Vysoké učení technické v Brně
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií

Dr. Ing. Jan Černocký

Časové zpracování pro výpočet příznaků v rozpoznávání řeči

Temporal processing for feature extraction in speech recognition

Teze habilitační práce

Brno 2003

Klíčová slova: automatické zpracování řeči, rozpoznávání řeči, příznaky pro rozpoznávání řeči, časová filtrace, neuronové sítě, daty řízené techniky.

Keywords: automatic speech processing, speech recognition, features for speech recognition, temporal filtering, neural networks, data-driven techniques.

Rukopis habilitační práce je uložen na Fakultě informačních technologií Vysokého učení technického v Brně, Božetěchova 2, 61266 Brno. Plný text habilitační práce je k disposici na:
`http://www.fit.vutbr.cz/~cernocky/publi/2002/habil.pdf`

ISBN 80-214-

ISSN 1213-418X

Contents

Autor

Dr. Ing. Jan Černocký (nar. 1970)
`http://www.fit.vutbr.cz/~cernocky`

Vzdělání: FEI VUT v Brně – Ing., 1993, Université Paris-Sud Orsay – studium DEA (Diplome d´Etudes Approfondies) d´Electronique, 1995, doktorská disertace současně na Université Paris-Sud Orsay (Francie) a na FEI VUT v Brně. – Dr., 1998.

Pracovní zařazení: odborný asistent a zástupce vedoucího ústavu na FIT VUT.

Odborné praxe v zahraničí: DEA, Université Paris Sud 1 rok 1994-5, ESIEE Paris (tři 6-měsíční stáže během doktorského studia) 1995-98, OGI (Oregon Graduate Institute for Science and Technology), Portland, USA, 7 měsíců 2001.

Odborné zájmy: Výzkum J. Černockého je zaměřen na číslicové zpracování řečových signálů. Nyní se zaměřuje na problematiku kódování řeči na velmi nízkých bitových rychlostech, rozpoznávání řeči a tvorbu řečových databází.

Výzkumné projekty: Byl řešitelem 2 grantových projektů VUT a spoluřešitelem projektu MŠMT "Posílení vědy a výzkumu na vysokých školách". Byl českým koordinátorem projektu "SpeechDat-E: Eastern European Speech Databases for Creation of Voice Driven Teleservices" v rámci programu 4th PCRD. Je zapojen ve dvou projektech v rámci 5th PCRD: "Multimodal Meeting Manager - M4" a "Speech driven interfaces for consumer devices - SpeeCon" a podílí se na přípravě dvou projektů (Network of Excellence a Integrated Project) v rámci 6th PCRD. Je spoluřešitelem rámcového projektu GAČR "Hlasové technologie v podpoře informační společnosti" a řešitelem post-doktorského projektu GAČR "Antropické a daty řízené rozpoznávání a kódování řeči".

Členství v organizacích: IEEE (sekretář Československé sekce), ISCA (International Speech Communication Association), Společnost pro radioelektronické inženýrství. Člen redakční rady časopisu Radioengineering a člen programového výboru konference TSD (Text–Speech–Dialogue).

Od roku 1999 je školitelem doktorandů. V současné době vede 7 studentů řádného doktorského studia, obhajoba L. Burgeta a P. Motlíčka je plánována na přelom léta a podzimu 2003.

Chapter 1

Introduction

This thesis is proposed in order to obtain the "associate professor" degree at Brno University of Technology, Faculty of Information Technology. Its main topic are temporal and data-driven methods for feature extraction in speech recognition. Rather than an individual research work, this is a topic worked on by several of my pre- and post-graduate students as well as students in our partner laboratory at OGI Portland.

1.1 Acknowledgements

The author would like to thank his tutors in speech and signal processing, starting at VUT Brno with Vladimír Šebesta and Milan Sigmund, through Geneviève Baudoin and Gérard Chollet at ESIEE and ENST Paris, till Hynek Hermansky (OGI Portland and VUT Brno). Second series of thanks goes to postgraduate students at the Dpt. of Computer Graphics and Multimedia (from seniors to juniors: Lukáš Burget, Petr Motlíček, Franta Grézl, Petr Schwarz, Martin Karafiát, Petr Jenderka and Tomáš Vícha) at FIT and Inst. of Radioelectronics of FEEC (Pavel Matějka); some of them have contributed important portions to this text: part of the introductory material on HMM recognition [4], some TRAP figures and new results [8], SpeechDat-E recognizer [26] and context-dependent HMMs [21] used in forced alignment of SpeechDat-E. But as important was to be in contact with Hynek Hermansky's students at OGI last year: Pratibha Jain, Sachin Kajarekar, Sunil Sivadas, and Andre Adami. Among the pre-graduate students, Jiří Kafka (graduated in June 2002) deserves great thanks for his diploma work [19] on LDA-filters tested on the recognition of Czech. His results are the core of the LDA experimental section.

The greatest thanks go of course to my wife Hanka for a steady support, and my sons Tomášek and Adámek for some distraction and lots of fun.

1.2 Scope of chapters

This work is divided into 5 chapters. The current one contains small review of speech recognition using Hidden Markov models and a section introducing speech feature extraction. Chapter 2 deals in more detail with data-driven features for speech processing. Chapter 3 concetrates on the use of Linear Discriminant analysis for filtering of temporal trajectories. Chapter 4 covers Temporal trajectory classifiers (TRAPs) as feature extractors for speech. Chapter 5 concludes this text.

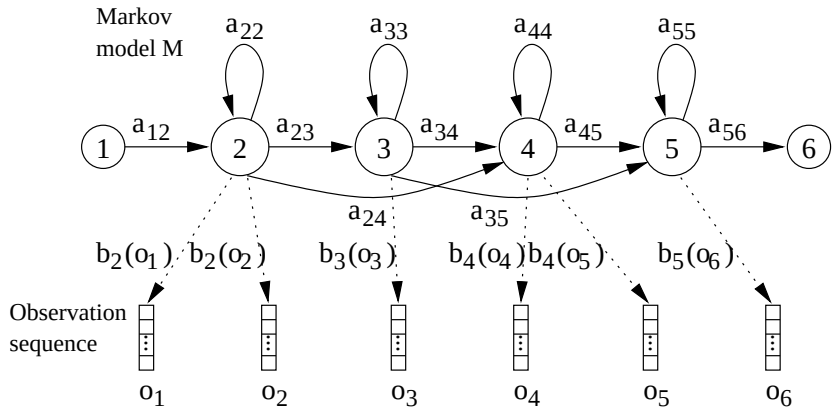

Figure 1.1: The Markov Generation Model

1.3 Speech recognition using Hidden Markov Models

Most of current systems [7] [25] for automatic speech recognition consist of three basic function blocks:

(1) **Feature Extraction -** In this phase speech signal is converted into stream of feature vectors – coefficients – which contain only that information about given utterance that is important for its correct recognition. Parameterization is performed for a size reduction of original speech signal data and for preprocessing of that signal into a form fitting requirements of following classification stage. An important property of feature extraction is the suppression of information irrelevant for correct classification, such as information about speaker (e.g. fundamental frequency) and information about transmission channel (e.g. characteristic of a microphone). Currently the most popular features are Mel frequency cepstral coefficients MFCC [5].

(2) **Classification -** The role of classifier is to find a mapping between sequences of speech feature vectors and recognized fundamental speech elements (words in a vocabulary, phonemes). This mapping can be done for example by simple recognizer based on Dynamic Time Warping (DTW), where the sequences of parameter vectors are stored as references. Word parameters are then compared directly with the references. More advanced classifiers are mostly based on Hidden Markov Models (HMM) [31], where parameters of statistical models (Fig. 1.1) are estimated using training utterances and their associated transcriptions. After this process, the well trained models can be used for recognition of unknown utterances. The output of the classifier is a set of possible sequences of speech elements (hypotheses) and their probabilities.

(3) **Language models -** The role of language models is selection of a hypothesis which is most likely the right sequence of speech elements (sentence) of a given language. The complexity of language model depends on complexity of the problem being solved (continuous speech vs. limited number of commands). Statistical models derived from data are also very often used for this purpose (N-grams). Interested reader can be referred to [18].

