COMPENSATION IN LANGUAGE AND SPEAKER IDENTIFICATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. VALIANTSINA HUBEIKA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PAVEL MATĚJKA

BRNO 2008

Abstrakt

Variabilita kanálu a hovoru je velmi duležitým problémen v úloze rozpoznávání mluvčího. V součané době je ve velkém množství vědeckých článku uvedéno několik technik pro kompenzaci vlivu kanálu. Kompenzace vlivu kanálu muže být implementována jak v doméně modelu, tak i v doménách příznaku i skóre. Relativně nová vykonná technika je takzvaná eigenchannel adaptace pro GMM (Gaussian Mixture Models). Nevýhodou této metody je nemožnost její aplikace na jiné klasifikátory, jako například takzvané SVM (Support Vector Machines), GMM s ruzným počtem Gausových komponent nebo v rozpoznávání řeči s použitím skrytých markovových modelu (HMM). Řešením muže být aproximace této metody, eigenchannel adaptace v doméně příznaku. Obě tyto techniky, eigenchannel adaptace v doméně modelu a doméně příznaku v systémech rozpoznávání mluvčího jsou uvedeny v této práci. Po dosažení dobrých výsledku v rozpoznávání mluvčího, byl přínos těchto technik zkoumán pro akustický systém rozpoznávání jazyka zahrnující 14 jazyku. V této úloze má nežádoucí vliv nejen variabilita kanálu, ale i variabilita mluvčího. Výsledky jsou prezentovány na datech definovaných pro evaluaci rozpoznávání mluvciho z roku 2006 a evaluaci rozpoznávání jazyka v roce 2007 obě organizované Americkým Národním Institutem pro Standard a Technologie (NIST).

Klíčová slova

Rozpoznávání mluvčího, rozpoznávání jazyka, akustický systém, variabilita kanálu, variabilita mluvčího, kompenzace kanalu, eigenchannel adaptace

Abstract

Varibiality in the channel and session is an important issue in the text-independent speaker recognition task. To date, several techniques providing channel and session variability compensation were introduced in a number of scientic papers. Such implementation can be done in feature, model and score domain. Relatively new and powerful approach to remove channel distortion is so-called eigenchannel adaptation for Gaussian Mixture Models (GMM). The drawback of the technique is that it is not applicable in its original implementation to different types of classifiers, eg. Support Vector Machines (SVM), GMM with different number of Gaussians or in speech recognition task using Hidden Markov Models (HMM). The solution can be the approximation of the technique, eigenchannel adaptation in feature domain. Both, the original eigenchannel adaptation and eigenchannel adaptation on features in task of speaker recognition are presented. After achieving good results in speaker recognition, contribution of the same techniques was examined in acoustic language identification system with 14 languages. In this task undesired factors are channel and speaker variability. Presented results are presented on the NIST Speaker Recognition Evaluation 2006 data and NIST Language Recognition Evaluation 2007 data.

Keywords

Speaker identification, language identification, accoustic system, session variability, interspeaker variability, channel compensation, eigenchannel adaptation, eigenfeatures

Citace

Valiantsina Hubeika: Odstranění vlivu kanálu pro identifikaci jazyka a mluvčího, diplomová práce, Brno, FIT VUT v Brně, 2008

Odstranění vlivu kanálu pro identifikaci jazyka a mluvčího

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně pod vedením ing. Pavla Matějky.

....
Valiantsina Hubeika
May 16, 2008

Contents

Chapter 1

Introduction

Text-independent recognition of speakers by their voice has been a subject of research for decades for its potential use. The goal is to recognize a speaker from recorded speech irrespective of what is being said. Speaker recognition comprehend two tasks: speaker verification, where the task is to either assign the given identity to the speech or not; and speaker identification where the identity of the speaker is to be determined given a set of reference speech segments.

There are two main areas where speaker recognition can be applied: security and search in audio data. In security, it may be needed to wait for a suspect on-line, assign the identity to the given speech signal from a database of suspects and so forth. As for search in audio, data stores are expanding rapidly therefore there is a growing demand of fast search in this data. In this case, recognizing identity of the speaker can significantly enhance searching in audio data by narrowing the search space.

Not of the less importance is the task of language recognition where the language of the speaker has to be recognized. Language recognition systems can be installed in international call centers or emergency services where the language of the speaker should be recognized during a short time period to enable a switch of the call to the operator with appropriate knowledge of the language.

Variety of speaker and language recognition systems have been developed by many research laboratories using techniques based on different approaches. To date, speaker and language identification became complex tasks composed of a number of sub-tasks to be solved. To perform accurate modeling, a large amount of data has to be available which is not always feasible to achieve in real conditions. Therefore, methods to compensate on data deficiency have to be developed. When little data is available, the reason of errors is often be cross-session variability as the training and test segments may be recorded with different session conditions. Cross-session variability comprises such factors as different channels, microphones, recording (environment) conditions, speaker's (temporal) health conditions (relevant in case of speaker recognition). In case of language recognition, cross-speaker variability is comprehended as unwanted as well.

The goal of this thesis is to analyze techniques on channel compensation for both, speaker and language recognition. As the primary task, speaker recognition was chosen for its simpler definition. All the introduced techniques are explained for speaker recognition task and consequently expanded to language recognition task.

The organization of the thesis is as follows: In chapter 2 state of the art techniques in speaker and language recognition are introduced. Chapter 3 describes acoustic systems in more detail. Chapter 4 deals with channel variability. Description of the approaches to compensate on the channel variability, eigenchannel adaptation in model and feature domain, is given in chapter 5.

Speaker recognition system and the experiments achieved are described in chapter 6. Language recognition system and the experiments achieved are described in chapter 7.

Chapter 8 sums up the work in conclusion.

Chapter 2

State of the Art

This chapter introduces the state of the art in speaker and language recognition. The structure of a general recognition system is given in figure 2.1. Overview of the methods used in single stages of recognition are listed in the following sections.

2.1 NIST Evaluations

NIST Speaker Recognition Evaluation and NIST Language Recognition are ongoing series conducted by NIST from the year 1996 on. These evaluations provide an important contribution to the direction of research efforts and the calibration of technical capabilities intended to be of interest to all researchers working on the general problem of text independent speaker and language recognition.

The goal of the NIST evaluation series is to establish the baseline of current performance capability for speaker and language recognition of conversational telephone speech and to lay the groundwork for further research efforts in the field.

2.2 Data Collection

However this thesis does not deal with data collection as such due to its complexity, the issues with the data are addressed. It is known that in recognition the more data is available the more accurate classification can be done.

In the recognition systems however, not always the required amount of data can be obtained for the target class. In speaker recognition, this problem is the most relevant in applications intended for security and defense when a new suspect who was never monitored before is to be tracked (in such cases, the application should be able to perform well using just a very little data).

Along with the large amount of data, recordings with different session conditions should be available. The biggest negative impact is seen when the target data and the data to be

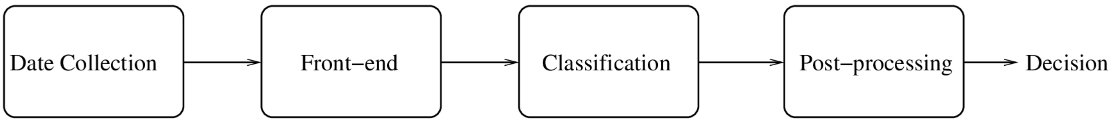

Figure 2.1: Stages of the recognition process.

classified do not match in session conditions. Cross-session variability covers such factors as inter-speaker variability, channel variability and environment variability.

In case of language identification, there are such issues like accent and dialect. Acoustic systems may get confused when enrollment and evaluation segments differ in these factors as acoustics reflect the 'sound' of the speech. Therefore, the data should be carefully selected.

Over past years, several big databases have been recorded covering a large amount of speakers and languages. Mixer database is the most suitable database for speaker recognition as it contains recordings over different channel per speaker. For the language recognition task, the CallFriend database was recorded containing conversations of half an hour; then OGI-multilingual and OGI 22 languages are available containing though only little data; Mixer database offers recordings of different languages as well.

2.3 Front-End

Before data are used for classification, feature extraction is done. The role of feature extraction process is to extract from speech signal information that is relevant for the recognition (speaker-dependent information in case of speaker recognition and language-dependent in case of language recognition) and reduce size of the input data. There are three types of features in speaker and language recognition tasks: acoustic features, phonotactic features and prosodic features. Additionally, techniques to minimize confounding effects from the acoustic features may be employed in the front-end processing such as Cepstral Mean Subtraction (CMS), Short Time Gaussianization (STG) [18], Vocal tract Length Normalization (VTLN) [27] (particularly, for language identification), RelAtive SpecTrAl (RASTA) filter [30], Heteroscedastic Linear Discriminant Analysis (HLDA) [4]. The output of this stage is a sequence of feature vectors representing the input signal.

2.3.1 Acoustic Features

The most popular features in recognition task are acoustic features. These low-level features reflect spectral identity information and are extracted from short time frames of speech. The most frequently used acoustic features in speaker and language recognition are Mel-Frequency Cepstral Coefficients (MFCC) [14] and Perceptual Linear Predictive (PLP) [17] features. To these statistical features are usually added their time derivatives (delta, double delta and triple delta), to catch inter-frame dynamics of the speech. In case of language identification Shifted Delta Ceptra (SDC) [28] calculated on MFCC proved to outperform delta features.

Acoustic Feature Transformation

After features are extracted from speech signal, there is still irrelevant information presented. To reduce this information and emphasize information important for classification, transformation techniques are used. This techniques deal with different aspects of the issue. First, distribution of the features can be transformed, secondly, dimensionality reduction can be applied.

Cepstral Mean Subtraction (CMS), short-time Gaussianization (STG) [1] are methods doing simple feature transformation. CMS computes and consequently subtracts mean value of the features over the whole utterance which intends to reduce stationary noise due to the channel. STG is a similar methods proven to outperform CMS. The basic idea is that it maps feature distribution to normal distribution over a defined window (the length of the window usually varies from 1 to 3 sec). In language recognition, VTLN [27] is applied in order to compensate of the impact due to the varying length of the speakers' vocal tract.

