Brno University of Technology
Faculty of Information Technology
Department of Computer Graphics and Multimedia

Dipl.-Ing. Marcel Kockmann

Subspace modeling of prosodic features for speaker verification

Study programme: Information Technology

Doctoral thesis

Supervisor: Doc. Dr. Ing. Jan Černocký

Abstract:
The thesis investigates into speaker verification by means of prosodic features. This includes an appropriate representation of speech by measurements of pitch, energy and duration of speech sounds. Two diverse parameterization methods are investigated: the first leads to a low-dimensional well-defined set, the second to a large-scale set of heterogeneous prosodic features. The first part of this work concentrates on the development of so called prosodic contour features. Different modeling techniques are developed and investigated, with a special focus on subspace modeling. The second part focuses on a novel subspace modeling technique for the heterogeneous large-scale prosodic features. The model is theoretically derived and experimentally evaluated on official NIST Speaker Recognition Evaluation tasks. Huge improvements over the current state-of-the-art in prosodic speaker verification were obtained. Eventually, a novel fusion method is presented to elegantly combine the two diverse prosodic systems. This technique can also be used to fuse the higher-level systems with a high-performing cepstral system, leading to further significant improvements.

Keywords:
speaker verification, prosody, Gaussian mixture models, channel compensation, Joint factor analysis, total variability model, iVector, probabilistic linear discriminant analysis, SNERFs, subspace multinomial model, iVector fusion

Contents

Chapter 1

Introduction

1.1 Automatic Speaker verification

Automatic speaker verification deals with the task of verifying the claimed identity of a previously trained speaker from a recorded utterance. One has to distinguish between text dependent and text-independent speaker verification. In the text-dependent task, the system knows the content of the utterance (e.g. a certain passphrase) as well as the speaker. This work deals with the text-independent verification of a speaker, in which case the system has no prior information about the speech. In both cases, the speaker has to be enrolled by a certain amount of speech before performing recognition. The most interesting application for text-independent speaker verification is likely to be in the field of forensics and intelligence.

Figure 1.1 shows the general approach to automatic speaker recognition. In the first step, some kind of information has to be extracted from the speech signal to represent discriminative characteristics of an individual. In the training phase, these so called features are extracted from speech of a particular speaker and are used to build a statistical model which represents the speaker. In the test phase, an unknown utterance is also transformed to the same kind of features and these are then fed to a classifier which decides if the features fit to the statistical model of the speaker in test.

1.2 Prosodic Speaker verification

High-level information has been used for over a decade to further enhance short-time, cepstral-based speaker verification systems. Many approaches make use of acoustic attributes of speech prosody that mainly involve variations in syllable length, loudness, and pitch. In recent NIST Speaker Recognition

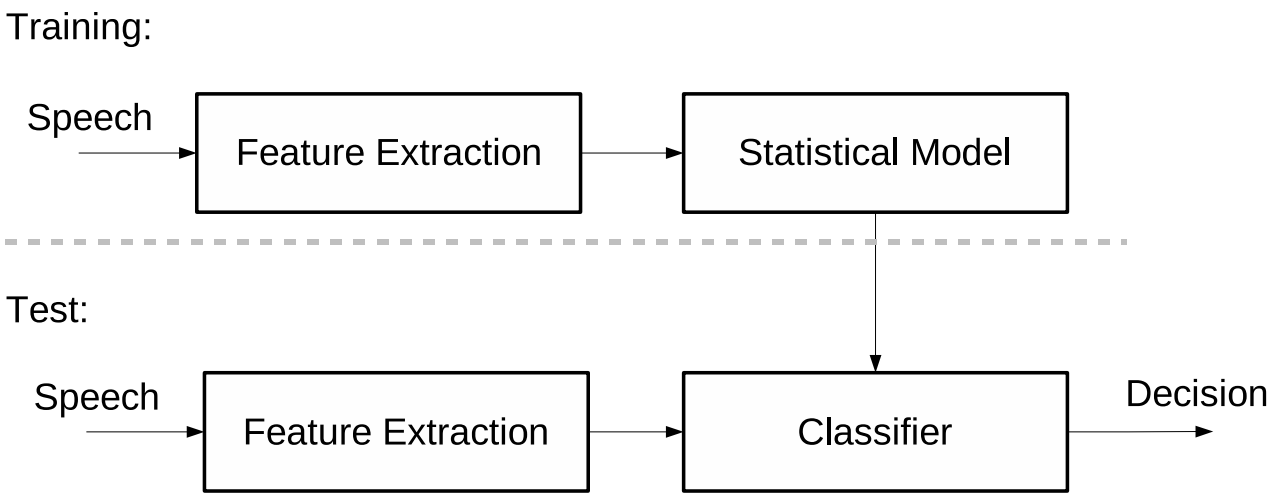


Figure 1.1: General approach to statistical speaker recognition.

Evaluations [20, 21], two families of prosodic feature sets were presented. One family corresponds to syllable-based, non-uniform extraction region features (SNERFs) [27], which are highly complex prosodic features originally proposed by SRI. These features in combination with specialized parameterization methods and support vector machine (SVM) modeling [11] result in a very good prosodic system.

Another family of systems uses a set of very simple prosodic features, originally proposed for language identification [19]. These features model the temporal trajectory of pitch and energy over the time span of a syllable. Joint Factor Analysis (JFA) modeling for these features was originally proposed by [7] and showed very promising results. This framework for prosodic modeling has been adopted by several sites and is investigated thoroughly in this thesis [16, 10]. The main reason for its success lies in JFA modeling, which is capable of coping with the problem of speaker and session variability in Gaussian mixture model (GMM)-based speaker verification [14] and has become the de facto standard for modeling low- and high-level features.

Moreover, excellent results on cepstral features were obtained with a simplified variant of JFA [8], where separate subspaces for channel and speaker variability are replaced by a single subspace covering the total variability. This model can be used to extract compact low-dimensional feature vectors representing a whole utterance, often called iVectors. Based on this idea, we propose a framework where the subspace modeling technique normally used to model means of GMMs is adapted to model occupation counts using a multinomial model. This so-called Subspace Multinomial Model (SMM) [17] is applicable

to the complex SNERFs to extract iVectors.

Probabilistic Linear Discriminant Analysis (PLDA) [23] has been proposed to model the speaker and channel variability in both types of iVectors, directly generating likelihood ratios for the trials [15, 5]. iVector modeling of SNERFs by SMMs with successive PLDA has been shown to give the best results for a prosodic speaker verification system so far [18].

To date, the iVector approach – using a total variability subspace followed by PLDA – has not been used (to our knowledge) for the simple prosodic features that are usually modeled by JFA.

In this work, we further present results on the prosodic JFA system as presented by Brno University of Technology in SRE 2010 and apply iVector modeling and PLDA back end to the same features. We show that the iVector approach is superior to the standard JFA modeling even for simple prosodic features.

Eventually, we have two diverse prosodic systems that achieve similar performance on our test sets: an iVector system that models means of GMMs based on simple well-defined prosodic features and an iVector system that models counts of multinomial distributions based on SNERFs. A combination of both systems seems relevant due to their complementary nature in terms of features and modeling. We propose an elegant way of combining these systems by simple concatenation of individual iVectors followed by a single joint PLDA model. This combination achieves an equal error rate (EER) of 5.4% on our NIST SRE 2008 telephone test set, a 23% gain over the best of the two systems.