1.4 Features for HMM speech recognition

While the acoustical matching (HMM) and decoding (Viterbi algorithm with a pronunciation dictionary and language model) are heavily trained on data, the feature extraction is often considered as

"given" and neglected [10]. In the following chapters we will concentrate on temporal processing of features and on some pre-classification that brings the features more closed to the data.

HMMs have one substantial drawback - one HMM state "sees" only the current speech frame and it does not know, what happened before and what will happen later in the feature matrix. Delta and delta-delta features introduced in the 70-ies, have added some notion of trends in feature space. Note, that computation of Δs and $\Delta\Delta$s can be also interpreted as temporal filtering. Hermansky and Morgan introduced RASTA filtering [12], being inspired by some properties of human auditory periphery, namely by the insensibility to too high and too low frequencies in modulation spectrum. RASTA processing has made especially recognizers based on context-independent phonemes far more robust than their non-filtered counterparts.

In RASTA, the characteristics of filter were "tuned" to match some auditory properties, but it was not shown to be optimal. From the classifier theory, we dispose however of some mathematical tools to design optimal projection of parameter space in order to preserve discriminability. As linear filtering is nothing but projection of the original temporal trajectory on the reversed impulse response of the filter, a filter can be designed using this theory. In chapter 3, we will see an application of LDA (linear discriminant analysis) derived filters to the recognition of Czech.

Till now, we are however still limited to feature-preparation using linear processing. Non-linear methods, especially Neural Nets [3] have been widely used in speech processing, mostly to derive class-posteriors for the Viterbi algorithm (they are replacing mixtures of Gaussians). On the other hand, one would like to use NNs directly to derive features and not to "touch" the recognizer itself – the Gaussian mixture modeling (GMM) is nowadays the most wide-spread technique to model the distribution of features in states (this inclination is also given by the popular HTK toolkit, using GMMs). The Tandem-approach or Feature-Net overcomes this barrier by processing the feature stream by NNs, but using the output likelihoods (after some post-processing, as Gaussianization and de-correlation) as input to standard GMM-HMM recognizer [11].

The classification of Temporal Patterns (TRAPs) using NNs, first introduced by Sharma [28], is a natural step in combining the temporal processing and non-linear classification into the feature-extraction block. Unlike conventional recognizers where the recognition is done of a "instantaneous cut" of the entire speech spectrum, TRAP classifiers detect acoustical units out of long temporal trajectories in each frequency band, and then the results are combined using a merging network. We hope that if the speech is corrupted in one more frequency bands, the merger will still be able to obtain a more reliable output. This idea is similar to multi-band recognizers promoted by Bourlard and Morgan [2], but in our case, we are not forced to "touch" the recognizer's architecture – the robustness should be achieved by the feature extractor itself.

Chapter 2

Data driven features for speech processing

The purpose of feature extraction is a reduction of speech data size and other processing required for an adaptation of these data for classifier (HMM). The standard way of feature extraction consists of the following steps:

(1) **Segmentation** – Speech signal is divided to segments where the waveform can be regarded as stationary (the typical duration 25 ms). The classifiers generally assume that their input is a sequence of discrete parameter vectors where each parameter vector represents just one such segment - frame.

(2) **Spectrum** – Current methods of a feature extraction are mostly based on the short term Fourier spectrum and its changes in the time, therefore the power or magnitude Fourier spectrum is computed in the next step for every speech segment.

(3) **Auditory-like modifications** – Modifications inspired by physiological and psychological findings about human perception of loudness and different sensitivity for different frequencies are performed on spectra of each speech frame.

(4) **Decorrelation** – Some technique for vector decorrelation is used for a better adaptation of features to requirements of classifier. In the case of HMM, only a variance vector can be used for a description of output probabilities instead of a full covariance matrix.

(5) **Derivatives** – Feature vectors are usually completed by first and second order derivatives of their time trajectories (delta and acceleration coefficients). These coefficients describe changes and speed of changes of the feature vector in the time.

2.1 Mel frequency cepstral coefficients

Mel frequency cepstral coefficients (MFCC) [5] are a commonly used feature extraction method. Brief description of this method is presented here and in Fig. 2.1 because MFCC are used as starting point for modifications described in section 2.2.

First, speech samples are divided into overlapping frames. The usual frame length is 25 ms and the frame rate is 10 ms. Hamming window is applied in the next step and magnitude Fourier spectrum is computed for this windowed frame signal. A filter bank is then applied for modification of the magnitude spectrum. Energies in the spectrum are integrated by the set of a band limited triangular weighting functions. These weighting functions are equidistantly distributed over Mel scale according to psycho-acoustic findings where better resolution in spectrum is preserved for lower frequencies than for higher frequencies. A vector of the filter bank energies for one frame can be seen as a smoothed and down-sampled version of spectrum. The log of integrated spectral energies is taken with agreement to the human perception of sound loudness. The feature vector is

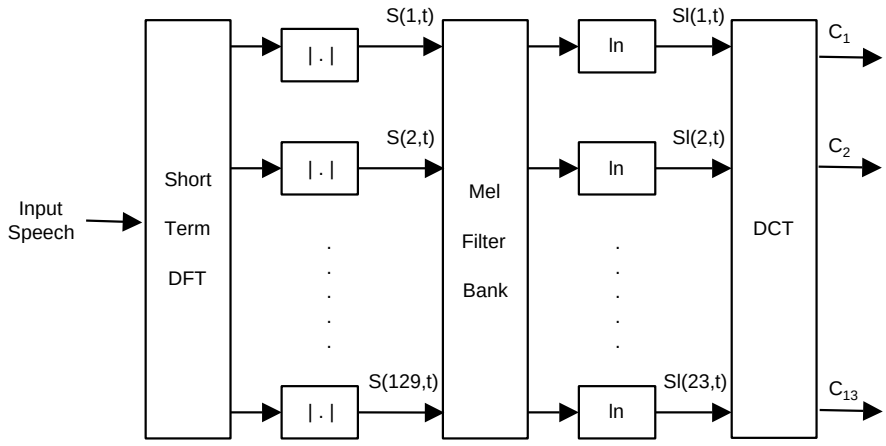

Figure 2.1: Block diagram showing steps of Mel frequency cepstral coefficients computation

finally decorrelated and its dimensionality is reduced by its projection to several first cosine basis (Discrete Cosine Transform).

2.2 Data driven feature extraction methods

While classification and language models are usually based on stochastic approaches where models are trained on data, feature extraction is generally based on knowledge and beliefs. However, since mechanism of the human auditory system is not fully understood, the optimal system for a feature extraction is not known. Moreover, psychoacoustic findings often describe limitations of the human auditory system and we do not know if modeling of those limitations is useful for the speech recognition. Therefore methods of data driven optimizations for some stages of above described standard feature extraction scheme are presented in following sections. The agreement between results of these methods and the psychoacoustic findings is shown. Principal Component Analysis and Linear Discriminant Analysis techniques are used by these optimization methods and they will be described first.

2.2.1 PCA and LDA

Principal Component Analysis (PCA) or Karhunen-Loevy transform (KLT) is a technique looking for such linear transform with orthogonal basis where the first base vector shows a direction of the largest variability of training data in N-dimensional space of input vectors. The second base vector than shows a direction perpendicular to direction given by the first vector with the second largest variability and so on. A limitation of this technique is the assumption that input data have Gaussian distribution. This transform has two important properties: (1) Elements of outputs vector are decorrelated (their values are not dependent each on other). (2) Projection to only several first base vectors can be performed for a dimensional reduction preserving most of a variability of original data.