For dimensionality reduction and decorrelation of the features, Heteroscedastic Linear Discriminant Analysis (HLDA) [19] and Principal Component Analysis (PCA) [4] are used. As it was shown in [4] HLDA performs better than PCA.

2.3.2 High-Level Features

As the speech caries not only the acoustic information but mainly the information the speaker intends to express, the smallest lexical units, phonemes, can serve as features. Both, in speaker recognition and in language recognition task, phoneme (or word) statistics can serve as the criterion for recognition.

Analysis of longer temporal regions brings additional improvement in fusion with systems based on acoustic features. These high-level characteristics include idiosyncratic patterns in pronunciation, word usage and prosody [11], [15]. The benefit of the features is not only complementary information but also their robustness against channel distortion. However usage of high-level features reliably requires long segments of speech [15].

These high-level features can be sequences of durations of phonemes or speech segments with pitch and energy fall/rise.

2.4 Classification

2.4.1 Modeling Feature Distribution in Acoustic Systems

The dominant approach for modeling in text-independent speaker and language recognition has been Gaussian Mixture Models (GMM) [26]. For model parameters estimation the following approaches can be used: Maximum Likelihood [5] when a lot of data is available, which is rarely the case for speaker recognition and Maximum a Posteriori [26] adaptation of a speaker-independent model, called universal background model (UBM), when little data is available (realistic condition). Additionally, Maximum Mutual Information [5] can be used for models' parameter re-estimation when a lot of training data is available. It provides modeling of boundaries of the classes discriminativelly (not applicable to speaker recognition task, good results for language recognition).

Another classifier widely used in speaker and language recognition is Support Vector Machines (SVM) [8] which is a two-class maximum-margin linear classifier.

The focus of the SVM training process is to model the boundary between classes. For a separable data set, SVM optimization chooses a hyper-plane in the expansion space with maximum margin.

SVM can be trained on so-called GMM supervectors which are vectors structured from GMM means stacked in a vector. GMM supervectors are obtained by training each utterance of a class (in case of GMM systems, each class is modeled by one GMM) producing a cluster of GMM supervectors for each class. In case of language recognition, phoneme counts or word lattices may be used [6].

An advantage of SVM is that it models data discriminativelly and provides good classification performance for sparse data.

2.4.2 Modeling of Phonotactic Features in Language Recognition

Phonotactic systems are based on a phoneme recognizer followed by a language model. First, tokenization of speech to speech units, phonemes, is done. Then the statistics on n-grams (usually, uni-grams, bi-grams or trigrams) are represented by language model (LM). Language modeling can be done not only on phoneme strings but also on the on the posterior weighted phoneme counts from the lattice. Statistics' modeling can be as well efficiently done by means of Binary Decision Trees [12], [21], [22].

Often, one phoneme recognizer is used to segment the speech of all target languages to phonemes. To achieve further improvement, Parallel Phone Recognition (PPR) may be employed, where multiple phoneme recognizers are trained on different languages and run in parallel with the following score fusion.

2.5 Post-processing

After classification is done, score normalization or score calibration is often applied.

2.5.1 Normalization of the Scores

Very simple methods are zero normalization (Z-norm) and test normalization (T-norm) [16] which proved to be efficient (in some cases score normalization is not needed as other normalization/compensation methods are enrolled before or during classification).

Z-norm method normalizes the score distribution using statistics (mean and variance) calculated from the score obtained testing the target model against a set of impostor test segments.

T-norm on the contrary uses statistics (mean and variance) of the score distribution obtained by scoring each test segment against a set of impostor models. At this point the likelihood of the test segment given the target model is normalized.

2.5.2 Calibration of the Scores

For the calibration of the obtained scores, linear Gaussian backend and multi-class linear logistic regression (LLR) [13] can be used.

Chapter 3

Acoustic System

In the previous chapter, a brief overview of the techniques used in speaker recognition and language recognition task is given. This chapter gives the description of the techniques used in the acoustic recognition systems developed in this work in more detail.

3.1 Front-End

Front-end includes feature extraction and following feature post-processing. In the post-processing phase, Cepstral Mean Subtraction (CMS), Short Time Gaussianization (STG) [18] and Heteroscedastic Linear Discriminant Analysis (HLDA) [4] were applied in speaker recognition system. CMS and Vocal Tract Length Normalization (VTLN) was employed in the language recognition system.

3.1.1 MFCC

As the static features for both, speaker and language recognition, MFCC are used. First, speech is divided into overlapping frames. The conventional frame length is 25 ms with the shift of 10 ms. Each frame is proceeded by pre-emphasis filter to amplify higher frequencies. This is the approximation of psychological findings about sensitivity of human hearing on different frequencies. Hamming window is applied in the next step and Fourier spectrum is computed for the windowed signal frame. Mel filter bank is then applied to smooth the spectrum. The filters are triangular and equally spaced along the mel-scale. The Mel frequency is defined as:

$$f_{mel} = 2595 \log_{10}(1 + \frac{f}{700}) \tag{3.1}$$

where f is the original frequency. To implement this filter-bank, the window of speech data is transformed using Fourier transform and the magnitude is taken. The magnitude coefficients are then binned by correlating them with each triangular filter. Here binning means that each FFT magnitude is multiplied by the corresponding filter gain and the results accumulated. Thus, each bin holds a weighted sum representing the spectral magnitude in that filter-bank channel. MFCC are calculated from the log filter-bank amplitudes m_j using the Discrete Cosine Transform

$$c_j = \sqrt{\frac{2}{N}} \sum_{j=1}^{N} m_j \cos(\frac{j\pi}{N}(j - 0.5)) \tag{3.2}$$

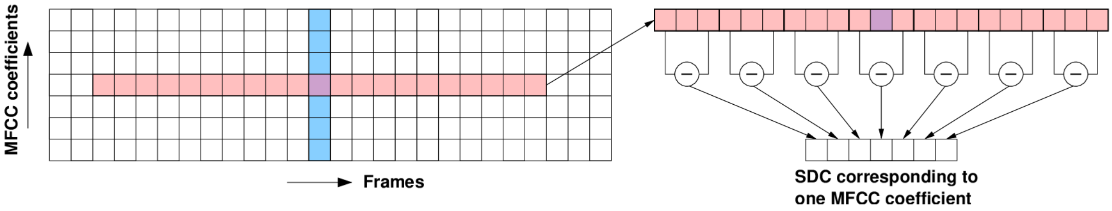

Figure 3.1: Shifted Delta Cepstra coefficients.

In case of speaker recognition, these statistical features are concatenated with deltas, double deltas and triple deltas resulting to 56 dimensional feature vector. Delta coefficients are computed from static MFCC as:

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta(c_{t+\theta} - c_{t-\theta})}{2\sum_{\theta=1}^{\Theta} \theta^2} \tag{3.3}$$

where d_t is a delta coefficient at time t computed in terms of the corresponding static coefficients $c_{t-\theta}$ and $c_{t+\theta}$. The value of Θ represents the range from which the delta coefficients are to be calculated. Double delta are computed in the same way as deltas with the only difference that they are computed from delta coefficients. And triple deltas are computed from double deltas.

In case of language recognition, time derivatives described above are proved to be outperformed by Shifted Delta Cepstra (SDC) coefficients [28] which are obtained by stacking delta cepstra computed across multiple speech frames. The SDC features are specified by 4 parameters, N, d, p and k, where N is the number of cepstral coefficients computed at each frame, d represents the time advance and delay for the delta computation, k is the number of blocks whose delta coefficients are concatenated to form the final feature vector, and P is the time shift between consecutive blocks. Accordingly, kN parameter are used for each SDC feature vector. The final vector at time t is given by the concatenation of all the $\delta c(t + iP)$, see figure 3.1, defined as:

$$\delta c(t) = c(t + iP + d) - c(t + iP - d) \tag{3.4}$$

3.2 Classification

This section gives description to the classification approach used. Reader can find here introduction to the problematics of the modeling of the feature space using probabilistic Gaussian Mixture Models (GMM). The models' parameters estimation methods and finally evaluation of the models are described.

3.2.1 Likelihood Ratio Detector

Speaker recognition comprehends speaker verification (1:1) and speaker identification (1:N). In speaker verification, the task is to verify whether the speaker is the target speaker or not. While in speaker identification the task is to assign the speaker the identity from a given set. Speaker verification can be thought of as speaker identification with only one target speaker and a firm decision threshold. Therefore the following explanation will be given for speaker verification task for its simpler model.

Speaker verification task can be stated as a basic hypothesis test between H_0, when speech segment X belongs to speaker S and H_1, when speech segment X does not belong to speaker S.

Let θ be the decision threshold for acceptance or rejecting H_0. The optimum test to decide between these two hypotheses is a likelihood ratio test given by

$$P(X) = \frac{p(X|H_0)}{p(X|H_1)} \tag{3.5}$$

if $P(X) >= \theta$ then X belongs to H_0, else X does not belong to H_0. $P(X|H_{0,1})$ is the probability density function for the hypothesis $H_{0,1}$ evaluated for the observed speech segment X, also referred to as the likelihood of the hypothesis $H_{0,1}$ given the speech segment. $H_{0,1}$ is represented by the model denoted $\lambda_{H_{0,1}}$ which represents the distribution of the features.

Gaussian Mixture Models (GMM) has proven to best represent $H_{0,1}$ in acoustic recognition systems. Thus λ_H denotes the mean vector and the covariance matrix of the GMM. The true, H_0, and alternative hypotheses, H_1, are then represented by the model λ_{H_0} and λ_{H_1} respectively. The likelihood ratio statistic can be rewritten then as $p(X|\lambda_{H_0})/p(X|\lambda_{H_1})$. Usually, the logarithm of this statistic is used to enhance the ratio computation giving the log-likelihood ration:

$$L(X) = \log p(X|\lambda_{H_1}) - \log p(X|\lambda_{H_0}). \tag{3.6}$$

While parameters of the model λ_{H_0} can be estimated from the training segments belonging to the speaker S, the parameters of the λ_{H_1} can be estimated in several ways. Perfectly, λ_{H_1} should model the distribution of all the alternative speakers. As this is not feasible, there are several approximation approaches. For tasks with a small number of the alternative (background) speakers a set of models representing these speakers can be used. Then the likelihood of the hypothesis H_1 is represented as an average or maximum of the likelihood values from the background speaker set.