Justification for use of a higher-level systems usually lies in an overall improvement by fusion with a cepstral baseline system. Usually, combination of low- and high-level systems is done by score-level fusion using a separate development set to train the fusion parameters. As the best-performing cepstral systems to date are also based on iVector modeling followed by PLDA modeling [15, 5, 4], we are inspired by the successful combination of two prosodic iVector front ends to further combine the cepstral and prosodic systems in the same manner. We achieve a relative reduction in terms of the challenging new detection cost function (DCF) [21] of 17% for SRE 2010 data and 21% for SRE 2008 data. The iVector combination consistently outperforms standard score-level fusion (11% and 13%) with no need for a separate development set to train the fusion parameters.

Chapter 2

Prosodic features

This section describes the two prosodic feature sets used in the thesis.

2.1 DCT contour features

The initial intention was to use a finer modeling of pitch and energy contours than used in the linear stylization by [28] and [2]. The use of a curve-fitting algorithm based on higher-order polynomials [24] seemed to be an appropriate way, suitable also for speaker recognition. This way, each pitch or energy segment can be represented by a fixed number of the corresponding polynomial coefficients and form a fixed sized feature vector. It is then possible to model these prosodic feature vectors by standard UBM-GMM paradigm [25] as used for standard cepstral based features.

In the very early literature research phase of this thesis, it was found that the same idea was recently implemented by [7]. Not only did they use a polynomial approximation of pitch and energy based on suprasegmental units, but also they already incorporated intersession variability compensation based on Joint Factor Analysis in the modeling approach.

Although this idea of curve-fitting based prosodic feature extraction had already been used and published, the excellent results obtained by [7] motivated me to continue the work on the prosodic level and to develop an own prosodic feature extraction module. The proposed approach to prosodic contour feature extraction mainly differs in two ways: First, a simpler way of parameterizing the temporal trajectories should be used, than by using a curve fitting module. [7] used Legendre polynomials that are fitted in a least-square-error sense to the original contour segments. Second, the idea was to derive the suprasegmental units in a different way. In [7], the segmentation is simply based on local minima in the signal energy. On the one hand, in the proposed approach, even

higher level information should be incorporated, by deriving pseudo-syllable units using a language independent phone recognizer. On the other hand, one idea was to use a very simple fixed-size long-temporal context.

This section describes the process of extracting the proposed prosodic contour features. First, it is briefly described how a loudness and fundamental frequency measure is obtained. Next, it will be described how the duration measure is obtained by segmenting the speech in suprasegmental units. Finally, it is shown how to parameterize the information encoded in loudness and fundamental frequency for each variable-sized suprasegmental unit to a fixed-sized feature vector.

2.1.1 Basic prosodic features

The quantity that is actually being estimated by all "pitch trackers" is the fundamental frequency (F0). F0 is defined as the lowest frequency of a periodic waveform and is an inherent property of periodic speech signals. It tends to correlate well with perceived pitch (that is strictly defined otherwise, see [29]). In time domain, it can be defined as the inverse of the smallest period in the interval being analyzed. For typical male adults, F0 will lie between 85–180 Hz and for females between 165–255Hz [30].

We will briefly describe a popular family of pitch algorithms that work directly on the time signal [29]. Those F0 estimation algorithms often comprises three stages:

1. Pre-processing.

2. Estimation of candidates for true periods.

3. Selection of best candidate and F0 refinement.

The aim of the pre-processing phase is to remove interfering signal components from the audio signal. This is usually done by a band-pass filter or some sort of noise reduction. Note, that a standard telephone signal (that we mostly work with) is already band-pass filtered from 300–3400Hz due to the standard telephone channel. However, the fundamental frequency can still be inferred through its harmonics in the signal.

The estimation of F0 candidates itself is mostly performed directly on the time signal using correlations within the signal as a traditional source of period candidates. A widely used and robust pitch tracking algorithm is the RAPT

algorithm [29], that is based on the Normalized Cross-Correlation Function (NCCF).

The RAPT algorithm consists of the following steps [29]:

1. Generate two version of sampled speech data, one at the original sample rate and one at a significantly reduced rate.
2. Compute NCCF of low sample rate signal for all lags in the F0 range of interest. This first pass records the located local maxima.
3. Compute NCCF of high sample rate only in vicinity of the peaks found in the first pass, again record new maxima.
4. Generate F0 candidates and unvoiced probability for each frame from the second NCCF pass.
5. Use Dynamic Programming (DP) to select the best path through the candidates of the whole utterance.

The output of a pitch tracker is a continuous F0 contour. When there is no pitch detected (in unvoiced regions or speech pauses) the algorithms simply outputs zeros.

Prosodic features measuring the loudness of speech are usually directly obtained from the signal energy [3]. The short-time energy of the speech signal can be either extracted directly from the time signal or equivalently from its squared magnitude spectrum.

Before any further processing, the raw pitch and energy values are first transformed to the logarithmic domain to compress their dynamic range. The energy values are further normalized by subtracting the maximum value over the whole utterance to make the loudness measure less dependent on any constant background noises in the utterance. The pitch values are further filtered by a median filter to smooth the contour.

2.1.2 Suprasegmental units

The time span of the prosodic suprasegmental units is used in two ways for the contour features: First, the size of each segment is used as a single duration feature. Second, the segment boundaries determine the pitch or energy sequence that is being modeled.

The literature proposes many methods to define suprasegmental units for prosodic feature extraction, most of them using phonetically motivated syllable-like units. A syllable can be seen as a unit of organization of speech sounds,

or as a phonological building block which has influence on rhythm, stress and other prosodic attributes of speech.

Various approaches will be investigated, with a special interest in their computational complexity and further constraints, like language dependence. Two of these approaches are newly proposed during the work on this thesis and are described in the following.

The first approach to segment the speech into syllable-like units is based on the basic assumption that a syllable is typically made up of a syllable nucleus (most often a vowel) with optional initial and final margins (onset and coda, typically consonants). By using this assumption we can derive syllable-like units from a phone recognizer. Further, to be less language independent, one can use a phonetically rich recognizer [26]. The proposed segmentation algorithm consists of the following steps:

1. Extract language independent phones.

2. Map phones to coarse classes silence, vowel and consonant.

3. For each region between two silence labels:

   - Consider each vowel as the nucleus of a syllable.
   - Set the syllable boundaries to equally distant (as far as possible) phone boundaries in between two vowels.
   - If syllable boundary "cuts" a pitch sequence, while a another possible candidate does not, move the boundary there

This process is illustrated in Figure 2.1. The vertical lines indicate the phone boundaries. Highlighted are the three vowels that are found for a speech segment between two pauses. Next, the algorithm tries to set the syllable boundaries equidistantly between the vowels. As there are three consonants between the first and the second vowel, the algorithm arbitrarily picks the first consonant boundary (near frame 5250) instead of the second. However, the successive processing stage finds that there is continuous pitch contour that would be cut by this segmentation, while there is no pitch detected at the boundary of the second consonant. The syllable boundary is shifted (indicated by the red arrow) to suppress large gaps in the pitch contour within one suprasegmental unit. The length of the obtained syllable segment is also used as a single duration feature.