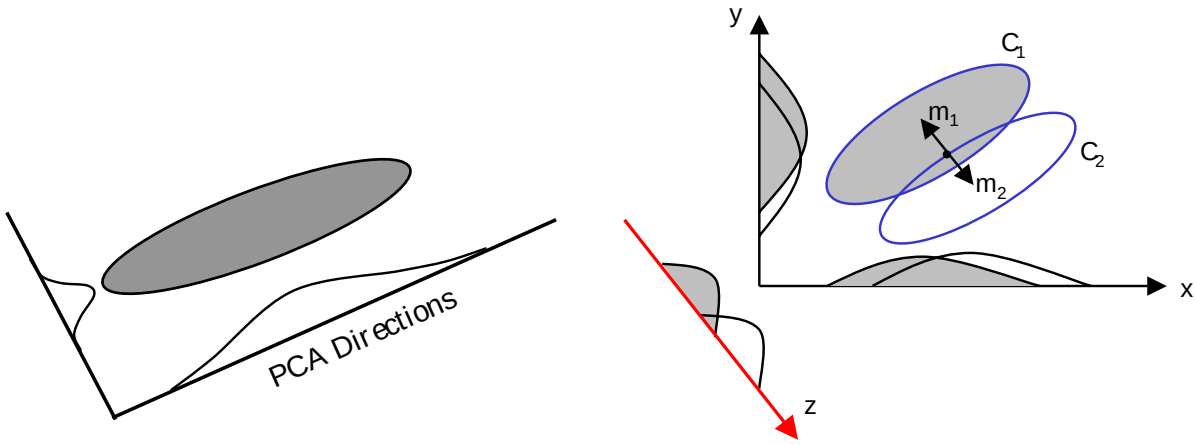

Figure 2.2: Principal Component Analysis (a) and Linear discriminant analysis (b) for 2-Dimensional Data

The figure 2.2a demonstrates the effect of PCA for 2-dimensional data vectors. The gray ellipse represents distribution of data, the axes show the new coordinates (directions) obtained by a rotation of original coordinates using PCA transform. The new distributions of uncorrelated data in both directions are also demonstrated in this figure.

Base vectors of PCA transforms are given by the eigen vectors of a covariance matrix which is computed from training data. The eigen value associated with each eigen vector represents the amount of variability preserved by the projection of input vectors to this particular eigen vector. Therefore only several eigen vectors corresponding to the highest eigen values are used as basis of PCA transform for the purpose of a dimension reduction.

Analogous to PCA, **Linear Discriminant Analysis (LDA)** proposed by Hunt [14] is a data driven technique looking for such linear transform allowing a dimension reduction of input data. However, it preserves information important for the linear discrimination among input vectors which belong to different classes. Therefore, unlike the case of PCA, we need also information about the class to which a particular input training vector belongs. The result of LDA are then base vectors of the linear transform sorted by their importance for discriminating among classes. We can therefore pick up only several first basis which preserve almost all the variability in data important for discriminability. Note, that LDA like a PCA ensures decorrelation of transformed data. Moreover, it does not decorrelate only overall training data as it is in the case of PCA, but data belonging to each particular class are also decorrelated. The figure 2.2b demonstrates effect of LDA for 2-dimensional data vectors which belong to two classes. The gray and the empty ellipses represent distributions of data in two different classes with mean vectors $\mathbf{m_1}$ and $\mathbf{m_2}$ and covariance matrices $\mathbf{C_1}$ and $\mathbf{C_2}$. The axes X and Y are coordinates of the original space. Large overlap of the class distributions can be seen in the directions of these original coordinates. The axis Z then shows the direction obtained by LDA in which the classes are well separated. Since this example deals just with two classes and since LDA assumes that distributions of all classes are Gaussian with the same covariance matrix ($\mathbf{C_1} = \mathbf{C_2}$) no other direction can be obtained for a better discrimination.

Base vectors of LDA transforms are given by the eigen vectors of a matrix $\mathbf{AC} \times \mathbf{WC}^{-1}$, where the within-class covariance matrix **WC** represents unwanted variability in data and the across-class covariance matrix **AC** represents the wanted variability.

An eigen value associated with one eigen vector represents the amount of variability necessary for the discriminability preserved by the projection of input vectors to this particular eigen vector. Only several eigen vectors corresponding to the highest eigen values can be used as LDA transform for the purpose of a dimension reduction.

2.3 From RASTA to temporal filters derived using LDA

The original design of RASTA filters [12] for filtering temporal trajectories was inspired by psychological findings about temporal masking and is the core of RASTA algorithm described in [12]. RASTA takes advantage of the fact, that the linguistic message is coded into movements of the vocal tract. It suppresses the spectral components that change more slowly or more quickly than the typical range of change of speech. Temporal trajectories of the log energies of each band are filtered by bandpass filter. The low pass character of such filter allows to remove fast energy changes which cannot be produced by the human articulatory tract. The high pass character of the filter is responsible for removing a static information about a channel, since it appears as an additive constant to the filter bank band output in the log domain. In principle, the RASTA processing can be done on time trajectories of any parameters.

Another possibility is to derive impulse responses of such filters using *Linear Discriminant Analysis*. The filters can be derived independently for each band. Vectors which are formed by consecutive values of one band time trajectory are used for the computation of across-class covariance and within-class covariance matrices. A typical length of the vector corresponds to one second of signal (101 points of a band time trajectory for 10ms frame rate). Vectors with central point representing frames labeled by the same phoneme belong to the same class.

The way how the basis (eigen vectors of $\mathbf{AC} \times \mathbf{WC}^{-1}$ matrix) are applied for data transformation (across time) corresponds to linear filtering using a convolution of a signal with a filter impulse response. In other words, every eigen vector represents one impulse response (inverted in the time) of a filter for filtering the time trajectory of one filter bank band.

Chapter 3

LDA filters for recognition of Czech

First experimental part of this text is devoted to the application of LDA-derived filters to feature extraction for Czech telephone-speech recognition.

3.1 Experimental setup

Databases *OGI Multilanguage Telephone Speech Corpus* and *Czech SpeechDat-E* (Eastern European Speech Databases for Creation of Voice Driven Teleservices) were used for tests. Detailed description of the experimental protocol is available in the full version of the habilitation thesis [30].

Recognizer and reference results Isolated word recognizer based on context-independent phoneme models defined in the diploma thesis of Petr Schwarz [26] was used in this work for evaluation of LDA filtering. The recognizer is trained on isolated, phonetically balanced words (2777 items) and tested also on isolated phonetically balanced words (1394 items). The recognition vocabulary contains 2175 words. Two flavors of training and testing were used: (1) using full-length items, including sometimes long portions of silence at the end. (2) "cut" version of the database where silence portions were stripped using GMM-based voice activity detector, boot-strapped by a simple energy-based segmentation. The diploma thesis of Petr Schwarz [26] presents this speech/silence segmentation in detail. The recognizer was built on standard HTK [31] tools. Recognition performance was evaluated using the word recognition accuracy.

To evaluate baseline performance of our recognizer, the very standard feature extraction: **MFCC**, appended with log-energy and velocity and acceleration features was tested first. The second experiment consisted in replacing the log-energy by the 0-th cepstral coefficient, a trick, that, according to experience of ASP group at OGI, improves the recognition performance in real noise conditions. This experiment provided superior results and is used as baseline in this work (MFCC_0).

3.2 Computing and use of LDA filters

Features for LDA filters The choice of features for deriving LDA filters was quite obvious: we wanted the LDA features to be as compatible with MFCC's as possible. The choice were therefore log-energies after Mel-scale filterbank, that are used as a mid-product in MFCC computation. They can be output by standard HTK tool `HCopy` using the `FBANK` feature type, they can be then processed by non-HTK software (Matlab in our case), and then converted to MFCC again by `HCopy`

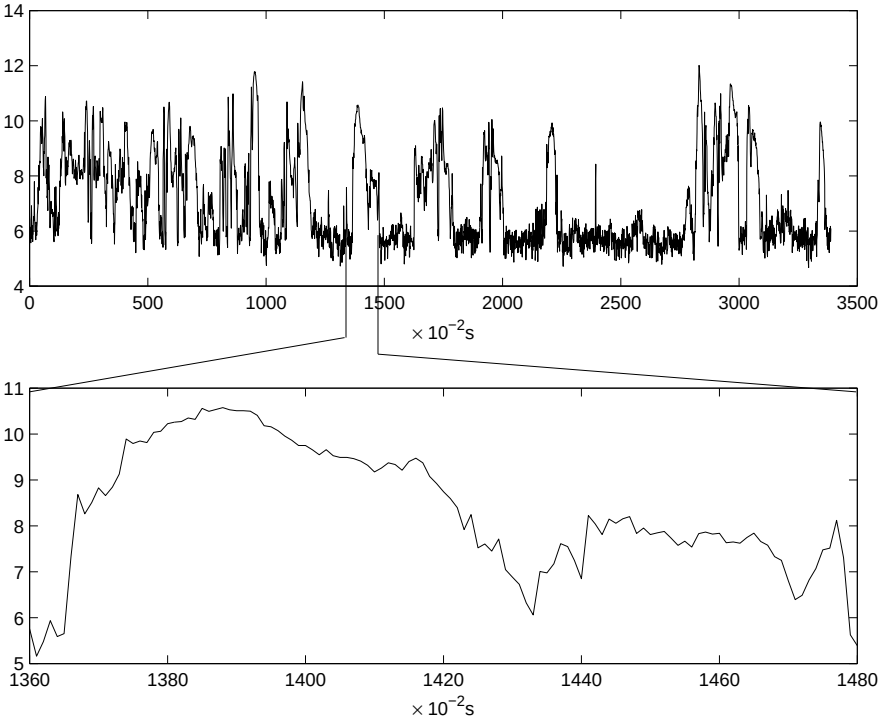

Figure 3.1: Output from 10-th Mel filter

with proper specification of the input type. The only problem we have faced was the 0-th cepstral coefficient, that had to be computed 'by hand'. The number of filters in Mel-filterbank was the HTK default: 20. Figure 3.1 shows an example of temporal trajectory in the 10-th band.