Another approach which is most used in the applications with a big number of alternative speakers is to train a single speaker-independent model representing all speakers where speech segments from all speakers are pooled together. This speaker-independent model is usually called as Universal Background Model (UBM) or world model[26].

3.2.2 Gaussian Mixture Models

Gaussian Mixture Models (GMM) is a likelihood function, $p(X|\lambda)$, composed of a mixture of probability density functions. GMM is a simple approach which proved to be a powerful tool in text-independent speaker recognition tasks.

Assuming the feature distribution is modeled by GMM, the likelihood function, represented as a mixture density for a D-dimensional feature vector, $\mathbf{x}$, is defined as

$$p(\mathbf{x}, \Lambda) = \sum_{i=1}^{M} w_i * p(\mathbf{x}|\lambda_i) \tag{3.7}$$

The likelihood is a weighted linear combination of M uni modal Gaussian probability densities, $p(\mathbf{x}|\lambda_i)$, each represented by a $D \times 1$ vector, μ_i, and $D \times D$ covariance matrix, $\mathbf{\Sigma}_i$:

$$p(\mathbf{x}|\lambda_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp -\frac{1}{2}(\mathbf{x} - \mu_i)'(\mathbf{\Sigma}_i)^{-1}(\mathbf{x} - \mu_i) \tag{3.8}$$

The mixture weights, w_i, satisfy the constraint $\sum_{i=1}^{M} w_i = 1$. The parameter set of the probability density model is demoted as $\lambda = \{w_i, \mu_i, \mathbf{\Sigma}_i\}$, $i = 1, ..., M$.

Usually, the full-covariance matrix is reduces to the diagonal covariance matrix. The diagonal covariance matrix has two advantages over the full-covariance matrix: first, diagonal-matrix GMMs are more computationally efficient than full covariance GMMs (in the phase of training); secondly, the features represented by a diagonal covariance matrix are not correlated.

The feature vector $\mathbf{x}$ is assumed to be independent (which is rather not correct), therefore the log-likelihood given the model λ_i and the sequence of feature vectors, $\mathbf{x} = x_1, ..., x_T$, is computed as the sum:

$$\log p(\mathbf{x}|\lambda_i) = \sum_{t=1}^{T} \log p(\mathbf{x}_t, \lambda_i) \tag{3.9}$$

where $p(\mathbf{x}_t|\lambda)$ is computed as in Eq. 3.8. Usually, log-likelihood value is divided by T to normalize the duration effects off the log-likelihood value.

3.2.3 Universal Background Model

UBM is a GMM trained on the data from a large population of speakers to model speaker-independent distribution of the features. UBM represents the likelihood of the alternatives to hypothesized speakers, $p(\mathbf{X}|\lambda_{H_1})$.

3.2.4 Expectation Maximization Algorithm

To directly estimate parameters of a multivariate GMM is an unfeasible task. Expectation Maximization (EM) algorithm is an iterative training algorithm used in statistics to find maximum likelihood estimates of the parameters of the probabilistic models.

In this approach, the parameter-estimation problem is structured to incorporate variables representing information that is not directly observed, but that is assumed to be part of the model that generated the data (such a variable is often called hidden or missing). For instance, in the Gaussian mixture case, a hidden variable could be the index of the Gaussian that generated a data point. The key idea of EM is to estimate the probability density by taking an expectation of the logarithm of the joint density between the known

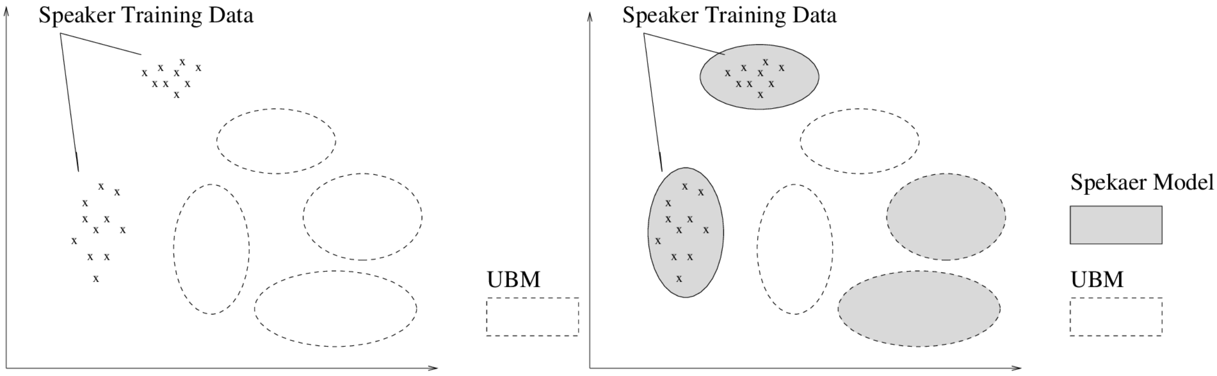

Figure 3.2: MAP adaptation of the UBM in UBM-GMM framework.

and unknown components, and then to maximize this function by updating the parameters that are used in the probability estimation. The process of Expectation and Maximization steps is then iterated to achieve good parameter estimation.

3.2.5 Maximum Likelihood Parameter Estimation

Maximum Likelihood Estimation of the GMM parameters is used when a large amount of data is available to estimate model parameters (as, for instance, for UBM parameter estimation). Given the training data $\mathbf{x}$, ML estimate, λ_{ML}, is defined as:

$$\lambda_{ML} = \arg\max_{\lambda} f(\mathbf{x}|\lambda) \tag{3.10}$$

ML training process is composed of two parts: Gaussians' parameters estimation and splitting of the Gaussians. When Gaussians are split their parameters are re-estimated using EM algorithm.

3.2.6 Maximum a Posteriori Parameter Estimation

When sparse training data are available, to obtain good models parameters' estimation UBM-GMM framework is used [26]. In UBM-GMM system, target models are derived from UBM parameters by means of Maximum a Posteriori (MAP) adaptation using the enrollment data, see figure 3.2.

Assuming λ to be a model parameter vector to be estimated from the sample $\mathbf{x}$ with probability density function (p.d.f.) $f(\cdot|\lambda)$, and g is the prior p.d.f. of λ , MAP adaptation is formulated as:

$$\begin{aligned} \lambda_{MAP} &= \arg\max_{\lambda} g(\lambda|\mathbf{x}) && (3.11) \\ &= \arg\max_{\lambda} f(\mathbf{x}|\lambda)g(\lambda) && (3.12) \end{aligned}$$

(When λ is assumed to be fixed and unknown, then equation 3.12 reduces to ML formulation.) New parameters of the model to fit better target speaker data distribution are estimated iteratively using EM algorithm in the following way: Given a sequence of training vectors of the hypothesized speaker $\mathbf{X} = (x_1, ..., x_n)$ of i.i.d (independent and identically distributed) and prior distribution of parameters given by UBM $N(\mu_{UBM}, \mathbf{\Sigma}_{UBM})$, the probabilistic alignment of the training vectors $\mathbf{x}_t$ into the UBM mixture components i, is determined and statistics for the mean (weights and variances preserve unchanged) parameters are estimated (expectation step). Then, these statistics are used to update the old UBM statistics for mixture i to create the adapted parameters for mixture i (maximization step). The process is run in several iterations. Detailed explanation of the approach can be found in [26].

The advantage over a standard Maximum Likelihood (ML) adaptation is a prior knowledge of feature distribution given by UBM. Which enables good models' parameter estimation even in the case when little training data available. Second advantage of deriving target model parameters from UBM is acceleration during scoring.

In this work, only mean vectors are MAP adapted from UBM.

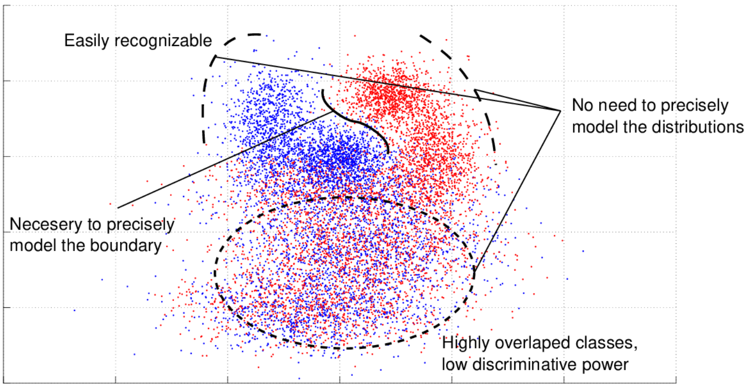

Figure 3.3: Highly overlapped distributions of features.

3.2.7 Maximum Mutual Information Parameter Estimation

Figure 3.3 presents highly overlapped distributions of the features of two classes. In such cases, training the models' parameters in a discriminative way brings significant improvement [5].

Unlike ML training which aims to maximize the overall likelihood of training data given the transcriptions, the MMI objective function is to maximize the posterior probability of correctly recognizing all training segments:

$$F_{MMI}(\lambda) = \sum_{r=1}^{R} \log \frac{p_\lambda(\mathbf{x}_r|s_r)^{K_r} P(S_r)}{\sum_{\forall s} p_\lambda(\mathbf{x}_r|s)^{K_r} P(s)} \tag{3.13}$$

where $p_\lambda(\mathbf{x}_r|s_r)$ is likelihood of r-th training segment, $\mathbf{x}_r$, given the correct transcription of the segment, s_r, and model parameters, λ. R is the number of training segments and the denominator represents the overall probability density, $p_\lambda(\mathbf{x}_r)$. Definition of the re-estimation formula is to be found in [20].