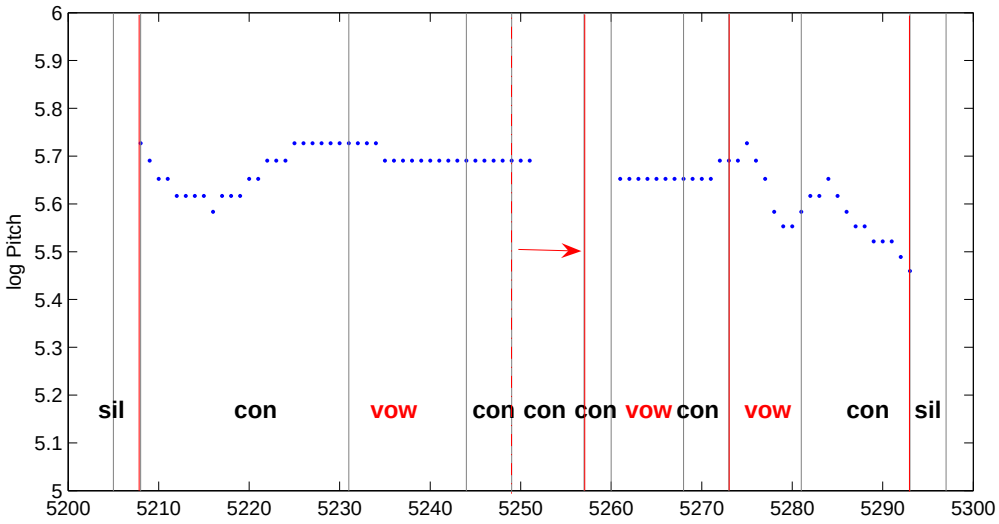


Figure 2.1: Pseudo-syllable generation from vowels and consonants. Each vowel is considered as the nucleus of a syllable. Preceding consonants as onset and successive consonants as coda.

While the algorithm itself is quite simple, it still needs a complex phone recognizer incorporating cepstral features. As a second approach, it is proposed to simply model the contours of pitch and energy over a fixed window size. As such a segmentation does not rely on any data driven assumption where to define the suprasegmental units, it is worked with highly overlapping windows and a window size that corresponds to an estimated average of the syllable length. This way, highly correlated and maybe redundant feature frames are generated, many more than for the non-overlapping and exclusive segmentation in the latter approach. As this approach is somehow similar to the extraction of MFCC with a fixed and overlapping analysis window, it is expected that the successive statistical modeling technique of GMMs learns the relevant information and can benefit from the increased number of features per utterance.

In this case, the number of voiced frames within the analysis window is used as a duration feature.

2.1.3 Contour approximation

Eventually, the extracted pitch and energy measures should be presented in the context of each suprasegmental unit. To be able to feed these prosodic features to a statistical model like a GMM, a fixed size representation for each variable sized suprasegmental unit is needed. For this purpose, some sort of curve-fitting algorithm seems appropriate that best fits a combination of different polynomials of different degrees to the original trajectory in a least-squared-

error sense. This way, it can capture the continuous contour by simply keeping the coefficients corresponding to the polynomial basis functions.

In [19] it is proposed to fit the energy and pitch contours extracted over a suprasegmental unit by a curve fitting based on Legendre polynomials [1]. The advantage over simpler polynomials is, that they are defined by orthogonal basis functions, resulting in decorrelated coefficients. As the Legendre polynomial is only defined in the interval of -1 to 1, all pitch and energy measures for the suprasegmental units need to be mapped to this interval first.

Here, it is proposed to simply apply Discrete Cosine Transformation (DCT) to the extracted pitch and energy values $x(n)$ extracted for each suprasegmental unit of length N:

$$y(k) = w(k) \sum_{n=1}^{N} x(n) \cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right) \tag{2.1}$$

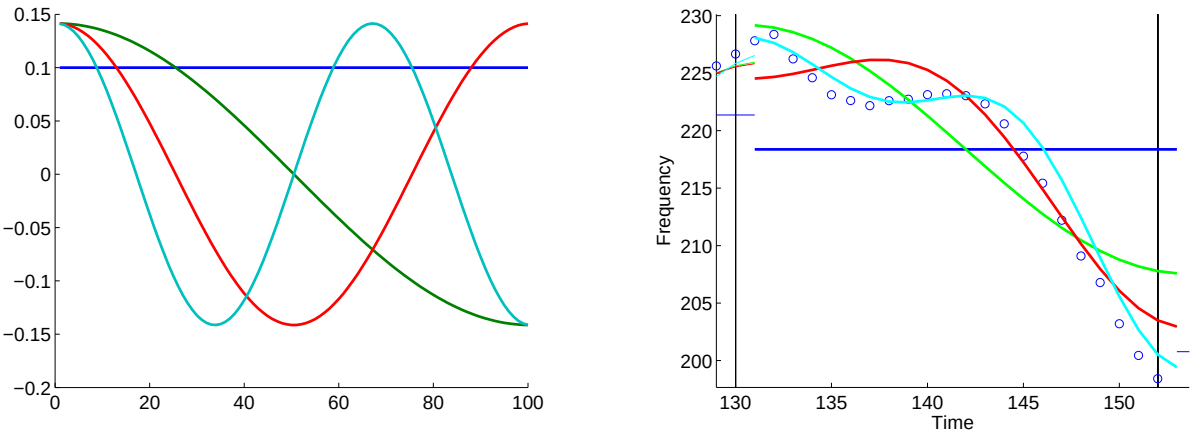
with $k = 1, 2, \ldots N$ and

$$w(k) = \begin{cases} \frac{1}{\sqrt{N}} & k = 1 \\ \sqrt{\frac{2}{N}} & 2 \leq k \leq N \end{cases} \tag{2.2}$$

Taking the Inverse Discrete Cosine Transformation (IDCT) of all coefficients $y(k)$ would result in perfect reconstruction of each pitch or energy contour extracted for each variable sized suprasegmental unit. However, taking only a fixed number of the leading DCT coefficients results in an approximated curve for each segment.

Figure 2.2.a shows the first four orthogonal DCT basis functions that are used to transform the original pitch and energy values. Figure 2.2.b shows an excerpt of a pitch contour. The solid lines show how the contours can be approximated by using only the first (blue) up to the first four (cyan) DCT coefficients.

This way, each variable sized pitch or energy contour can be translated to a fixed sized parametric representation. Similar to the Legendre polynomials, the coefficients correspond to the mean, slope, curvature and fine details of the original contour. This becomes clear when observing the first DCT basis as plotted in Figure 2.2.a.



(a) DCT basis functions

(b) Contour approximation by DCT

Figure 2.2: Approximation of pitch contour by first four DCT basis functions.

2.1.4 Final feature vector

Figure 2.3 shows how the final feature frames are constructed per syllable-like unit in the utterance. The segmentation boundaries determine the length of the segment which is stored in the feature vector as a single discrete number. Next, the first n DCT coefficients (four in the example) are stored for the pitch as well as for the energy contour. So, for each syllable we obtain a $2n+1$ dimensional feature vector.

2.2 SNERF features

In the second phase of this thesis, we use SNERFs, which are syllable-based prosodic features based on estimated pitch, energy, and duration information. Characteristics like minimum, maximum, mean, and slope of pitch and energy trajectories are extracted for each detected syllable in an utterance and its nucleus, as well as duration of onset, nucleus, and coda of the syllable. All values are further normalized with different techniques and form several hundred features for each syllable. The used syllable segmentation is generated from the output of a large-vocabulary continuous speech recognition (LVCSR) system using a simple maximum onset algorithm (Section 3.4.1 of [9]) on the phone-level alignments. Detailed information on SNERFs is given in [27].