Classes for derivation of LDA filters 43 phonetic classes were available for STORIES, however, two of them were excluded: (1) the 'other' class `oth` where some rarely occurring phonemes were put and which was not very consistent. (2) the silence `pau` with its huge proportion in the database (almost 20%). This class with its broad distribution could heavily bias the within-class covariance matrix. Therefore it was discarded too and the derivation of filters was done using 41 classes.

Handling edges As everything, each speech file begins and ends. For derivation and subsequent use of LDA filters, in case we consider 1 second trajectories, we need 50 acoustical frames of left context and 50 frames of right one. There is no exact answer — experiments have shown that the optimal solution is to concatenate the the consecutive files in the database. This approach that is the most natural (usable however only in the case we dispose of all the files at the same moment) and was used in most of the recognition experiments.

Using LDA filters After computation of LDA filter, the filtering is done by

$$y(n) = \sum_{m=1}^{101} x(n+m-50)h(m),$$

where $x(n)$ is the filtered trajectory and $h(m)$ is the LDA filter. As this equation does not contain $n-m$ term that we would expect in a convolution, a more precise term would be a *projection of*

time-trajectory onto LDA eigen vector. It would be however easy to invert $h(n)$ in time and obtain proper filtering equation.

We are motivated to use velocity and acceleration parameters. There are two approaches to compute them while using LDA filtering: (1) to apply LDA filtering for band energies only with the static coefficients, then de-correlate using DCT and compute Δ and $\Delta\Delta$ at the end of processing in standard way, using approximations of first and second derivative. (2) to use LDA filters related not only to the first, but also second and third biggest eigen-values. We have seen that they present some similarities to the first and second derivative of the first filter. The outputs of all filters have to be de-correlated by DCT or PCA.

Another issue is the computation of energy-related 0-th cepstral coefficient, which is actually a sum of log-energies in bands. This can be computed from LDA-filtered energy trajectories or from original ones.

3.3 Derivation of LDA filters on STORIES

LDA filters were computed on STORIES. Leading 50 and trailing 50 vectors of each were used just as context. LDA filters were applied to filter-bank outputs without any modifications and different setups were tested depending on the dealing with edges and processing of c_0. In the following steps, different kinds of processing of "edges" of filters were tested (these edges are not reliably estimated): (1) Weighting filters by a Hann window [17], (2) Influencing first 5 and last 5 samples by a sharply raising and falling function, (3) zeroing of first and last sample.

The results, including the baseline, are summarized in Table 3.5 in [30]. All modifications do approximately as well as original filters. The "winner" is the simplest processing - just zeroing the first and last sample in each LDA filter. We have also confirmed, that the results on original database (silences included) is consistently better than on the "cut" version and that the c_0 should be computed from the original energy trajectories. The best result so far is recognition accuracy of 91.32%.

3.4 LDA filters trained on SpeechDat

So far, we have tested the recognition of Czech isolated words with features computed using LDA filters derived on US-English. As a next step, we wanted to train those filters directly on SPEECHDAT to see the influence of: (1) training of LDA filters on the target language. (2) training of LDA filters on the target database.

Unfortunately, other telephone Czech database was not at our disposal, so that we could not investigate into those two points independently. The following results will therefore present a situation, where the derivation of LDA's is very closed to the target task (language *and* channel characteristics are the same). For the training of LDA filters, the 101-point vectors with centers situated in silence `sil` (41.55% in the database) were not used for similar reasons as for STORIES (section 3.2). Interword short pauses `sp` and `oth` class were not used as well, so that the so that the number of phonetic classes was 41, accidentally the same as in previous experiments. The LDA filters were computed in the same way as in previous case.

Similar experiment as for filters derived on STORIES were conducted with the summary of results in Table 3.7 in [30]. The best result obtained with the same setup as for the STORIES training. It outperforms the baseline by more than **2%** absolute.

3.5 LDA filtering – conclusions

Conducted experiments In the experiments conducted, Mel-filterbank output trajectories were processed by LDA-derived filter filtering. The filters were first trained on US-English OGI-Stories database, then on the recognizer's target data: Czech SpeechDat-E database.

The results have clearly shown the advantage of LDA filtering even for filters derived on a different database (outperforming MFCC_0 baseline by more than 1% absolutely). The filters derived on the target data have shown superior performance, and gave more than 2% absolute improvement over the baseline.

We may conclude, that LDA filtering of temporal trajectories is a cheap and efficient way to improve the performance of simple speech recognizers. This is confirming the results the ASP group at OGI has obtained in numerous experiments during the dissertation of Sachin Kajarekar [20] and during the AURORA project [1].

Open problems and questions Even if the results were encouraging, we should not forget that there are many unresolved problems in this work necessitating further research. From the more technical to more general they include:

(1) in this work, we have not investigated the possibility to use *LDA-filters corresponding to the 2-nd and 3-rd eigenvalues* for computation of Δ and $\Delta\Delta$ approximations. When trying to use those, we should cope with the correlation of those features: (a) in MFCC, the discrete cosine transform is used to convert the estimate of log-spectrum back to temporal domain, but its main purpose is to decorrelate the features. Will the use of DCT be justified also for features filtered by the 2-nd and 3-rd filters ? (b) there is of course a possibility to train a PCA to decorrelate the features. Should this transform be trained per-stream (ie. separately for 1-st, 2-nd and 3-rd filter) or as a whole ?

(2) the current work does not make any use of feature normalization while we know, that off-line or on-line normalization dramatically improves the recognition accuracies for noise conditions [15]. On the other hand, this normalization can behave in quite unpredictable way for clean conditions, and for scenarios with some non-standard features (see for example the work of Petr Motlíček on all-pole modeling of log-spectra [23]).

(3) there are many *engineering choices* to be made while computing/implementing the LDA filters: (a) selection of classes was done by a brief investigation of how do the phoneme proportions look like. We have not at all investigated using of broad phonetic classes, that are known to perform well for TRAP-based systems [16]. Also, the ASP group at OGI reports to obtain slightly smoother filters and better results while dividing each phoneme into 3 sub-classes. The original work of Kafka [19] however reports quite discouraging results of such experiments. Kafka has argued that there is too little data to estimate the covariance matrices reliably for this method. (b) post-processing of filter coefficients was found to be quite important for good performance. However, only a few "ad-hoc" experiments were performed, based on "looking at the filter".

(4) The final point questions the very base of this work: the linear discriminant analysis LDA. The basic problem of this method is that it expects the same statistical distribution of data in all classes. This is obviously never satisfied, so that the derived filters are globally optimal, but they can perform very badly for the discrimination of some classes.

The problem is not only to find mathematical methods that relax the assumption of equal distribution of data within classes, but also to implement them and test their performances on speech recognition tasks. See [30] for the overview of current work on LDA-based features.

Chapter 4

TempoRAl Patterns – TRAPs

"Classical" features for speech processing (as MFCC's) provide information about the entire spectrum of speech signal for a very limited time (the spectrum is usually computed in frames of 20-25 ms with a window-shift of 10 ms) [31]. If noise is present in the speech signal, it affects the entire feature vector, and impairs the accuracy of the recognizer. Multi-stream approach [28, 3] overcomes this problem by running several speech recognizers independently in different frequency bands, and recombining their results. The recognition in bands and recombination of results is mostly done using an HMM-ANN hybrid speech recognizer.