3.2.8 Log-Likelihood Ratio Computation

In the UBM-GMM approach, the log-likelihood ratio for a test speech segment X is computed as:

$$\Lambda(X) = \log(p(X|\lambda_H)) - \log(p(X|\lambda_{UBM})) \tag{3.14}$$

The fact that the hypothesized speaker model was adapted from the UBM, however, allows acceleration of the scoring process as not all Gaussians have to be evaluated in contrast as it was formerly proposed in Eq. 3.9. Top-N best Expectation Log Likelihood Ratio (ELLR) fast scoring method aims at recognizing N most contributing Gaussians in the recognition process. This fast scoring approach is based on two observations. First, when a large GMM is evaluated for an incoming test feature vector, only a few of the mixtures contribute significantly to the likelihood value. This is due to the fact that the UBM represents a large space of speaker-independent distribution of features and a single speaker GMM is derived from the UBM where not all mixtures are adopted. Additionally, in case of a short test speech segment, distribution of the features will fit under only few Gaussians. Therefore, likelihood values can be approximated using the top N best components.

The second observed effect of UBM-GMM approach is that the components of the adapted GMM retain a correspondence with the mixtures of the UBM, so that the test feature vectors close to a particular mixture in the UBM will also be close to the corresponding mixture in the speaker model. Thus, it is sufficient to determine N best scoring mixtures of the UBM for computing likelihood for both, UBM and a speaker-dependent model given the test sequence of feature vectors.

Fast scoring is efficient especially in task with multiple hypothesized speaker models for each test feature vector. In this work, top 10 components are evaluated. (For a UBM with M mixture components, calculation of log likelihood ratio using N best mixtures requires $M+N$ computations per feature vector compared to $2M$ computations for original likelihood ratio evaluation).

Chapter 4

Channel Variability

Often, enrollment and test data are recorded over different channels or under different conditions (session environment, emotional state of the speaker, possible illness, language mis-match and so on) which lowers the accuracy of correct recognition significantly. When training and test data do not match in the recording conditions, it can easily happen that the target will be recognized by the session configuration in the recording omitting the target-related information, see figure 4.1. The figure presents an example of a model distribution when two-dimensional GMM are trained each on an utterance recorded over different condition.

To date, several techniques, such as feature mapping [25], factor analysis (FA) [23], eigenchannel adaptation [2] and nuisance attribute projection (NAP) [7] have been used to compensate channel distortion.

Formerly, channel compensation was proposed task by Kenny [23] in terms of factor analysis (FA). Brümmer [2] has proposed a simplified version of FA, eigenchannel adaptation. These methods were developed within GMM framework and are implemented in model domain. Later, Castaldo in [9] has introduced an approximation of eigenchannel adaptation, eigenchannel adaptation in feature domain. With channel compensation performed in feature domain, different approaches can be used for the feature distribution modeling.

This work examines effects of eigenchannel adaptation approach for speaker recognition task. As reported in [31], eigenchannel adaptation brings significant improvement when relatively long speech segments are used for training and testing and fails when only short segments are used. It was presented that in the later case eigen-channel compensation even

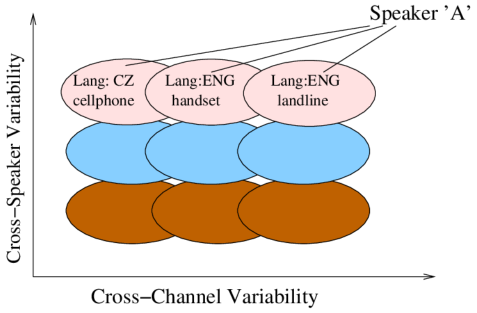

Figure 4.1: Cross-session variability vs. cross-speaker variability.

affect recognition.

Channel variability is an issue in language recognition as well as in speaker recognition. In speaker-independent language recognition, however, additionally to channel variability, influence of speaker variability has to be compensated. Additionally, dialects of the language and pronunciation (for instance, American English, British English and Indian English) may also affect correct recognition. Although, Eigenchannel adaptation was formerly proposed [2] for speaker recognition, the approach performs very well in language recognition task.

Chapter 5

Eigenchannel Adaptation

The chapter gives the theoretical background on eigenchannel adaptation method in both, model and feature domain.

5.1 Eigenchannel Subspace Estimation

Let supervector be a MD dimensional vector constructed by concatenating all GMM mean vectors and normalized by corresponding standard deviation. M is the number of Gaussian mixture components in GMM and D is dimensionality of features.

Before eigenchannel adaptation can be applied, the directions in which the supervector is mostly affected by a changing channel must be identified. These directions, which are referred to as eigenchannels, are defined by columns of $MD \times R$ matrix $\mathbf{V}$, where R is the chosen number of eigenchannels. The matrix $\mathbf{V}$ is given by R eigenvectors of average within class covariance matrix, see figure 5.1, where each class is represented by supervectors estimated on different segments spoken by the same speaker.

For each speaker, i, and all his conversations, $j = 1, ..., J_i$, UBM is adapted to obtain a supervector, s_{ij} . The corresponding speaker average supervector given by $s_i = \sum_{j=1}^{J_i} s_{ij}/J_i$ is subtracted from each supervector, s_{ij} , and resulting vectors form columns of $MD \times J$ matrix S, where J is the number of all conversations from all selected speakers ($J = 2961$ in our case). Eigenchannels (columns of matrix V) are given by R eigenvectors of $MD \times MD$

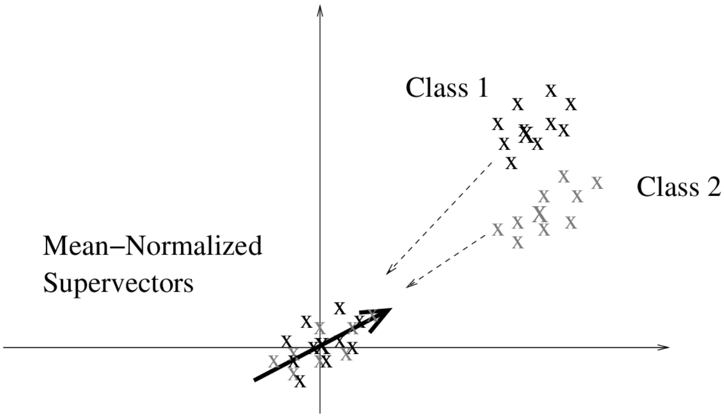

Figure 5.1: Eigenchannel vector estimation.

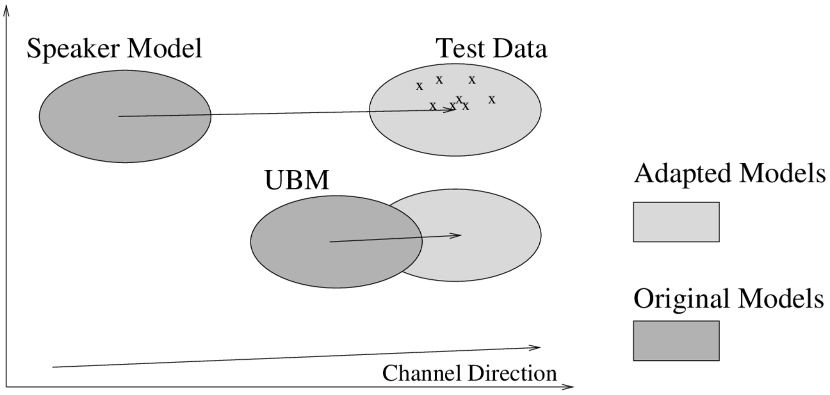

Figure 5.2: Eigenchannel adaptation in model domain.

average within speaker covariance matrix $\frac{1}{J}SS^T$ corresponding to R largest eigenvalues. Unfortunately, for our system, where MD is of a high dimension, direct computation of these eigenvectors is unfeasible. The solution is to compute eigenvectors, V , of $J \times J$ matrix $\frac{1}{J}S^TS$; eigenchannels are then given by $V = SV$. The length of each eigenchannel must be also normalized to the average within speaker standard deviation of supervectors along the direction of the eigenchannel (i.e. each eigenvector obtained in the previous step must be multiplied by the square root of the corresponding eigenvalue).

5.2 Eigenchannel Adaptation in Model Domain

In model domain, eigen-channel adaptation is applied on the supervector. The supervector is shifted in the directions given by eigenchannels to best fit the test conversation data, see figure 5.2. The task is to find the channel factor, $\mathbf{x}$, that maximizes the following MAP criterion:

$$p(\mathbf{O}|\mathbf{s} + \mathbf{V}\mathbf{x})N(\mathbf{x}; \mathbf{0}, \mathbf{I}) \tag{5.1}$$

where $\mathbf{s}$ is supervector representing the model to be adapted, $\mathbf{V}$ is given by R most relevant eigenvectors of average within-class covariance matrix, where each class is represented by supervectors estimated on different segments spoken by the same speaker, $p(\mathbf{O}|\mathbf{s} + \mathbf{V}\mathbf{x})$ is likelihood of the test conversation given the adapted supervector and $N(\mathbf{x}; \mathbf{0}, \mathbf{I})$ denotes normally distributed vector. Assuming fixed occupation of Gaussian mixture components by test conversation frames, $\mathbf{o}(t)$, $t = 1, ..., T$, it can be shown that $\mathbf{x}$ maximizing criterion (5.1) is given by:

$$\mathbf{x} = \mathbf{A}^{-1} \sum_{m=1}^{M} \mathbf{V}_m^T \sum_{t=1}^{T} \gamma_m(t) \frac{\mathbf{o}_t - \mu_m}{\sigma_m} \tag{5.2}$$

where $\mathbf{V}_m$ is $M \times R$ part of matrix $\mathbf{V}$ corresponding to m^{th} mixture component, $\gamma_m(t)$ is the probability of occupation mixture component m at time t, μ_m and σ_m are the mixture component's mean and standard deviation vectors and

$$\mathbf{A} = \mathbf{I} + \sum_{m=1}^{M} \mathbf{V}_m^T \mathbf{V}_m \sum_{t=1}^{T} \gamma_i(t) \tag{5.3}$$