We use 182 basic features that are extracted for each syllable. Furthermore,

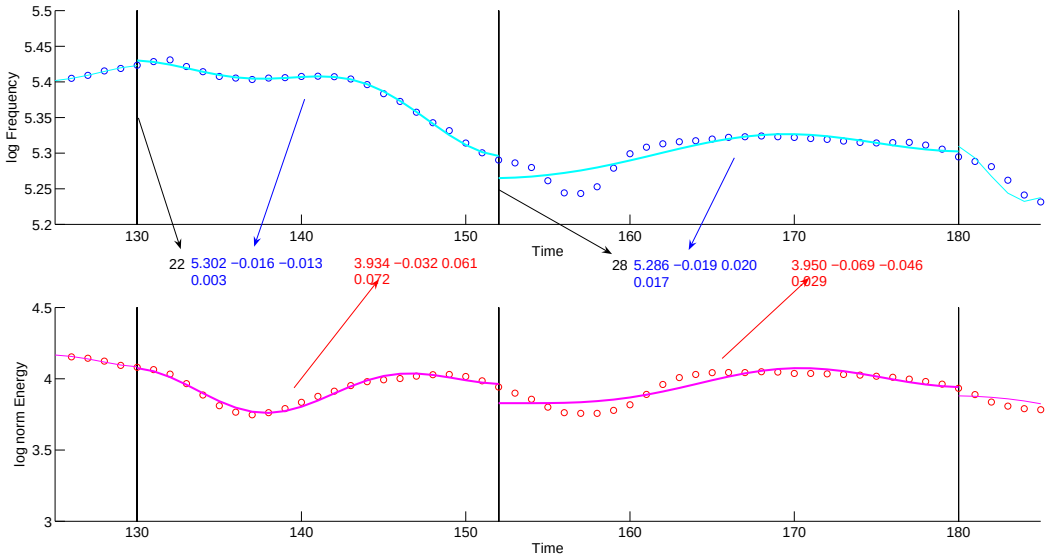


Figure 2.3: The final feature vector consists of one value for duration and the n leading coefficients per pitch and energy contour.

temporal dependencies are modeled by constructing small vectors concatenating features from consecutive syllables and pauses. These so-called tokens are formed for each basic feature by concatenating as many as three values (feature values and duration of pauses; more details are given in [11]). Nine different n-gram tokens are used.

The first line of plots in Figure 3.1 shows an example of the feature extraction process. The segments are given by the syllables found from the ASR output. The pitch (blue curve) and energy (red curve) signals are estimated from the waveform. For our example, we assume that we extract only three features per segment: its duration (from one vertical black line to the next), the mean pitch value (blue squares), and the mean energy value (red stars).

The SNERFs are parameterized by use of GMMs. This can be seen as a soft binning of each SNERF value into a meaningful set of discrete classes and makes it possible to accumulate soft counts for all SNERFs and tokens extracted for one utterance (for details see [11]).

The second line of Figure 3.1 shows a toy example in which three small GMMs are trained on a background data set. A two-component model is trained for the syllable durations, a three-component model for mean pitch values, and a four-component GMM for means of syllable energies.

The values from the exemplified feature extraction process (syllable duration, mean pitch, and mean energy) are further depicted as bars in middle

row of Figure 3.1. The occupation counts (numbers next to mixtures) are the responsibilities for each Gaussian component to generate these values. Each Gaussian component can be seen as a discrete class (nine in total, including Gaussians from the three GMMs) and the occupation counts can be seen as soft-counts of discrete events.

Chapter 3

Subspace models for prosodic features

The basic assumption in subspace modeling is that the natural parameters of a model usually live in a much smaller subspace than the full parameter space. This subspace can be learned by introducing latent variables in the model.

3.1 iVectors based on GMMs

The classical formulation of JFA for speaker verification [14] assumes that the concatenated mean vectors $\boldsymbol{\phi}_{\text{GaussJFA}}$ of a GMM are distributed according to a subspace model with separate subspaces for speaker and channel variability:

$$\boldsymbol{\phi}_{\text{GaussJFA}} = \mathbf{m} + \mathbf{V}\mathbf{y} + \mathbf{U}\mathbf{x}, \tag{3.1}$$

where $\mathbf{m}$ is a speaker- and channel-independent supervector, and $\mathbf{V}$ and $\mathbf{U}$ span linear subspaces (for speaker and channel variability) in the original mean parameter space. The components of $\mathbf{y}$ and $\mathbf{x}$ are the low-dimensional latent variables corresponding to the speaker and channel subspaces.

A simplified variant of JFA [8] assumes that speaker and channel subspaces are not decoupled and uses only one subspace covering the total variability in an utterance:

$$\boldsymbol{\phi}_{\text{GaussIV}} = \mathbf{m} + \mathbf{T}\mathbf{w}. \tag{3.2}$$

Again, $\mathbf{T}$ spans a linear subspace in the original mean parameter space and the components of $\mathbf{w}$ are the low-dimensional latent variables corresponding to the total variability subspace. The low-dimensional vectors $\mathbf{w}$ are also known as iVectors.

In the latter approach, the JFA-like model serves only as the extractor of the vectors $\mathbf{w}$, which can be seen as low-dimensional fixed-size representations of utterances, and which are in turn used as inputs to another classifier.

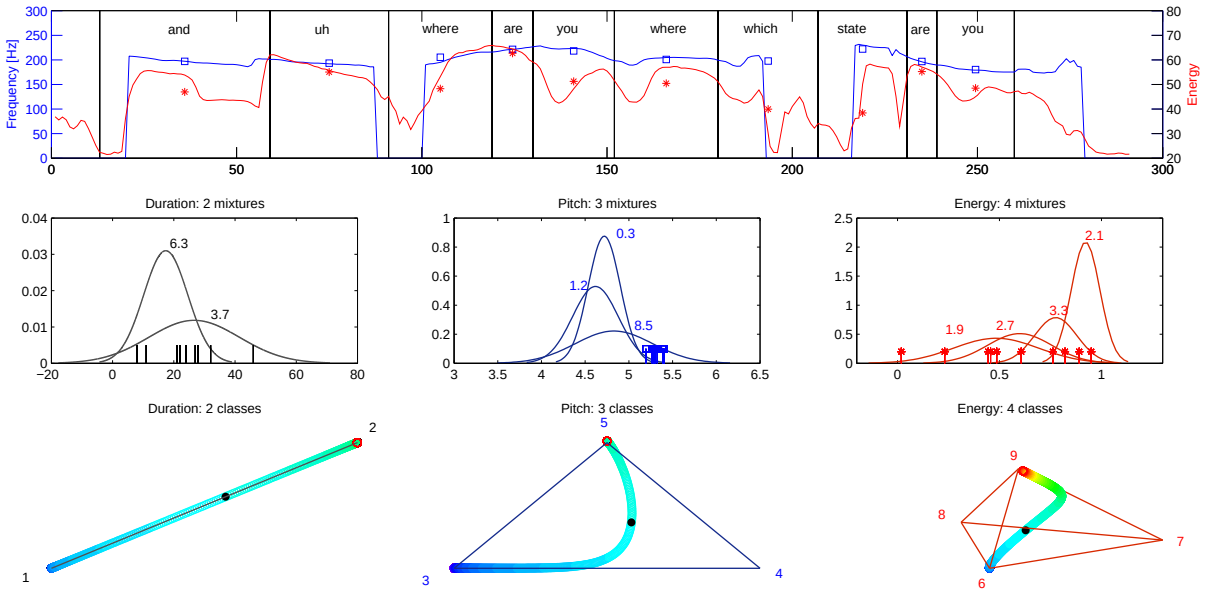


Figure 3.1: **Top row:** Extraction of three SNERF parameters from a speech segment containing 10 single-syllable words: Syllable duration (determined by black vertical lines), mean pitch value per syllable (blue squares), and mean energy per syllable (red stars). **Middle row:** Parameterization of SNERF sequences: Small GMMs are trained on background data for each individual SNERF. Two mixtures are used for duration, three mixtures for pitch, and four mixtures for energy. Occupation counts for the values extracted in the top row (here as bars) are collected using the GMMs. **Bottom row:** Multinomial model spaces for duration, pitch, and energy. The colored lines show various one-dimensional iVectors (the values are mapped to colors) projected to the full ensemble of multinomial spaces.