In recent years, people around Hynek Hermansky have shown [13, 27, 11, 6] that non-linear mapping using ANN can be used in conventional HMM-GMM recognizers. The net simply produces a stream of probabilities, which is, after post-processing, used as input to HMMs. This opens the possibility to use such non-traditional features with "standard architecture" speech recognizer, as HTK (in the Aurora task) or Sphinx (in the SPINE project).

4.1 TRAP architecture

Briefly described, a TRAP system (Fig. 4.1) classifies long (for example 1 second) temporal trajectories of spectral features into classes using neural networks (NN). Then, band-outputs are merged by another NN to form the final probability vector. If used with an HMM recognition system, those probabilities are post-processed to better fit HMM requirements (feature independence and Gaussian distribution).

In more detail, the TRAP system consists of the following: (1) *input features* - log of energies in critical bands. We used 15 Bark-scale critical bands from 0 to 4000 Hz, the log energies were computed by the 'rasta' executable from ICSI. We need to have phonetic labels for the input data. (2) *generation of TRAPS*. (3) *TRAP- or band-classifiers* perform classification of TRAPs into phonetic classes or broad phonetic categories. (4) *post-processing* of band-classifier outputs. This involves conversion of linear probabilities to logs. (5) *merging net* putting all the band-classifier outputs together. (6) *post-processing* of the merger output (again phoneme probabilities) to form features suitable for an HMM recognizer. (7) *HMM recognizer*. We worked with two HMM recognizers (Digits built using HTK and Sphinx on SPINE).

The input: Log energies are in all experiments computed on Bark-scale using the `rasta` executable from ICSI: (1) the signal is divided into frames of 25 ms with widow shift of 10 ms. For 8000 Hz sampling frequency, this means 200-sample frames with shift of 80 samples. (2) power

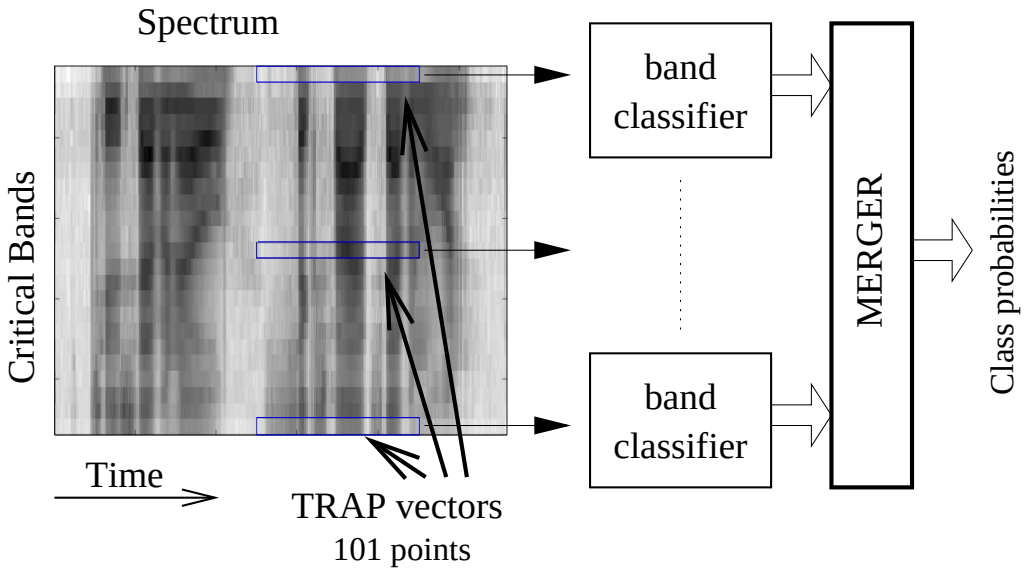

Figure 4.1: TRAP system.

FFT spectrum is taken. (3) filter energies are computed using a 15-band Bark-scaled filterbank. (4) log is taken.

We should note, that the TRAPs used need fairly long context (50 past and 50 future frames for 101-point TRAPs), so that a special care must be given to the beginning and end of each file.

In experiments, we used pre-generated phonetic labels: (1) converted from Stories and Numbers label-files by Sharma for the Stories-Numbers-Digits (SND) experiments. (2) converted and re-mapped from TIMIT and Stories for the reference setup by Lukáš Burget. (3) generated by Sunil Sivadas for SPINE.

Generation of TRAPS A TRAP is nothing but a piece of temporal trajectory of a given band energy of certain length. Most of experiments were conducted with 101-point (1 second) TRAPs. The label of the TRAP is the original label of its central frame. In addition to a mechanical re-arranging of 101-point trajectories into an output matrix, the following options were tested:
(1) *mean and variance normalization:* mostly done independently for each TRAP. Sentence-based mean and variance normalization were tested too (and proved to give similar results as the TRAP-based in SND experiment). Advantage of the sentence-based normalization is that we can tell the NN training software (Quicknet) to select TRAPs on-line (just by specifying left and right context) rather than to create huge input files. For multi-band TRAPs, the normalization is always independent band-by-band.
(2) *Hamming windowing* of TRAPs was done rather for historical reasons (when Sharma did experiments with distance-based classification of TRAPs, the Hamming windowing helped to pre-accentuate the center of TRAP in the distance computation). As the NN training software does a global mean and variance normalization of each feature prior to NN training, the effect of Hamming windowing is canceled.
(3) *Discarding some labels*. It was found advantageous to discard TRAPs carrying some data from the training. In the SND experiments, all phonemes not appearing in Numbers had to be discarded. In reference experiments, frames carrying the 'other' label were discarded.
(4) *Balancing the data*. The amounts of frames per class in the training set are mostly heavily unbal-

anced (most of silence frames, followed by long vowels, little data for phonemes like 'th', etc.). The data can be balanced prior to NN training by specifying down-sampling factors per class.

Band classifiers Band classifiers (also called TRAP classifiers, small nets, first step, band-posterior estimators) classify the TRAPs into phonetic classes, or, in some experiments, into broad phonetic categories. Each band classifier is a standard multi-layer perceptron (MLP) with 3 layers: (1) the input layer's size is determined by the length of TRAP (mostly 101 points). (2) the hidden layer, with sigmoid non-linearities, having 300 neurons in most experiments. (3) the output layer whose size is given by the number of classes. The softmax [3] non-linearity was used in final layer in band-classifiers.

The training data is split into a training and cross-validation (CV) sets. The learning rate of the net is determined upon the accuracy on the CV set after each epoch by the "new-Bob" (see the documentation to QuickNet training executable `qnstrn`) algorithm.

Post-processing of band-classifiers output Before being introduced to the merger, the following processing is done on class posteriors: (1) log is taken to gaussianize the posteriors. An experiment was conducted also with letting the softmax output intact, but it gave slightly worse performance. (2) multiplication by priors (some experiments) done physically as the addition of priors in the log-domain. This is again a historical step, which does not have sense while training the merger: before training, the data are globally mean and variance normalized so that any prior effect is canceled.

Merger The merger is generally trained on different data from those used for training band-classifiers. It implies that for this training data, TRAPs must be generated and forward-passed through the band classifiers. Resulting posteriors (after post-processing described above) are then used to train the merger. The classifier is an MLP with 3 layers: (1) the input layer's size is determined by the product of number of bands times the number of classes per band. For example, for 15 bands and 42 phonemes, the input layer size is 630. (2) the hidden layer, with sigmoid non-linearities. We used mostly 300 neurons in the hidden layer, which seems quite few compared to the input layer size. Unfortunately, more neurons in the hidden layer result in very long training files. (3) the output layer whose size is given by the number of classes. Softmax was used in final layer for training. In the forward-pass, the softmax was kept and followed by an additional off-net non-linearity (log or atanh), or it was removed.

The training of the merger was also driven by the "new-Bob" algorithm for determination of the learning rate.