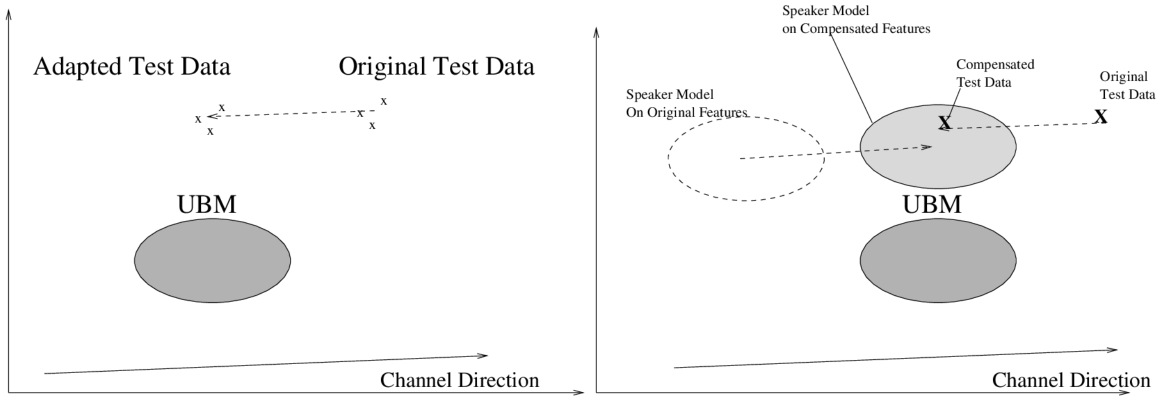

Figure 5.3: Eigenchannel Adaptation in Feature Domain.

In this implementation, occupation probabilities, $\gamma_m(t)$, are computed using Universal Background Model (UBM) and assumed to be fixed for the given test conversation. This allows to pre-compute matrix $\mathbf{A}^{-1}$ only once for each test conversation. For each frame, only Top-N occupation probabilities are assumed not to be zero.

During testing, both UBM and the target model are shifted to best fit the test segment and the score is obtained for both compensated models.

5.3 Eigenchannel Adaptation in Feature Domain

Eigenchannel adaptation in feature domain provides mapping of the compensation supervector on the acoustic features. In other words, while, eigenchannel adaptation in model domain shifts the models in the directions representing the channel variability to best fit the feature vector, eigenchannel adaptation in feature domain, shifts the feature vector to best fit the given model, see figure 5.3, left. For all the feature vectors to be adapted, the compensation supervector is represented by the UBM supervector.

The adaptation of the feature vector at time t, $\mathbf{o}'(t)$, is obtained by subtracting to the observation feature either the channel compensation offset value or a weighted sum of compensation offset values as it is proposed in [29]:

$$\mathbf{o}'(t) = \mathbf{o}(t) - \sum_{m=1}^{N} \gamma_m(t)\mathbf{V_m x} \tag{5.4}$$

where N is the selected number of best-scoring Gaussians used for the compensation. $\mathbf{V}_m$ and $\mathbf{x}$ are estimated in the same way as for the eigenchannel adaptation in model domain. We experimented with N is set to 10 and to 1. The adaptation using only 1-best Gaussian performs better.

Both, the training data and the test data are compensated. Figure 5.3, right, present the scenario of using the compensated features in the recognition system.

Chapter 6

Speaker Recognition Experiments

The speaker recognition system was built on the development data provided by NIST for SRE 2006 Evaluation according to the NIST SRE 2006 Evaluation plan [1]. The recognition system presented in this work is a test-independent speaker detection system where the task was to determine whether the target speaker is speaking in the given segment of conversational telephone speech.

6.1 Setup

6.1.1 Baseline System

As the baseline system, state of the art UBM-GMM system was built.

6.1.2 Training and Test Conditions

The training and test segments are telephone continuous conversational excerpts with no prior removal of intervals of silence. Both sides of two-channel conversations were provided.

There are 15 conditions defined by NIST in the NIST SRE 2006 Evaluation. However, the focus of this work is put to two conditions: the NIST 2006 core-condition, 1side4w-1side4w, and an optional condition, 10sec4w-10sec4w. For both conditions, only English trials were evaluated.

In the core-condition, 517 speakers were used as the target speakers. for every speaker, one two-sides conversation of the length of approximately 5 minutes was used. Each one-side segment results into approximately 2 minutes of speech.

The 10sec4w-10sec4w condition defines 429 speakers as the target speakers. For each speaker, one two-sides conversation of the length of approximately 20 seconds was provided. Each one-side segment results into approximately 10 seconds of speech.

6.1.3 Performance Measure

The results are presented in terms of the detection cost function C_{Det} computed over the sequence of trials provided where each trial is independently judged as "true" (the model speaker speaks in the test segment) or "false" (the model speaker does not speak in the test segment) and the correctness of these decisions is tallied. The detection cost function

[1] http://www.nist.gov/speech/tests/spk/2006/sre-06_evalplan-v9.pdf

is defined as a weighted sum of miss and false alarm error probabilities:

$$\begin{aligned} C_{Det} &= C_{Miss} \times P_{MissTarget} \times PTarget \quad (6.1)\\ &+C_{FalseAlarm} \times P_{FalseAlarm!NotTarget} \times (1 - P_{Target}) \quad (6.2)\end{aligned}$$

The parameters of this cost function are the relative costs of detection errors, C_{Miss} and $C_{FalseAlarm}$, and the a priori probability of the specified target speaker, P_{Target}.

Detailed information on C_{Det} computation is to be found in the NIST SRE 2006 Evaluation plan.

Graphical representation of the results in terms of Detection Error Trade-off (DET) curve are also provided in order to show how misses may be traded off against false alarms.

6.1.4 Front-End

Features

As the features 12 MFCC (plus C0, 20-ms window, 10-ms shift, and 23 bands in a Mel filter bank) were used. To compensate for channel mismatch in different conversations, two simple feature processing techniques were applied: the cepstral mean over the whole conversation is subtracted from the features and STG with 3-s window was applied. To this feature vector, its delta, double delta and triple delta coefficients were concatenated resulting in 56 dimensional vector containing information about the context of 13 frames.

Segmentation

At this phase, non-speech frames are discarded, and only speech frames are considered in the following stages of training and identification. To extract only voiced segments, BUT Hungarian phoneme recognizer was used where each recognized phoneme was assign to the voiced segments class. A post-processing with two rules based on the short-time energy of the signal is applied. 1) If the average energy in a speech segment is 30 dB less than the maximum energy in the conversation side, then the segment is labeled as silence. 2) If the energy in the opposite conversation side is bigger that the maximum energy minus 3 dB in the processed side, the segment is also labeled as silence.

HLDA

To decorrelate the features and reduce the dimensionality to 39, HLDA was applied. Several experiments were run in order to investigate to what extent the dimensionality should be reduced (chosen dimensions were 30, 39, 45 and no dimensionality reduction). The results showed that 39-dimensional feature vector performs the best.

6.1.5 Classification

First, single UBM was trained using the data for NIST 2004 evaluation. Consequently, the UBM was MAP adapted using enrollment segments. For the core condition, the models were composed of 2048 Gaussian Mixture components. In case of 10sec4w-10sec4w condition experiments with different number of Gaussian components (256, 512, 1024, 2048) were run. Scoring was done using top-10 Gaussians.

6.1.6 Eigenchannel Statistics Computation

For channel statistics calculation 2004 SRE data (both, defined as training and test segments) were used. The nominal length of the recordings is about 2 minutes.

Only speakers with at least two recordings available were used in further processing. Recordings from 310 speakers were used to calculate the statistics for the compensation. The eigenchannel matrix was estimated on all the available data (will be referred to as 2-sess matrix).

Consequently, further investigation was done on the data selection for the statistics calculation as it was assumed that when using only data with higher variability, eigenchannels may be estimated more robust.

First, speakers with at least 15 recordings were used (will be referred to as 15-sess matrix). In this case recordings only from 230 speakers were available. In the second experiment, speakers who has segments recorded over at least 3 different channels were selected (referred to as 3-chan matrix). Here, the number of speakers decreased to 202.

Third experiment incorporated both restrictions: speakers (here, 125) recorded over at least 3 channels with at least 15 recordings were used for the eigenchannel computation (referred to as 3-chan-15-sess matrix).

Analysis, which eigenchannel matrix performed better was made on the eigenchannel adaptation in model domain for its simpler implementation. The number of 50 eigenchannels with the biggest corresponding eigenvalues were used.

6.2 Results

6.2.1 Core Condition Results

Table 6.1 presents results achieved with the baseline UBM-GMM systems and the UBM-GMM system with eigenchannel adaptation in both, model and feature domain. The UBM-GMM system with eigenchannel adaptation (denoted as UBM-GMM-ccMD system) in model domain was scored several times with different eigenchannel matrices.

When employing eigenchannel adaptation in model domain, UBM-GMM-ccMD, the error rate decreased by almost 60 % relative. When experimenting with different eigenchannel matrices, no significant improvement was achieved though. As it can be seen from the table 6.1 the performance did not improve much when only the speakers with more variable data are used to estimate eigenchannels. What should be noted though is that in case of 15-sess, 3-chan and 3-chan-15-session significantly less data were used therefore eigenchannel computation was less resource-consuming.