Both techniques, the JFA (*GaussJFA*) as well as the iVector modeling (*GaussIV*), are applicable to mean supervectors of GMMs trained on the low-dimensional well-defined DCT features as presented in Section 2.1. All model parameters are trained using an expectation-maximization (EM) algorithm [14].

3.2 iVectors based on multinomial distributions

We propose a novel subspace modeling approach for multinomial distributions, applicable to the parameterized SNERFs. In our proposed approach, we combine the advantage of the JFA-like subspace model with the flexibility of representing prosodic features as the super-vector of occupation counts. Since the occupation counts can be seen as counts of discrete events - a component generating a frame - the process of their extraction can be seen as discretization of the original prosodic features. Therefore, as a generative model, multinomial

distribution would appear as a natural choice for modeling such counts.[1]

In our model, the super-vector of model parameters is also constrained to live in a subspace defined by (3.2). However, the super-vector of Gaussian means is replaced by a super-vector of log probabilities, which are the natural parameters of our underlying multinomial distribution. A similar idea of subspace modeling of multinomial distribution was proposed for inter-session variability compensation in phonotactic language identification in [13]. A similar model is also applied for modeling GMM weights in subspace GMM, which is a recently proposed acoustic model for speech recognition [22].

The bottom row of Figure 3.1 illustrates the multinomial model spaces for our toy example. The natural parameters for the duration model exist on a line; the pitch model parameters, in a 2D simplex; and the energy parameters, in a 3D simplex space. Note that the natural parameters for all classes, in each case, are constrained to sum up to one.

The basic idea of the Subspace Multinomial Model (SMM) [17] is to learn a low-dimensional subspace of high intersession variability within the ensembles of multinomial models. That way we can (1) reduce the number of free parameters to efficiently model differences between single utterances, and (2) learn dependencies between the individual SNERFs.

In our example, in bottom row of Figure 3.1, we use a one-dimensional subspace, depicting (by the colored lines) how the subspace restricts the possible movement in the individual higher-dimensional model spaces.

3.2.1 Likelihood function

The log-likelihood of data $\mathcal{D}$ for a multinomial model with C discrete classes is determined by model parameters ϕ and sufficient statistics γ, representing the occupation counts of classes for all N utterances in $\mathcal{D}$:

$$\log p(\mathcal{D}) = \sum_{n=1}^{N} \sum_{c=1}^{C} \gamma_{nc} \log \phi_{nc}, \tag{3.3}$$

where γ_{nc} is the occupation count for class c and utterance n and ϕ_{nc} are probabilities of (utterance dependent) multinomial distribution, which is defined

[1] More precisely, there would be a set of multinomial distributions, one for each GMM in the ensemble. For each frame, each GMM is expected to generate a feature by one of its components. This corresponds to co-occurring events that has to be modeled by separate multinomial distributions.

by a subspace model according to Equation 3.2:

$$\phi_{nc} = \frac{\exp(m_c + \mathbf{t}_c \mathbf{w}_n)}{\sum_i^C \exp(m_i + \mathbf{t}_i \mathbf{w}_n)}, \tag{3.4}$$

where $\mathbf{t}_c$ is the c-th row of subspace matrix $\mathbf{T}$ and $\mathbf{w}_n$ is an r-dimensional column vector (i-vector) representing speaker and channel of utterance n.

3.2.2 Parameter re-estimation

The model parameters are obtained by maximum likelihood (ML) estimation. First, the subspace parameters $\mathbf{m}$ and $\mathbf{T}$ need to be estimated from training data. This is an iterative process, where we alternate between estimating subspace parameters $\mathbf{m}$ and $\mathbf{T}$ with fixed i-vectors, and estimating i-vectors $\mathbf{w}_n$ (one for each training utterance) with fixed subspace parameters. Even with fixed subspace parameters, there is no closed-form solution for ML update of i-vectors, and each i-vector must be updated using an nonlinear optimization technique, which is again an iterative procedure. Likewise, there is no closed-form solution for ML update of subspace parameters with fixed i-vectors. The updates we have adopted in our implementation are based on updates used for subspace GMM [22]. Vectors $\mathbf{w}_n$ are updated as

$$\mathbf{w}_n^{new} = \mathbf{w}_n^{old} + \mathbf{H}_n^{-1} \mathbf{g}_n, \tag{3.5}$$

where $\mathbf{g}_n$ is the gradient of the log likelihood function

$$\mathbf{g}_n = \sum_{i=1}^{C} \mathbf{t}_i^T (\gamma_{ni} - \phi_{ni}^{old} \sum_j^C \gamma_{nj}) \tag{3.6}$$

and $\mathbf{H}_n$ is an $r \times r$ matrix

$$\mathbf{H}_n = \sum_{i=1}^{C} \mathbf{t}_i^T \mathbf{t}_i \max(\gamma_{ni}, \phi_{ni}^{old} \sum_j^C \gamma_{nj}), \tag{3.7}$$

where ϕ_{ni}^{old} refers to the multinomial distribution (3.4) defined by the parameters from the preceding iteration. Note that the matrix $\mathbf{H}_n$ can be interpreted as an approximation to the Hessian matrix and the update formula (3.5) can be then seen as a Newton-Raphson update. The rows of matrix $\mathbf{T}$ are updated as

$$\mathbf{t}_c^{new} = \mathbf{t}_c^{old} + \mathbf{H}_c^{-1} \mathbf{g}_c, \tag{3.8}$$

where $\mathbf{g}_c$ is the gradient of the log likelihood function

$$\mathbf{g}_c = \sum_{n=1}^{N} (\gamma_{nc} - \phi_{nc}^{old} \sum_{i=1}^{C} \gamma_{ni}) \mathbf{w}_n^T \tag{3.9}$$

and $\mathbf{H}_c$ is an $r \times r$ matrix

$$\mathbf{H}_c = \sum_{n=1}^{N} \max(\gamma_{nc}, \phi_{nc}^{old} \sum_{i=1}^{C} \gamma_{ni}) \mathbf{w}_n \mathbf{w}_n^T. \tag{3.10}$$

The updates for both $\mathbf{w}_n$ and $\mathbf{T}$ may fail to improve likelihood by making too large an update step. In the case of such failure, we start halving the update step until an increase in likelihood is obtained. We have not provided any formula for updating vector $\mathbf{m}$. However, this can be simulated by fixing one of the coefficients in vectors $\mathbf{w}_n$ to be one and regarding the corresponding column of matrix $\mathbf{T}$ as the vector $\mathbf{m}$.

So far, we considered only subspace modeling of single multinomial distribution in our equations. However, for the prosodic features extracted by the ensemble of GMMs, the occupation counts should be modeled by a set of multinomial models, one for each GMM. We consider these to be concatenated into single super-vector of multinomial distributions, which is modeled by one subspace matrix $\mathbf{T}$. In other words, there will be only one i-vector $\mathbf{w}_n$ defining the whole set of multinomial distributions for each segment n. To achieve this, the indices c from Equation (3.4) must be divided into subsets, where each subset corresponds to mutually exclusive events (counts from one GMM). Then, the only difference will be in the denominator of (3.4), where we normalize only over the appropriate subset of indices that the current c belongs to. After the subspace parameters are estimated on training data, the model can be used to extract i-vectors $\mathbf{w}_n$ for all enrollment, test and background utterances using the same update formulae (3.5-3.7).