Merger output post-processing We want to convert the output of the merger to feature files suitable for HMM recognizer. Two steps are necessary:

(1) *Gaussianization*: the outputs of softmax are not Gaussian at all, they have bi-modal distribution with sharp peaks closed to 0 and 1 for most represented classes (as silence) and peaky uni-modal distribution (peak closed to 0) for the other classes. The Gaussianization can be done (a) by taking *log of the softmax* output (this is going to expand the probabilities closed to 0 to an approximately Gaussian shape, but the problem of probabilities closed to 1 persists: they are going to create a sharp edge in the resulting distribution, or even a peak), (b) by taking *hyperbolic arcus-tangens of softmax output*: $\text{atanh}(2x + 1)$, where x is the softmax output (produces more Gaussian-like distribution), (c) by *removing the softmax* from the output layer of the net (which is the simplest solution – no

Gaussianization necessary – and gave the best results), (d) by an *explicit Gaussianization* [22] (not used in the TRAP framework).

(2) *De-correlation*. HMM's with diagonal covariance matrices "like the features de-correlated". Therefore a PCA is computed on the training data, and then applied to the entire data. Experiments were done on the PCA using raw or normalized covariance matrix.

In addition to those two steps, we can test some processing known from "standard" features, as delta and acceleration coefficient computation, mean and variance normalization, etc.

4.2 Visuzation and performance testing of TRAPs

It is difficult to visualize weights and biases of a trained net. **Mean TRAPs** were generated to see, if they are consistent with Sharma's results and also if they are consistent among experiments. In addition, the analysis software 'trapalyzer' can produce variance (or better standard-deviation) TRAPs, that tell us, how much variability we can expect at which place of the time trajectory.

Phoneme recognition accuracies are the quickest way to learn, if nets are classifying TRAPs well or bad. Cross-validation set accuracy is the figure to look at both in band-classifier and merger training. Also, phoneme recognition accuracy per class is helpful. Quicknet software can not output this per-class accuracy, but it can be obtained using the in-house 'ffrapalyzer' software.

Phoneme confusion matrices are the way to see how precisely the net is able to classify, and where it makes most of the errors. Consider number of classes L, and a data set with N frames. We have correct labels for this set, so that we know, that class i has N_i frames. The priors of classes are therefore given: $P_l = \frac{N_l}{N}$. For each frame, we have a vector of net outputs giving class posteriors: $\mathbf{x} = [x_1, x_2 \dots x_L]$.

Hard confusion matrix for each posterior vector, the highest posterior determines the classification of the frame. We can compute how many times a phoneme of correct class i was classified as class j (in ideal case, i would be always equal to j): counts C_{ij}. The hard confusion matrix is then given by a simple division by prior counts: $H_{ij} = \frac{C_{ij}}{N_i}$. Ideally, this matrix would be unity (everything correctly classified).

Soft confusion matrix Rather than taking a decision, this matrix takes into account all the posteriors, and sums them up for each class. Each row of the soft confusion matrix is then defined: $\mathbf{S}_{i,:} = \frac{\sum_{\forall i} \mathbf{x}}{N_i}$, where $\sum_{\forall i} \mathbf{x}$ means the sum of all posterior vectors for frames with correct label i. This matrix is therefore going to give more 'blurred' picture of how the posteriors look like for different classes. Ideally, this matrix would be again unity (net each time sure, that it is the correct class and no else).

Variance matrix of posteriors per class The motivation to compute this matrix was: "if a variance of the posterior for a given correct class is low, it does not matter, if this posterior is high where it should not be – the net works consistently and the merger will take care of it". It is defined by: $\mathbf{V}_{i,:} = E\{(\mathbf{x}_{\forall i} - \mu_i)^2\}$, where E denotes the expectation, $\mathbf{x}_{\forall i}$ all posterior vectors for correct class i and μ_i the mean posterior vector for this correct class. Unfortunately, we found this matrix not very representative, as the variances of posteriors depend on their values (higher variances for

posteriors $\gg 0$ and very low for posteriors closed to 0). The resulting matrix is therefore very similar to the soft confusion one.

Output covariance matrix Here, we do not use any knowledge of the correct classes, we just compute the correlation of posteriors at the output of the net. The covariance matrix is given by definition: $\mathbf{C} = E\{(\mathbf{x}^T\mathbf{x} - \mu^T\mu)\}$, where μ is the global posterior mean. For the visualization and class clustering, we have computed normalized covariance-matrix, with elements: $\rho_{ij} = \frac{c_{ij}}{\sqrt{c_{ii}c_{jj}}}$. The ideal form of this matrix is again unity (no outputs correlated with each other).

Note on visualization of matrices Except for the normalized covariance matrix, the visualization suffers from silence class being recognized more precisely than the other classes. The other elements then do not have sufficient resolution. It is then a good idea to visualize the matrices without the row and column corresponding to the silence.

Word error rate of HMM recognizer The ultimate number while using TRAPs is the word-error rate (WER) of the HMM recognizer using merger-posteriors as features (after some post-processing). This number should be compared to the WER obtained using "classical" features, as MFCC's.

4.3 Basic experiments: Stories–Numbers–Digits

Those experiments were conducted exactly on the same data Jain and Sharma used in their work, the results are therefore directly comparable. Due to limited coverage of this shortened version of habilitation thesis, it was not possible to include all experimental results (also for the following section). Interested reader is refered to the complete version [30].

First experiments were conducted in order to verify and reproduce Jain's results. Finally, her experiment was successfully reproduced including porting of the code to standard Linux environment (no use os specialized SPERT boards). Furthermore, a software for work with TRAPs – `trapper` – was written. The example of mean TRAPs generated on Stories (5th band) is shown in Fig. 4.2. They correspond to Sharma's results in [28].

Following experiments concentrated on the effect of sentence-based mean and variance normalization. We have shown that sentence-based normalization gives better results than TRAP-based one. As for "classical" methods, this is probably due to more reliable estimation of mean and variance on the length of a sentence rather than on 101-point TRAP. Unfortunately, this also brings sentence-latency and is not suitable for latency-critical tasks.

Broad phonetic categories in bands were also investigated. We have observed that training and recognition performance in bands increased from phonemes to broad categories. Unfortunately, given the probabilities only for 4 categories per band, the merger is not able to recognize phonemes reliably and this is translated into a big hit on the overall recognition performance.

A set of experiments was also conducted with "balanced training" where numbers of TRAPs per class were artificially equalized. We have can concluded, that the band-nets should be trained with all the available data.

Stories-Numbers-Digits: Conclusions We have reproduced the baseline experiment with satisfactory results and performed some other experiments with broad phonetic classes and balancing of the training data. The results were represented in terms of band-classifier cross-validation accuracy,