For the eigenchannel adaptation in feature domain, eigenchannels calculated from all the available data were used. However, table 6.1 presents improvement after setting restrictions on the data used for statistics calculation, the decrease of the error is minor, therefore the matrix calculated from all the available data is used due to its assumed better capability

Table 6.1: *Results for speaker identification task. The abbreviation MD means the application of eigenchannel adaptation was done in model domain*

	Baseline	2-sess	15-sess	3-chan-15-sess	3-chan
UBM-GMM-ccMD	8.7	3.79	3.74	3.74	3.74
UBM-GMM-ccFD	-	3.97	-	-	-

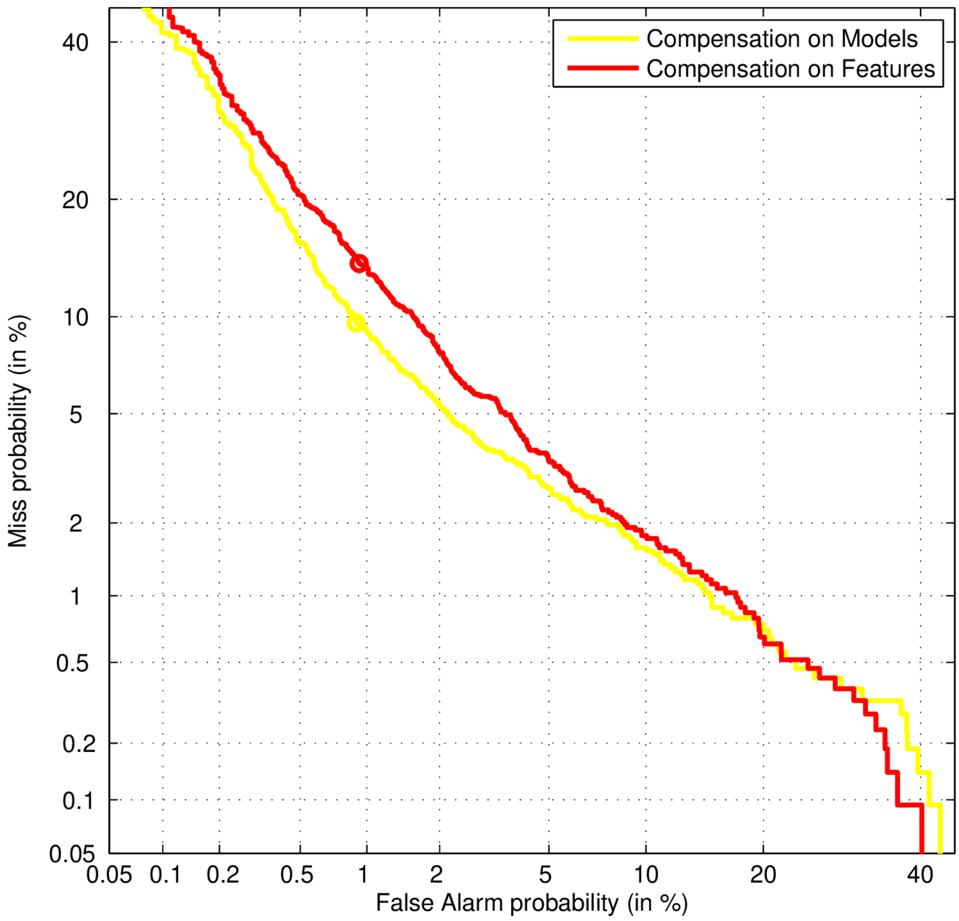

Figure 6.1: Eigenchannel adaptation in model domain vs. eigenchannel adaptation in feature domain.

of generalization. When, the system was trained on the compensated data, UBM-GMM-ccFD, the accuracy was lower than in case of the UBM-GMM-ccMD system. Graphical representation of the performance of both system, with eigenchannel adaptation in model and feature domain, is presented in figure 6.1.

6.2.2 Short-Duration Condition Results

As the short-duration conditions are of a great interest for they simulate the real condition scenario of having little training and test data, investigation of the effect of eigenchannel adaptation was studied. The main goal of the following experiments was to verify consistence in the results with other research sides and have a full set of experiments reflecting eigenchannel adaptation effects. The experiments were run with eigenchannel adaptation only in model domain.

Two top lines of the table 6.2 present the results achieved with the baseline system and the system employing eigenchannel adaptation. The results show that application of eigenchannel adaptation in the system with the same configuration as for the core condition, has a negative effect. As only 10 seconds of speech were used for the system training, using 2048

Table 6.2: *Results for speaker identification task on 10-sec condition. D stands for delta coefficients, A stands for acceleration coefficients, T stands for triple delta coefficients. HLDA39 stands for HLDA with dimensionality reduction to 39.*

	ON	OFF
2048G, DAT_HLDA39	28.23	26.88
2048G, D	24.88	24.39
1024G, DAT_HLDA39	30.76	26.79
1024G, D	26.00	**23.91**
512G, DAT_HLDA39	32.69	26.48
512G, D		24.24
256G, D		24.48
128G, D		25.44

Gaussians could lead to the incorrect modeling and therefore of affecting the performance. Hence, further experiments were run in order to find best performing configuration for the 10 second condition system.

Models with different number of Gaussian components were trained and scored without and with eigenchannel adaptation. Experiments with a shorter feature vector were run as well. The feature vector was eliminated to only static MFCC and delta coefficients with no HLDA being applied on it, thus resulting in the vector of the dimension of 26.

As it can be seen from the results in table 6.2, eigenchannel adaption affects accurate estimation in short-duration condition experiments. Negative shift in accuracy is significant.

Experiments show, that the best configuration of the system for the short-duration condition is using 12 MFCC with the only deltas and 1024 Gaussians for the feature distribution modeling without employing eigenchannel adaptation.

6.3 Conclusion

The experiments show that eigenchannel adaptation in both, model and feature domain greatly decrease the error in the core condition, where about 2 minutes of speech are available for both, training and scoring the models. Eigenchannel adaptation in model domain slightly outperforms eigenchannel adaptation in feature domain.

When only 10 seconds are available for the models' training and testing, eigenchannel adaptation (done in model domain) decreased the accuracy of the system.

Chapter 7

Language Recognition Experiments

The language recognition system was built on the development data provided by NIST for LRE 2007 Evaluation according to the NIST LRE 2007 Evaluation plan [1]. The recognition system presented in this chapter is a test-independent language detection system where the task was to determine whether the target language is spoken in the given segment of conversational telephone speech.

7.1 Setup

7.1.1 Training and Test Conditions

In the language recognition task, 14 languages were defined as the target languages. These languages were modeled using multiple databases with the amount of data varyng from 1.4 for Thai to 264 hours for English. The databases used for the training the system are the following:

CF	CallFriend
CH	CallHome
F	Fisher English Part 1.and 2.
F	Fisher Levantine Arabic
F	HKUST Mandarin
SRE	Mixer (data from NIST SRE 2004,2005,2006)
LDC07	development data for NIST LRE 2007
OGI	OGI-multilingual
OGI22	OGI 22 languages
FAE	Foreign Accented English
SpDat	SpeechDat-East[2]
SB	SwitchBoard

.

The information on duration in hours for each particular subset for each language is presented in table 7.1.

For testing the implemented system, three conditions were defined. The models had to be scored against utterances of the length of 30, 10 and 3 seconds. Altogether, over 20000 utterances were used for testing.

[1] http://www.nist.gov/speech/tests/lre/2007/LRE07EvalPlan-v8b.pdf

[2] see http://www.fee.vutbr.cz/SPEECHDAT-E or the ELRA/ELDA catalog

Table 7.1: *Training data in hours for each language and source.*

	sum	CF	CH	F	SRE	LDC07	OGI	OGI22	Other
Arabic	212	19.5	10.4	175	5.93	1.45		0.33	
Bengali	4.27				2.86	1.42			
Chinese	93.2	41.7	1.64	17.2	44.9	4.2	0.87	0.85	
English	264	39.8	4.68	162	34.9		6.77	0.52	15.6 (FAE)
Hindustani	23.5	19.6			0.64	1.32	1.53	0.42	
Spanish	54.3	43.8	6.71		2.63		1.18	0.38	
Farsi	22.7	21.2			0.03		1.00	0.42	
German	28.2	21.6	5.10				1.12	0.38	
Japanese	23.9	19.1	3.47				0.87	0.35	
Korean	19.7	18.4			0.09		0.72	0.5	
Russian	15.1				3.38	1.33		0.43	10.0 (SpDat)
Tamil	19.6	18.4					0.96	0.26	
Thai	1.45				0.15	1.23			
Vietnamese	21.6	20.6					0.79	0.27	

7.1.2 Front-End

Features

SDC with configuration of 7-1-3-7 + 7 MFCC (including C0, 20-ms window, 10-ms shift, and 23 Mel bands) were used resulting in the 56 dimensional feature vector.

VTLN

The features were transformed using vocal-tract length normalization (VTLN) [10]. The warping factors are estimated using single GMM (512 Gaussians), ML-trained on the whole CallFriend database (using all the languages). The model was trained in standard speaker adaptive training (SAT) fashion in four iterations of alternately re-estimating the model parameters and the warping factors for the training data.

Segmentation

Segmentation was done using BUT phoneme recognizer where the phoneme classes were linked to the speech class. The frames labeled as silence were dropped from the further processing. Further, the segments shorter than 50 ms were omitted as it was supposed that the information it held was not reliable enough.

7.1.3 Classification

Two UBM models with 256 and 2048 Gaussian mixtures were trained using all training data. Consequently, 14 target models were MAP adapted from the UBM using enrollment data. Several experiments were then run with MMI models' parameter re-estimation (for the MMI criterion see chapter 3.2.7) using both, 256-Gaussian and 2048-Gaussian systems.

7.1.4 Calibration

All systems were first processed by linear backend and then calibrated using multi-class linear logistic regression (LLR) [3] after classification. Both linear backend and calibration parameters were trained using the part of the training data which were not used for the training of the models. The FoCal Multi-class toolkit by Niko Brummer[3] was used for the processing.

7.1.5 Eigenchannel Statistics Computation

In the language detection task the most disturbing factors are channel and speaker variability, recordings from a high number of speakers from different databases were used to perform the channel and speaker variability compensation.