3.2.3 Model initialization

While Section 3.2.2 is quite general, the model initialization is described here more specifically for the used system. First, we estimate multinomial distributions for individual GMMs from the ensemble using all training utterances. This corresponds to summing all training super-vectors of occupation counts and normalizing the resulting super-vector over the ranges corresponding to individual GMMs. We will denote such super-vector of multinomial distributions as $\mathbf{sv}_{UBM}$. The vector $\mathbf{m}$ is simply initialized to a log of $\mathbf{sv}_{UBM}$. Note

that we did not observe any advantage from its further retraining using the updates from the previous section. All vectors $\mathbf{w}$ are initialized with zero. To ensure a good starting point, the subspace matrix $\mathbf{T}$ is initialized to represent the most important directions in the space of model parameter super-vectors. $\mathbf{T}$ is initialized by eigenvectors of covariance matrix computed from smoothed utterance super-vectors $\mathbf{sv}_n$ centered around the vector $\mathbf{m}$. The vectors $\mathbf{sv}_n$ are computed per component as

$$sv_{nc} = \log(\alpha \frac{\gamma_{nc}}{f_{nc}} + (1 - \alpha) sv_{UBM_{nc}}), \tag{3.11}$$

where f_{nc} is the number of feature frames seen for the utterance and for the GMM that the occupation count γ_{nc} corresponds to. The smoothing constant $\alpha = 0.9$ ensures that we do not take log of zero for classes that have not been occupied at all by any frames of utterance n.

3.3 PLDA modeling of iVectors

The fixed-length iVectors extracted per utterance (from the *GaussIV* as well as from the *MultinIV* model) can now be used as input to a pattern recognition algorithm. Note that unlike in the standard JFA, where two subspaces are used to account for speaker and intersession variability, the iVector variant uses a single subspace accounting for all the variability. Therefore, the extracted vectors $\mathbf{w}$ are not free of channel effect, and intersession compensation must be eventually considered during classification.

For verification of speaker trials we use a special case of Probabilistic Linear Discriminant Analysis (PLDA) [23], a two-covariance model, providing a probabilistic framework where speaker and intersession variability in the iVectors is modeled using across-class and within-class covariance matrices $\mathbf{\Sigma}_{ac}$ and $\mathbf{\Sigma}_{wc}$. We assume that latent vectors $\mathbf{s}$ representing speakers are distributed according to

$$p(\mathbf{s}) = \mathcal{N}(\mathbf{s}; \boldsymbol{\mu}, \mathbf{\Sigma_{ac}}) \tag{3.12}$$

and for a given speaker $\mathbf{y}$ the iVectors are distributed as

$$p(\mathbf{w}|\mathbf{s}) = \mathcal{N}(\mathbf{w}; \mathbf{s}, \mathbf{\Sigma_{wc}}). \tag{3.13}$$

Figure **??** exemplifies this assumption by a toy example in 2D iVector space. The dots represent several iVectors extracted from several utterances stemming from four different speakers. It can be observed, that the four solid dots, representing the explicit speakers means s, are Gaussian distributed with mean

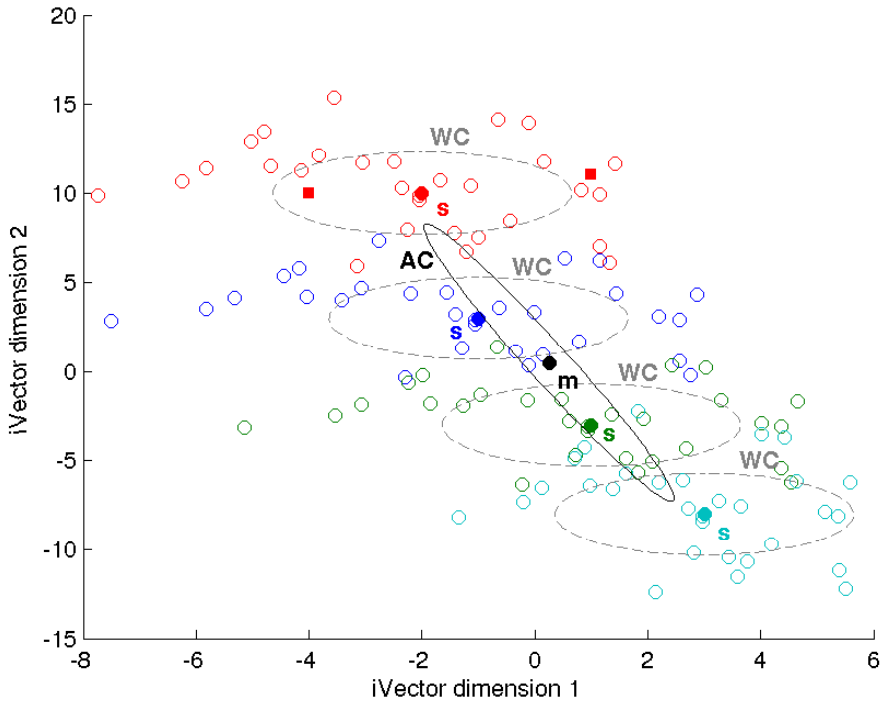


Figure 3.2: LDA assumption is iVector space.

$\mathbf{m}$ and covariance $\mathbf{\Sigma}_{ac}$. Further, the individual iVectors per speaker are also Gaussian distributed with its mean $\mathbf{s}$ and a globally tied covariance $\mathbf{\Sigma}_{wc}$. Using this model, it becomes clear that two new data points (red squares) will be identified as belonging to the same speaker although they are quite far apart from each other. As we use a probabilistic model, the parameters can be estimated using an EM algorithm and the model can be directly used to compute likelihoods.

Model parameters $\boldsymbol{\mu}$, $\mathbf{\Sigma}_{ac}$ and $\mathbf{\Sigma}_{wc}$ are trained using an EM algorithm [15]. Using the PLDA model, one can directly evaluate the log-likelihood ratio for the hypothesis test corresponding to "the two iVectors were generated by the same speaker or not":

$$s = \log \frac{\int p(\mathbf{w}_1|\mathbf{s})p(\mathbf{w}_2|\mathbf{s})p(\mathbf{s})d\mathbf{s}}{p(\mathbf{w}_1)p(\mathbf{w}_2)} \tag{3.14}$$

The numerator gives the marginal likelihood of producing both iVectors from the same speaker, while the denominator is the product of the marginal likelihoods that both iVectors are produced from different speakers. The integrals can be evaluated analytically and scoring can be performed very efficiently as described in [5].

Chapter 4

Experiments

This section shows selected experiments and results for the individual prosodic systems and for the combination of these systems with each other and with a baseline cepstral system.

4.1 Data

Results are presented on the telephone core conditions of the NIST Speaker Recognition Evaluations 2008 [20] (*dev*) and 2010 [21] (*eval*). Trials involve English conversational speech recorded over various telephone channels. Our development set is based on the original NIST SRE 2008 evaluation set, but was extended to include about two orders of magnitude more impostor samples, to adjust for the new DCF point. It includes 1,154 target and 1,516,837 nontarget trials. Our evaluation set corresponds to the official extended condition 5 of NIST SRE 2010 and contains 7,169 target and 408,950 nontarget trials.

Training of background, subspace, and PLDA models is performed on data from Switchboard corpora as well as NIST SRE 2004 – 2006 corpora. This set includes 13,482 recordings from 752 male and 16,782 recordings from 963 female speakers.