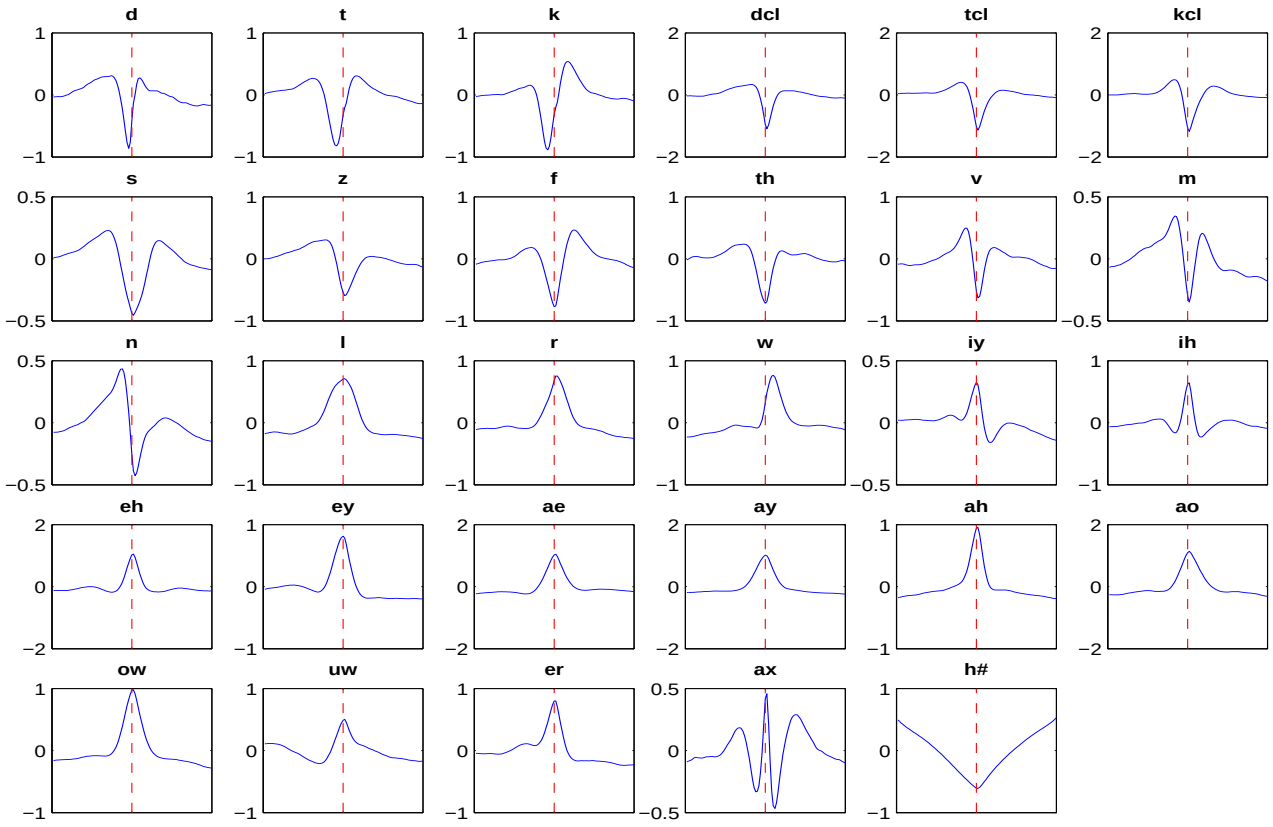

Figure 4.2: Mean TRAPs on Stories

phoneme recognition accuracy on Numbers, CV accuracy when training the merger and finally of word recognition accuracy of the HMM recognizer.

There are however the following problems with the SND experiments: (1) the phoneme set for band-classifier training is not full and contains only the 29 phonemes present in Numbers. There are lots of 'bad labels' we have to discard. (2) Although Stories provide good phonetic coverage, Numbers contain a very limited vocabulary, so that the phonemes appear all the time in the same context. This may heavily bias the phoneme recognition accuracies. (3) Global numbers are biased by the distribution of data among classes. We need more detailed analysis of what is happening in bands and in the merger.

4.4 Reference experiments: Timit and Stories

The reasons for switching to this experimental setup from SND were: (1) to have a coherent phoneme set for both band-classifier and merger training. (2) to dispose of phonemes in various context for both band-classifier and merger training. Also, some visualization tools were produced for this setup allowing to see the confusion matrices and to do detailed per-class analysis. Those experiments are called 'Reference-TRAPs'.

These experiments were run on parts #2 and #3 of 4 data-sets defined by Lukáš Burget:

part	purpose	sub-division	source	amount	comment
1	eventually for LDA training	train	Stories	165 min	-
2	Band-classif. training	train	TIMIT	106 min	-
	(tnn-timit)	CV	TIMIT	20 min	-
3	Merger training	train	Stories	145 min	same data as for part 1
	(mnn-stories)	CV	Stories	20 min	-
4	phoneme HMM recognizer	train	TIMIT	84 min	-
		test	TIMIT	49 min	different data from part 2

No HMM recognizer was trained at the top of Reference TRAPs. The evaluation of results was based on what we have seen during the training and cross-validation of nets: (1) final phoneme recognition accuracy on the cross-validation set of tnn-timit while training the band-classifiers for 3 bands (0-th, 5-th and 10-th). (2) phoneme recognition accuracy while forward-passing Stories through the band-classifiers for 3 bands (0-th, 5-th and 10-th). (3) final phoneme recognition accuracy on the cross-validation set of Stories in merger training.

Baseline experiments were performed with the same TRAP generation and net configuration as for SND, to assess the phoneme recognition accuracy in bands and at the output of the merger, and to do class-based analysis. The CV accuracies sound in bands were lower than in the SND setup, which is understandable, as we have much less silence in TNN-Timit than in the original Stories (14% here versus 27% before). What is more shocking is the accuracy after training the merger: from 80% for the SND experiment, we go down to mere 50%. A smaller proportion of silence in MNN-Stories (19% versus 25% before in Numbers) can be blamed, but is not solely responsible for 30% hit. The *variability of contexts* is probably the factor responsible for this huge difference.

On this baseline experiment, new *visualization and analysis tools* were tested. To asses the performance of TRAPs in bands, hard and soft confusion matrices and output normalized covariance matrices were computed for each band, based on the MNN-Stories data forward-passed through the band-classifiers. For band #5, the soft confusion matrix can be seen in Figure 4.3. We can see, that the phonemes form clusters similar to broad phonetic categories. It is impossible to include all the figures for all the bands in this report, but it is interesting to see, how certain phonemes (especially liquids) “travel” among classes from band to band.

The y-axis of figures is completed by two important numbers: the first is the percentage of occurrences of the given phoneme in MNN-Stories while the second is the recognition accuracy (’hit-rate’) of this phoneme. Not surprisingly, we see, that the silence is hit in most cases (82%).

The following experiments were conducted again with *automatically generated broad phonetic classes*. The band accuracies are not comparable with the previous setup, as we have lower number of broader classes. Obviously, the accuracy is higher. At the output of the merger, we have 4% hit. As it was mentioned erlier, by limiting the number of classes per band, we have also limited the number of merger’s parameters. A fair comparison would require increasing the size of the hidden layer. Also, a simpler experiment with uniform classes for all bands should be conducted.

The *effect of frequency-context* was studied in experiments with 2-band TRAPs. Theoretically, this approach is supported by the studies of co-modulation masking. To ensure comparable number of net parameters with the previous experiments, we have made the TRAPs shorter- just 51-frames instead of original 101. This brings the size of a 2-band TRAP to 102, which is almost the same as 101. In this experiment, the 2 bands were adjacent, e.g. 0-1, 1-2, ... 13-14. The total number of couples was 14. We see a nice improvement from isolated bands to couples of bands. At the output of the merger, the improvement is however not spectacular: just 0.7%.

Another flavor of the previous experiment is the use of 2 bands with 1-band skip: we should

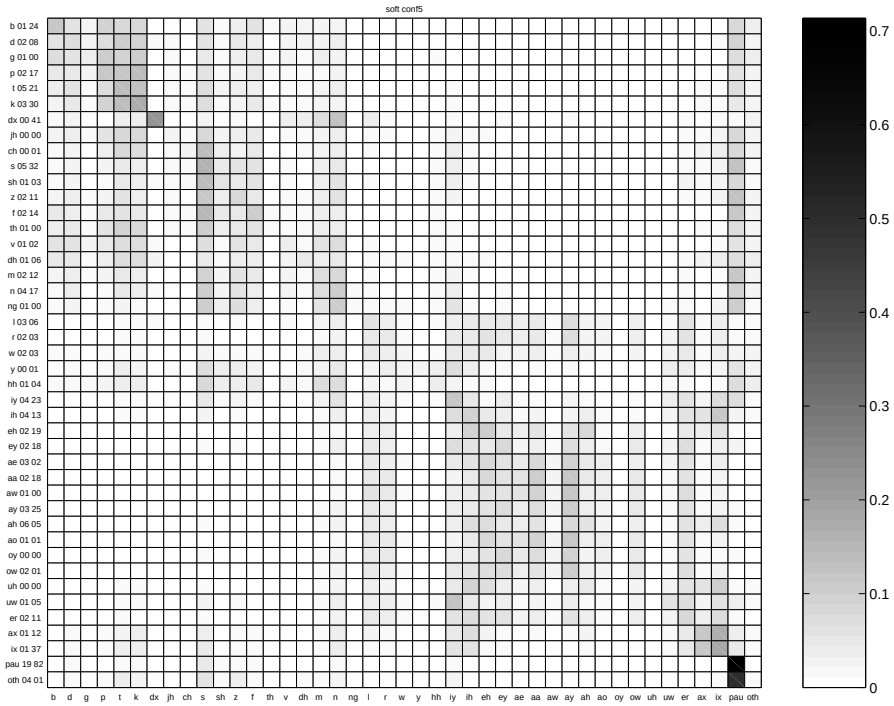

Figure 4.3: Soft confusion matrix of band #5 (reference TRAPs–baseline experiment).

remember that the frequency characteristics for adjacent bands overlap, so that the 2 adjacent band outputs are necessarily correlated. Therefore, we conducted an experiment with couples of bands with the skipping of one band. The couples are: 0-2, 1-3, ...12-14, and their total number is 13. We have seen again an improvement in bands and again a slight improvement at the output of the merger.