To calculate the covariance matrix for eigenchannel vectors estimation, recordings from a large number of speakers were used covering 14 languages presented in the system. For each language, maximum 100 recordings from different speakers were randomly selected. First, to calculate the covariance matrix the mean value was subtracted over a language. The eigenchannel matrix computed in from this average within-class covariance matrix will be referred to as 14-class eigenchannel matrix.

Then, recordings from each language were divided to subgroups containing data only from one particular database. The maximum number of recordings for each language was increased to make 100 in each subgroup. Not for every subgroup though enough data was available to fulfill the limit (very little data for Thai and Bengali). Subgroups with less than 8 recordings were omitted. The number of subgroups resulted thus into 56 and the computed eigenchannel matrix will be referred to as 56-class eigenchannel matrix. The number of 50 eigenchannels with the biggest corresponding eigenvalues were used.

7.1.6 Assure Metric

The results are introduced in terms of avgCDET which is the measure defined by NIST. Basic pair-wise likelihood ratio (LR) performance is computed for all target/non-target language pair. Basic LR performance is represented directly in terms of detection miss and false alarm probabilities. For each test, miss probability is computed separately for each target/non-target language pair. In addition, these probabilities are combined into a single number that represents the cost performance of a system, according to an application-motivated cost model:

$$\begin{aligned} C(L_T, L_N) &= C_{Miss} P_{Target} P_{Miss}(L_T) && (7.1) \\ &+ C_{FA}(1 - P_{Target}) P_{FA}(L_T, L_N) && (7.2) \end{aligned}$$

where L_T and L_N are the target and non-target languages, and C_{Miss}, C_{FA} and P_{Target} are application model parameters. For LRE07, the application parameters are set to be:

$$\begin{aligned} C_{Miss} = C_{FA} = 1 && (7.3) \\ P_{Target} = 0.5 && (7.4) \end{aligned}$$

These performance are computed separately for each of the three segment duration categories.

[3] http://niko.brummer.googlepages.com/focalmulticlass

Table 7.2: *Results for language identification task,* 2048*-GMM.*

	Baseline	MD, 14 classes	MD, 56 classes	FD, 56 classes
30 sec	8.03	5.4	2.76	2.91
10 sec	12.89	8.32	7.37	7.64
3 sec	21.77	18.98	17.63	17.48

Table 7.3: *LID: eigenchannel adaptation in feature domain and MMI parameter re-estimation.*

	256G-chcf	256G-MMI	256G-MMI-chcf	2048G-MMI-chcf
30 sec	4.88	4.15	3.73	2.41
10 sec	11.07	8.61	9.81	7.02
3 sec	22.45	18.43	20.98	16.90

7.2 Results

The results are presented in table 7.2. As it can be seen from the table, eigenchannel adaptation in model domain brought a big improvement. Eigenchannel statistics calculation using 56 classes had an advantage over using only 14 classes. When using the 56-class eigenchannel matrix, the error decreased almost to its third comparing to the baseline. Therefore in all the following experiments the 56-class eigenchannels are considered.

Eigenchannel adaptation in feature domain performed similar to the eigenchannel adaptation in feature domain and slightly outperformed on the short-duration condition.

Further MMI training was applied on the system trained on the compensated features and further decrease of the error was reached. Two systems were built on the compensated features, 256-Gaussian system and 2048-Gaussian system.

Table 7.2 presents three system with 256 Gaussians. 256G-chcf is the UBM-GMM system trained on the compensated features whereas 256G-MMI is the system with the MMI on top of MAP adaptation trained on the original features. The 256G-MMI performed better than 256G-chcf but the combination of both, 256G-MMI-chcf, brings slight decrease in the error on the 30 second condition. Then a system similar to 256G-MMI-chcf but with 2048 Gaussians was trained and further decreased of the error was reached. (The details on the compensation setup is described in the following subsection, see the 2048G-256G system.)

Besides the improvement of the results when using MMI in combination with eigenchannel adaptation in feature domain, the number of iterations required for the re-estimation decreased to 3 comparing to the 15 iteration in case of 256G-MMI system.

Duration-Dependent Eigenchannel Adaptation in Feature Domain for the 2048 Gaussian Systems

The approach of eigenchannel adaptation in model domain requires a fixed number of Gaussians in the system. The length of the supervectors used for the eigenchannel statistics calculations must equal to the numbed of Gaussians in the language-dependent models. Eigenchannel adaptation in feature domain overcomes this restriction. The number of

Table 7.4: *LID: Results achieved using alternatively high or low dimensional UBM for eigenchannel adaptation in feature domain depending on the length of the speech segment.*

	2048G-2048G	2048G-256G
30 sec	4.03	2.94
10 sec	7.51	7.40
3 sec	18.39	17.93

Gaussians used for statistics calculation depends only on the number of Gaussians of the UBM used for the compensation but may differ from the number of Gaussians in the system. An intuitive assumption (also backed by a number of experiments) would be that with decreasing duration of speech segments the number of Gaussians in the system should be lowered in order to achieve better generalization.

As the length of the training segments varies (from 20 milliseconds to tens of minutes) and the nominal length of the test data is 3, 10 and 30 seconds, using 2048 Gaussians for the compensation may not always fulfill the expectation of catching the main directions of the channel distortion.

Additionally, overly high number of Gaussians may cover insignificant changes in the channel. This may result in lowering the performance of the approach. At this point, several duration-dependent compensation experiments were performed. The aim was to use eigenchannel matrix (and the UBM) of lower dimension for the speech segments of short duration.

In the experiments, the supervector dimensions of 256 (for short segment) and 2048 (for long segments) were chosen. The results for the following experiments are presented in table 7.4. First, both the training and the test data were compensated using the 2048 Gaussian UBM, see the 2048G-2048G system. Then, the data from all three test data sets were compensated using the UBM with 256 Gaussians, see the 2048G-256G. The 2048G-256G system performed better.

Observing such a decrease in the error, the training data were compensated using the 256-Gaussian UBM, unfortunately decrease of the performance was achieved. Further, the training data were divided to two subsets, the short-duration data subset and the long-duration data subset depending on the set threshold. The threshold was set to two different values, 1 minute and 2 minutes. Yet, no improvement was reached comparing to the 2048G-256G system. (The results for these three experiments are not enclosed).

7.2.1 Calibration

The calibration of the obtained scores was an important part in building the systems. To outline the effect of the calibration, the results of the uncalibrated GMM2048-MMI-chcf system are present as well as of the calibrated system (see Tab 7.5). However, in case of 3 second condition, the decrease of the error is only about 8 % relative, in case of 30 sec condition, more than 50 % of relative reduction in the error was observed.

7.2.2 English, Indian and Chinese Languages

Special attention should be paid to English, Indian and Chinese languages as they are assumed to be 'difficult' languages due to the variety in pronunciation and a big number of

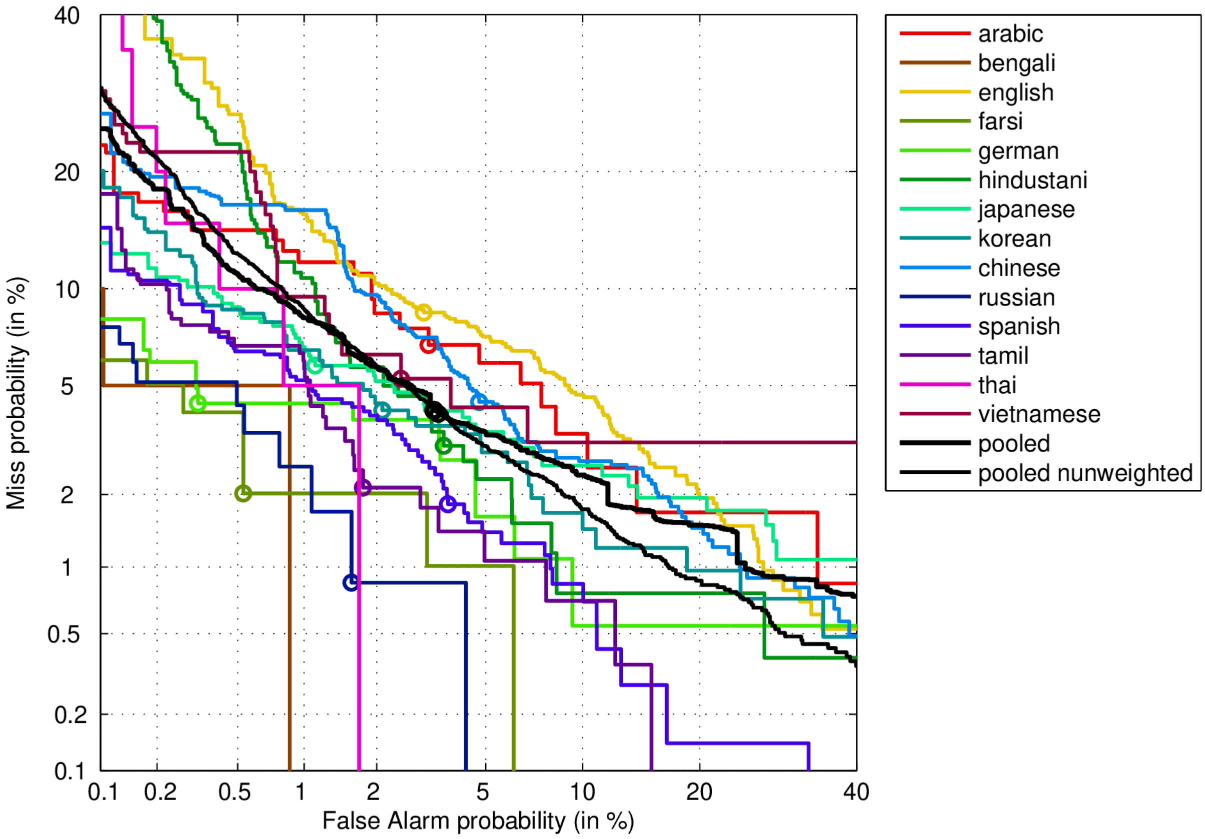

Figure 7.1: LID system with eigenchannel adaptation in model domain, 30 sec test condition.

dialects. As it can be seen from figure 7.2 the performance of the system on these languages is significantly lower that on the other languages.