4.2 Prosodic systems

Experiments are carried out to evaluate the performance of the iVector modeling approach for the simple DCT features. For both, the *GaussJFA* and the *GaussIV* systems, we extract 13-dimensional DCT contour features (1 duration, 6 pitch and 6 energy values) and train gender-dependent multivariate universal background models (UBMs) with 512 Gaussian components and diagonal covariances. The *GaussJFA* and the *GaussIV* models are trained using

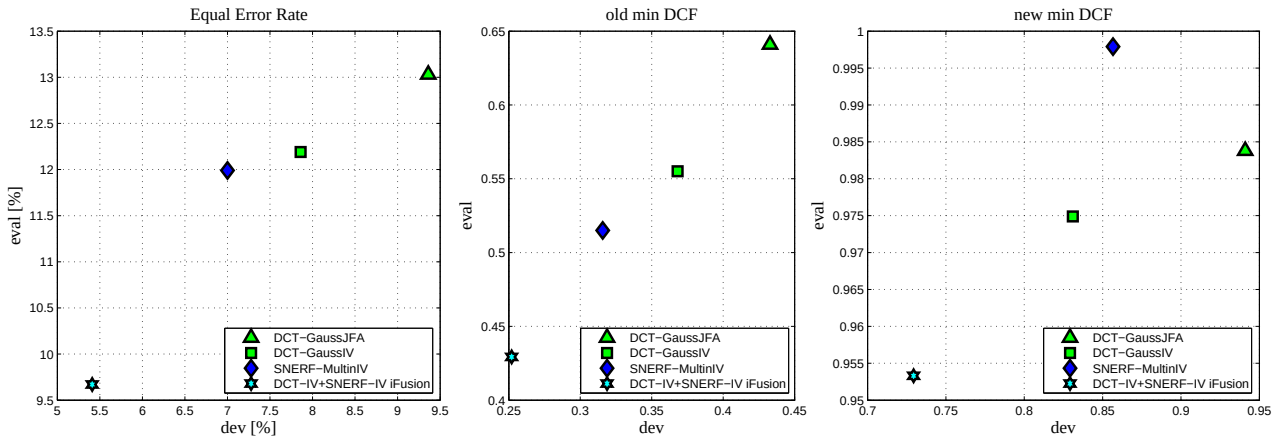


Figure 4.1: Results for SRE 2008 (dev) versus SRE 2010 (eval) in terms of EER, old DCF and new DCF, from left to right, for three different prosodic systems and combination of the two best.

sufficient statistics extracted for all background data using the same UBMs. For the *GaussJFA* model we train 100-dimensional speaker subspace $\mathbf{V}$ and 50-dimensional channel subspace $\mathbf{U}$. For the *GaussIV* model we train 300-dimensional total variability subspace $\mathbf{T}$ on the same data. These subspace sizes were found optimal in earlier experiments. The *GaussJFA* model is evaluated directly by log-likelihood ratio using a fast scoring technique [12] followed by zt-norm. The extracted DCT iVectors for all background data are used to train a full rank PLDA model. The PLDA model is then used to evaluate the log-likelihood ratio for speaker trials. Figure 4.1 shows results for the two DCT-based systems (green markers). The *DCT-GaussIV* system with PLDA (square) clearly outperforms the *DCT-GaussJFA* system (triangle) on all operating points on both test sets.

To compare the simple *DCT-GaussIV* system with the best prosodic system presented so far [18], we train a *SNERF-MultinIV* system on the same setup. The SMM models an ensemble of 1,638 multinomial distributions representing 9 different n-gram tokens of 182 individual SNERFs. We obtain 300 dimensional iVectors. While the *SNERF-MultinIV* system (blue diamonds in Figure 4.1) is still superior on both test sets for EER and old DCF, we achieve better results with the *DCT-GaussIV* system on both test sets in terms of new DCF.

As both prosodic systems perform very well, but are significantly different in terms of features as well as modeling approach, a combination of both seems natural. Since both modeling techniques translate the long-temporal prosodic feature vectors of variable size to a single fixed-length feature vector

System	DEV SRE 2008			EVAL SRE 2010		
	EER	oDCF	nDCF	EER	oDCF	nDCF
Cepstral iVector system `CEP-iV`	2.02	0.90	4.71	3.14	1.55	5.04
Concatenated `CEP+DCT-iV`	1.69	0.80	4.00	2.72	1.36	4.31
Concatenated `CEP+SNERF-iV`	**1.65**	0.80	3.89	2.74	1.34	4.44
Concatenated `CEP+DCT+SNERF-iV`	1.70	**0.75**	**3.68**	**2.63**	**1.29**	**4.21**
Score fusion `CEP-iV`, `DCT-iV` & `SNERF-iV`	1.92	0.78	4.06	3.09	1.49	4.47

Table 4.1: Results (old and new DCF $\times 10$) for single cepstral baseline system (`CEP-iV`) and for combinations with one or two prosodic iVector systems.

per utterance (what we call iVector), it is possible to simply concatenate the iVectors resulting from these diverse models and to model them jointly with a PLDA model. We train a single full-rank PLDA model on 600-dimensional iVectors. The effectivity of the joint modeling of complementary iVectors can be observed in Figure 4.1. The combination of *DCT-GaussIV* and *SNERF-MultinIV* iVectors (cyan hexagons) results in significant improvement over the best individual system on all operating points on both test sets, achieving an EER of 5.4% and a new DCF of 0.72 on 2008 data, which are (to our knowledge) the best results reported for a purely prosodic system.

4.3 Combination with cepstral baseline system

Our baseline system is a cepstral iVector system followed by a PLDA model (*CEP-GaussIV*). This system was the best-performing individual system from the ABC NIST SRE 2010 submission [4]. It is based on 60-dimensional cepstral features and a 2048-component full covariance UBM. Four hundred-dimensional iVectors are used and the dimension is further reduced to 200 by standard LDA and normalized by their length[1] before PLDA modeling. The first row of Table 4.1 gives the results for our two data sets[2].

Again, the iVector nature of our baseline system allows us to use a novel way of combining low- and high-level systems by simple concatenation of their iVectors and joint PLDA modeling. First, we apply an LDA reduction to 200 dimensions and length normalization to both 300-dimensional sets of prosodic

[1] This pre-processing of iVectors is very helpful for cepstral iVectors but did not show any improvement for our prosodic iVectors

[2] We are aware that better results are reported in the literature, simply by training the PLDA on more data, which we did not have for SNERFs.

iVectors. In this way we have three same sized sets of 200 dimensional iVectors (one cepstral and two prosodic). Next, we concatenate the cepstral iVectors separately with each of our prosodic iVectors to obtain two sets of four hundred-dimensional iVectors. Then we train a standard PLDA model with full rank of 400 for each type of combination. The second and third row of Table 4.1 give the results for these combinations. We see that we can achieve significant improvements for both *iVector fusions* of cepstral and prosodic features. Finally, we concatenate all three iVector types (one cepstral and two prosodic) and train a PLDA model with full rank of 600. The fourth row of Table 4.1 gives the results for this combination. We achieve further improvements leading to reductions as high as 21% relative on the challenging new DCF measure.

As a last experiment we compare this approach to the conventional score-level fusion. For this purpose we train a linear logistic regression [6] to fuse the three individual system scores on the development set and apply this fusion to the evaluation set. The last row of Table 4.1 indicates that consistent gains are also achieved by score-level fusion (as high as 13% on new DCF), but joint PLDA training of concatenated iVectors remains superior. iVector fusion of the cepstral system and the simple prosodic *DCT-GaussIV* system already outperforms the score-level fusion of all three systems.

Chapter 5

Conclusions and Lookout

We present the first results on the use of total variability modeling of the mean supervector space for a set of prosodic features. We show that this iVector approach outperforms the standard JFA approach originally proposed for these features. We note that this improvement over JFA is observed only when the iVectors are modeled using the PLDA back end. No gain was observed during SRE 2010 system development [4] when iVectors were modeled with simpler scoring techniques [7].

Furthermore, we present combination results of two prosodic systems, one where iVectors based on GMMs are used to model simple DCT features extracted from uniform regions and another one where iVectors based on multinomial distributions are used to model a complex set of syllable-level features. These two systems are different at both the feature and modeling levels. We show gains on the order of 20% when combining these two systems with respect to the single best. The combination is performed using an iVector-level fusion: the individual iVectors for the two systems are concatenated and the joint iVector is modeled using PLDA. An important advantage of iVector-level fusion compared to score-level fusion is that it can make use of the full information encoded in the iVectors while for the score-level fusion all information is already reduced to a single number.

The iVector-level fusion technique followed by PLDA modeling can also be applied to fuse heterogeneous features, such as low-level cepstral and high-level prosodic features. Using this procedure we achieve 20% relative improvement on new DCF over a cepstral iVector baseline, significantly outperforming score-level fusion. These are, to our knowledge, the largest relative gains obtained in speaker recognition from combination of cepstral systems with prosodic features in several years.

Bibliography

[1] M. Abramowitz and I. A. Stegun, editors. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, chapter 8: Legendre Functions, pages 331–339. 1972.

[2] A.G. Adami, R. Mihaescu, D.A. Reynolds, and J.J. Godfrey. Modeling prosodic dynamics for speaker recognition. In *ICASSP*, pages 788–791, 2003.

[3] Katarina Bartkova, David Le Gac, Delphine Charlet, and Denis Jouvet. Prosodic parameter for speaker identification. In *ICSLP*, pages 1197–1200, 2002.

[4] Niko Brummer, Luk Burget, Patrick Kenny, PaveMatejka, Edward Villiers de, Martin Karafit, Marcel Kockmann, Ondrej Glembek, Oldrich Plchot, Doris Baum, and Mohammed Senoussauoi. ABC system description for NIST SRE 2010. In *Proc. NIST 2010 Speaker Recognition Evaluation*, pages 1–20. Brno University of Technology, 2010.

[5] Lukas Burget, Oldrich Plchot, Sandro Cumani, Ondrej Glemebk, Pavel Matejka, and Niko Brummer. Discriminatively trained probabilistic linear discriminant analysis for speaker verification. In *ICASSP*, 2011.

[6] Edward de Villiers and Niko Brummer. BOSARIS toolkit. 2010.

[7] N Dehak, P Dumouchel, and P Kenny. Modeling prosodic features with joint factor analysis for speaker verification. *Audio*, Jan 2007.

[8] Najim Dehak, Patrick Kenny, R eda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING*, pages pp. 1–23, Jul 2009.

[9] L. Ferrer. *Statistical Modeling of Heterogeneous Features for Speech Processing Tasks*. PhD thesis, Stanford University, 2009.

[10] L. Ferrer, N. Scheffer, and E. Shriberg. A comparison of approaches for modeling prosodic features in speaker recognition. *Proc. ICASSP, Dallas*, 2010.

[11] L Ferrer, E Shriberg, S Kajarekar, and K Sonmez. Parameterization of prosodic feature distributions for SVM modeling in speaker recognition. *Proc. ICASSP, Taipei*, 4:233–236, 2007.

[12] O Glembek, L Burget, N Dehak, N Brummer, and P Kenny. Comparison of scoring methods used in speaker recognition with joint factor analysis. *Proc. of ICASSP, Taipei*, 2009.

[13] O. Glembek, P. Matejka, L. Burget, and T. Mikolov. Advances in phonotactic language recognition. In *Proc. Interspeech, Brisbane*, pages 743–746, 2008.

[14] P Kenny, P Ouellet, N Dehak, V Gupta, and P Dumouchel. A study of inter-speaker variability in speaker verification. *IEEE Trans. Audio*, Jan 2008.

[15] Patrick Kenny. Bayesian speaker verification with heavy tailed priors. In *Keynote presentation, Odyssey*, 2010.

[16] Marcel Kockmann, Lukas Burget, and Jan Cernocky. Investigations into prosodic syllable contour features for speaker recognition. *Proc. of ICASSP, Dallas*, pages 1–4, Sep 2010.

[17] Marcel Kockmann, Lukas Burget, Ondrej Glembek, Luciana Ferrer, and Jan Cernocky. Prosodic speaker verification using subspace multinomial models with intersession compensation. In *Proc. Interspeech, Tokyo*, 2010.

[18] Marcel Kockmann, Luciana Ferrer, Lukas Burget, Elizabeth Shriberg, and Jan Cernocky. Recent progress in prosodic speaker verification. In *Proc. ICASSP, Prague*, 2011.

[19] C-Y. Lin and H-C. Wang. Language identification using pitch contour information. *Proc. ICASSP 2005, Philadelphia, PA*, pages 601–604, 2005.

[20] NIST. The NIST year 2008 speaker recognition evaluation plan. 2008.

[21] NIST. The NIST year 2010 speaker recognition evaluation plan. 2010.

[22] D. Povey and Lukas Burget. The subspace gaussian mixture model – a structured model for speech recognition. *Computer Speech and Language*, 2010.

[23] Simon J. D. Prince. Probabilistic linear discriminant analysis for inferences about identity. In *ICCV*, 2007.

[24] U.D. Reichel. Data-driven extraction of intonation contour classes. In *Proc. 6th ISCA Workshop on Speech Synthesis, Bonn*, pages 240–245, 2007.

[25] DA Reynolds, TF Quatieri, and RB Dunn. Speaker verification using adapted Gaussian Mixture Models. *Digital Signal Processing*, 10(1-3):pp. 19–41, 2000.

[26] Petr Schwarz, Pavel Matejka, and Jan Cernocky. Hierarchical structures of neural networks for phoneme recognition. *Proceedings of ICASSP 2006, Toulouse*, pages pp. 325–328, Mar 2006.

[27] E Shriberg, L Ferrer, S Kajarekar, and A Venkataraman. Modeling prosodic feature sequences for speaker recognition. *Speech Communication*, Jan 2005.

[28] Kemal Soenmez, Elizabeth Shriberg, Larry Heck, and Mitchel Weintrau. Modeling dynamic prosodic variation for speaker verification. In *ICSLP*, pages 3189–3192, 1998.

[29] David Talkin. *Speech Coding and Synthesis*, chapter 14: A Robust Algorithm for Pitch Tracking (RAPT). 1995.

[30] Ingo R. Titze. *Principles of voice production.* National Center for Voice and Speech, 2000.

Author

Dipl.-Ing. Marcel Kockmann
http://www.fit.vutbr.cz/~kockmann

Marcel was born on May 23, 1981 in Nürnberg, Germany. He received his Master's degree in Media Technology from the Ilmenau University of Technology, Germany, in 2007. The topic of his diploma project was the use of phonetic features for speaker verification, carried out at Siemens Professional Speech Processing Group in Munich.

Continuing his work on speaker verification at Siemens, he joined the Speech group at Brno University of Technology as an external Ph.D. student in September 2007. In this collaboration, he worked on enhancing the Siemens speaker verification technology by implementing state-of-the-art techniques. Further, the main topic of his thesis is the usage of prosodic information for speaker verification.

He subsequently worked on AMIDA, MOBIO and IARPA projects on the topic of prosodic speaker verification. He participated within the Speech@FIT group at NIST SRE 2008 and 2010 evaluations. Further, he built the winning systems for Interspeech 2009 Emotion Challenge and Interspeech 2010 Paralinguistic Challenge.

During his graduate studies, Marcel has authored and co-authored more than ten conference and journal papers presented on international events and magazines. He won the Special Session's Best Paper Prize at Interspeech Emotion Challenge 2009, and was nominated twice for the Best Student Paper Award at Interspeech conferences. He is member of the IEEE and ISCA.