7.3 Languages with Little Training Data

Another issue in the language detection task can be lack of the training data. In case of Thai and Bengali, there were very little training data available. The solution can be using the telephone speech data acquired from the public media such as radio. By now, several preliminary experiments with Thai language were performed on investigation of the performance of the recognition system using the radio telephone speech data. The results are presented in table 7.3 for the 10-second condition. The 10-second condition was chosen as the results on the 30-second condition are over-optimistic. Although, the performance in case of the radio data is not as good as in case of the standard telephone data, the usage of public media data would be a good solution in case of the language for which no data is available. More information on the scenario of the acquiring the data and more results can be found in [24].

Table 7.5: LID: Effect of calibration for the GMM2048-MMI-chcf on LRE 2007 data

	30 sec	10 sec	3 sec
No back-end	5.75	9.45	18.44
LDA+LLR	2.41	7.02	16.90

Table 7.6: LID: Results achieved using alternatively original Thai telephone data and Thai telephone data acquired from radio.

	No channel comp.		Channel comp.	
	NIST	Radio	NIST	Radio
DCF all lang.	12.83	13.66	7.30	7.56
Thai DCF	7.81	11.61	3.93	6.05

7.4 Conclusion

Eigenchannel adaptation in both model and feature domain proved to be efficient in the language detection task. A big decrease of the error is seen when using either of the compensation technique comparing to the baseline system. Eigenchannel adaptation in feature domain more over in combination with MMI parameter re-estimation brings further decrease of the error.

It was shown that eigenchannel adaptation performs well when the radio data are used instead of the standard the data which allows including to the recognition system the languages which had been not previously used due to the absence of the training data.

Chapter 8

Conclusions

Cross-channel variability is one of the most important issues faced in recognition systems from recorded speech. When little data is available for the target to be recognized, cross-channel variability may be a key factor in affecting the correct recognition as the training and test data can be recorded over different channels. When the recording condition of the segment to be recognized was never seen in the training data, then compensation techniques have to be applied. This work showed that cross-channel variability can be compensated efficiently using eigenchannel adaptation.

Eigenchannel adaptation performs well in speaker and in language recognition systems based on the UBM-GMM framework. Both versions of the technique, the original eigenchannel adaptation in model domain and its approximation, eigenchannel adaptation in feature domain proved to carry out good results. However, eigenchannel adaptation in model domain, being applied in the GMM systems during scoring, has its limitations. Eigenchannel adaptation in feature domain is a transformation performed on features. Therefore, once channel compensation is done in feature domain, channel variability does not have to be taken into account any longer. Eigenchannel adaptation in feature domain brings possibility in training the GMM parameter using different training methods (MMI), modeling the feature distribution with different models(SVM, GMM with different number of Gaussians, HMM). Improvement of the performance of the recognition systems, when using eigenchannel adaptation, was presented in two tasks, speaker recognition task and language recognition task.

Both, eigenchannel adaptation in model and feature domain were performing well. The results show, however, eigenchannel adaptation in model domain outperforms eigenchannel adaptation in feature domain, the system using eigenchannel adaptation in feature domain may be more robust than the one using eigenchannel adaptation in model domain as addition improving techniques may be applied. This work shows on the language recognition task, that when a GMM system is trained on the compensated features and the models' parameters are consequently re-estimated using MMI re-training, the system significantly outperformed the classical UBM-GMM system with eigenchannel adaptation in model domain. Further investigation is to be done on training different type of models on the compensated features, especially promising is to use SVM classifier in combination with GMM supervectors trained on the compensated features.

Experiments on language recognition using radio telephone data showed that for the previously unprocessed languages due to the missing training data, recordings from public media can be used. Although the data are not of the same quality as the standard telephone data, the recognition system employing eigenchannel adaptation performed reasonably well.

Further investigation on the radio telephone data for language recognition is to be done as the radio could be the only source of a large amount of data for some languages of interest.

Bibliography

[1] J.-L. Barras, C.; Gauvain. Feature and score normalization for speaker verification of cellular data. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP apos;03). 2003 IEEE International Conference*, volume 2, pages 49–52, apr 2003.

[2] Niko Brummer. Spescom DataVoice NIST 2004 system description. In Proc. NIST Speaker Recognition Evaluation 2004, 2004.

[3] N. BrĂźmmer, L. Burget, J. Cernocky, O. Glembek, F. Grezl, M. Karafiát, D. van Leeuwen, P. Matejka, P. Schwarz, and A. Strasheim. Fusion of heterogeneous speaker recognition systems in the stbu submission for the nist speaker recognition evaluation 2006. In *IEEE Transactions on Audio, Speech and Language Processing*, volume 15, 7, pages 2072–2084, 2007.

[4] Lukáš Burget. Combination of speech features using smoothed heteroscedastic linear discriminant analysis. In *Proc. 8th International Conference on Spoken Language Processing*, pages 2549–2552. Sunjin Printing Co,, 2004.

[5] Lukáš Burget, Pavel Matějka, and Jan Černocký. Discriminative training techniques for acoustic language identification. In *Proceedings of ICASSP 2006*, pages 209–212, 2006.

[6] W. M. Campbell, F. Richardson, and D. A. Reynolds. Language recognition with word lattices and support vector machines. In *IEEE International Conference on Acoustics, Speech and Signal Processing ICASP 2007*, volume 4, pages IV–989 – IV–992, Apr 2007.

[7] W.M. Campbell. Compensating for mismatch in high-level speaker recognition. In *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006*, pages 1–6, 2006.

[8] W.M. Campbell, D.E. Sturim, D.A. Reynolds, and A. Solomonoff. Svm based speaker verification using a gmm supervector kernel and nap variability compensation. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006*, volume 1, pages 97–100, may 2006.

[9] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, and C. Vair. Compensation of nuisance factors for speaker and language recognition. In *IEEE Transactions on Audio, Speech, and Language Processing*, volume 15, 7, pages 1969–1978, 2007.

[10] J. Cohen, T. Kamm, and A.G. Andreou. Vocal tract normalization in speech recognition: Compensating for systematic speaker variability. In *J. Acoust. Soc. Am.*, 97, page 2346, 1995.

[11] Speaker Recognition Douglas. The supersid project: Exploiting high-level information for high-accuracy.

[12] O. Glembek et al. Advances in phonotactic language recognition. In *submitted to International Conferences on Spoken Language Processing (ICSLP)*, 2008.

[13] P. Matějka et al. Brno university of technology system for nist 2007 language recognition evaluation. In *submitted to International Conferences on Spoken Language Processing (ICSLP)*, 2008.

[14] Zheng Fang, Zhang Guoliang, and Song Zhanjiang. Comparison of different implementations of mfcc. *J. Comput. Sci. Technol.*, 16(6):582–589, 2001.

[15] L. Ferrer. Modeling duration patterns for speaker recognition, 2017.

[16] Auckenthaler R.; Carey M.; Lloyd-Thomas H. Score normalization for text-independent speaker verification systems. In *Signal Processing Applications for Public Security and Forensics, 2007. SAFE '07. IEEE Workshop*, pages 1–5, apr 2007.

[17] H. Hermansky and L. A. Cox. Perceptual linear predictive (plp) analysis-resynthesis technique. *Applications of Signal Processing to Audio and Acoustics, 1991. Final Program and Paper Summaries., 1991 IEEE ASSP Workshop on*, pages 0_37–0_38, 1991.

[18] S. Sridharan J. Pelecanos. Feature warping for robust speaker verification. In *A Speaker Odyssey*, pages 213–218, jun 2001.

[19] N. Kumar. *Investigation of Selicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition.* PhD thesis, John Hopkins University, 1997.

[20] P. Matějka, L. Burget, P. Schwarz, and J. Černocký. Brno university of technology system for nist 2005 language recognition evaluation. In *IEEE Odyssey: The Speaker and Language Recognition Workshop*, pages 57–64, 2006.

[21] J. Navratil. Spoken language recognition-a step toward multilinguality in speech processing. In *IEEE Trans. on Speech and Audio Processing*, pages 678–685, 2001.

[22] J. Navratil. Recent advances in phonotactic language recognition using binary-decision trees. In *International Conferences on Spoken Language Processing (ICSLP)*, 2006.

[23] P. Dumouchel P. Kenny. Experiments in speaker verification using factor analysis likelihood ratios. In *Odyssey: The Speaker and Language Recognition Workshop*, pages 219–226, 2004.

[24] O. Plchot, V. Hubeika, L. Burget, P. Schwarz, and P. Matějka. Acquisition of telephone data from radio broadcasts with applications to language recognition. In *submitted to: 11th International Conference on Text, Speech and Dialogue (TSD)*, 2008.

[25] D. A. Reynolds. Channel robust speaker verification via feature mapping. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP 03).*, volume 2, pages II – 53–6 vol.2, April 2003.

[26] D. A. Reynolds. Speaker verification using adapted gaussian mixture models. *title*, pages 19–41, January year.

[27] William Tecumseh Sherman. Vocal tract length perception and the evo lution of language.

[28] Pedro A. Torres-Carrasquillo, Elliot Singer, Mary A. Kohler, Richard J. Greene, Douglas A. Reynolds, J.R. Deller, and Jr. Approaches to language identification using gaussian mixture models and shifted delta cepstral features.

[29] C. Vair, D. Colibro, F. Castaldo, E. Dalmasso, and P. Laface. Channel factors compensation in model and feature domain for speaker recognition. In *Speaker and Language Recognition Workshop, IEEE Odyssey*, pages 1–6, jun 2006.

[30] S. van Vuuren and H. Hermansky. the importance of components of the modulation spectrum for speaker verification, 1998.

[31] R. Vogt, Ch J. Lustri, and S. Sridharan. Factor analysis modelling for speaker verification with short utterances. In *Odyssey 2008: The Speaker and Language Recognition Workshop*, 2008.