The last reference experiment was performed with a bit of "brute force" on 2-band TRAPs: we have increased the hidden layer size for band-classifiers was increased to 500 neurons, that for the merger to 1000 neurons. As result, we have neat improvement in bands, and what is the best, a 4% improvement at the output of the merger. As it was mentioned at the beginning of this section, the question "Which of the changes is responsible for most of the improvement" is open.

Reference TRAPs – Conclusions TRAP experiments were conducted on a set of databases providing, in our opinion, more reliable and realistic assessment of their performance than Stories–Numbers–Digits. It was found, that on a database with phonemes occurring in varying context, the phoneme recognition accuracy at the output of the merger is rather around 50% than 80% (SND experiments).

Experiments with *automatically generated broad phonetic classes* showed that the overall phoneme recognition accuracy is inferior to baseline results. The following issues are open: (1) as mentioned, limitation of number of classes in bands implies the limitation of merger parameters. For a fair comparison, the size of hidden layer should be increased to match the number of parameters of the baseline. (2) there are many ways to generate the classes: manual creation, and automated creation uniform for all bands should be tested. (3) we need not necessarily have the same number of classes per band. Instead of a "hard" number, a variable number based on a distance measure (or mutual information) could be used. (4) finally, the merger could be trained not to recognize phonemes but also broad phonetic classes. Those probabilities (after post-processing) could serve as

input to an HMM recognizer [16].

2-band TRAPs have shown a huge potential in performance. While testing 51-point TRAPs, we should remember that the normalization of input features was TRAP-based - the estimation of mean and variance was therefore on $2\times$ less data than for the baseline. This calls for a normalization using a different window, or whole sentence.

In SND experiments, *sentence-based normalization* provided good results (the mean and variance are more reliably estimated). This approach was not tested with the reference setup, those experiments should be completed.

4.5 TRAPs on SPINE

SPINE (Speech in Noisy Environments) is an evaluation run by the Naval Research Laboratory. The task a medium-sized vocabulary recognition on several military environments. The training and evaluation data from 2000 were used to assess performances of our features. These data come as stereo-recordings, but we disposed of data pre-segmented into speech and silence regions at CMU [29]. The recognizer – SPHINX – came also from CMU [24].

Several experiments were run with the TRAP-based feature extraction using neural nets trained on TIMIT and STORIES (previous section) or with TRAPs derived directly from SPINE data. Significant efforts were devoted to te study of post-processing of merger output (de-correlation and Gaussianization using various methods). Full description is again available in [30].

TRAPs on SPINE: Conclusions In the last experiments on SPINE, only TRAP-based mean and variance normalization were used. In the SND experiments, we have seen some improvement when going from TRAP based to sentence-based normalization. This should be tested on SPINE. We can go even beyond: we know, that one speaker is always at one side of a conversation, so that the normalization could be conversation and channel based. That would provide us with more reliable estimates of means and variances.

The training of band-classifiers on SPINE was abandoned, as they did not perform so good as those trained on other database. This approach was unfortunately no more tested in the "improved" data (silence regions deleted), which should make the band-classifier training more consistent. Recent results of Jain show, that especially for 2-band TRAPs, this should be a very promising way.

4.6 TRAP summary

Although being comparable MFCC's on the small vocabulary task with context-independent phonemes, TRAPs seem to have hard time to reach the performance of "classical" features on LVCSR task using context-dependent tied models. The SPINE experiments however show promising approaching of TRAP performance to the MFCC (41.2 versus 36.7% WER).

The availability and quality of *labels* seems to play crucial role in the TRAP work. We have seen a dramatic improvement from 53.7 to 47.9% just by re-mapping the ICSI 56 phoneme set to a smaller one containing 34 phonemes. We are however still far from optimum, as we tune the TRAPs to context-independent (CI) phonemes (often not the same, as the recognizer is using) while the LVCSR systems use CI-models just for the initialization. It is however difficult to train any net aiming at the discrimination of classes finer than CI-phonemes due to their big numbers (e.g. 2600 tied states in SPHINX).

We have done experiments with processing of the *output* of the merger, but similar care should be taken while *processing the band-classifier outputs* for the *input* to the merger. Currently, a log of softmax output is taken. We have tested that log performs better than taking just the raw output. It would be worth to investigate if a hyperbolic arcus-tangens or removing the softmax do not bring similar improvement as at the output of the merger.

We use the neural nets as a black box, without changing their architecture (which is determined by Quicknet), number of hidden layers (Quicknet supports just 1), learning strategies, etc. There is certainly a potential of improvement here.

PCA applied at the output of all the processing is the simplest and probably also the worst way to de-correlate the features for HMM recognizer. LDA and newly MLLT (Maximum-likelihood linear transforms) are certainly of interest. As it was already mentioned, the mean normalization could be done on conversation and channel basis for SPINE, where this information is available. See [30] for the discussion of current and future work on TRAPs at OGI Portland, VUT Brno and ICSI Berkeley.

Chapter 5

Conclusions

As each of the experimental parts ended by a conclusive section, those global conclusions will be quite brief.

We have shown, that the feature extraction is of great importance in nowadays speech recognition systems, and that quite a simple operation, as filtering of temporal trajectories using 101-tap filters, can bring important increase in recognition performance (2% absolute improvement over the baseline of 90% indeed *is* a very important increase). But even more interesting than this is the verification of the basic idea: it is possible to train filters on labeled speech data, those filters resemble to the ones obtained previously by studies of human auditory periphery, and they even improve recognition accuracy! Of course, the experiments conducted are far from complete[1] and it should be for example appended by a thorough study of LDA-filter behavior in different types of noise.

On the other hand, TRAP experiments have not yet shown brilliant improvement over MFCC's in this work, and for some tasks they are well behind. An attempt to excuse ourselves would be stating that researchers needed more than 20 years to come up with today's form of MFCC's with their Δs and $\Delta\Delta$s, and that TRAP efforts started only two-three years ago. Also, there is quite a number of questions in TRAP system design (summarized in 4.6) ranging from label selection to neural net architecture. We believe however that we are on a very promising way and that temporal trajectories and non-linear classifiers will have a significant role in future speech feature-extraction algorithms. Last experiments of Grézl [9] show that TRAPs, derived from 3 frequency bands or obtained by spectro-temporal operators, outperform MFCC's for some tasks.

To conclude this work, I would like to express my great will to continue in the speech processing research and teaching, a field that I consider very interesting, challenging from the scientific point of view but also bringing quite a lot of fun. I hope to be a good tutor to my pre- and post-grad students. I hope to take part in international projects, bringing not only a huge amount of know-how and lots of work, but also visits of nice places and meeting smart and funny people. I sincerely believe that Czech republic will get new possibilities by joining the EC — by the work of myself, and of my group, I would like to make a contribution to the success of our country in the common Europe.

[1] But is a set of experiments ever complete?

Bibliography

[1] A. Adami, L. Burget, S. Dupont, H. Garudadri, F. Grezl, H. Hermansky, P. Jain, S. Kajarekar, N. Morgan, and S. Sivadas. Qualcomm-ICSI-OGI features for ASR. In *Proc. ICSLP 2002*, Denver, Colorado, USA, 2002.

[2] H. Bourlard. Nouveaux paradigmes pour la reconnaissance robuste de la parole. In *Proc. XXI-èmes Journées d'Étude sur la Parole*, pages 263–272, Avignon, France, June 1996.

[3] H. Bourlard and N. Morgan. *Connectionist Speech Recognition: A Hybrid Approach*. Number 247 in Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, 1994.

[4] L. Burget. Concepts of the dissertation. Technical report, Brno University of Technology, Inst. of Radioelectronics, April 2001.

[5] S. B. Davis and P. Mermelstein. Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. on Acoustics, Speech & Signal Processing*, 28(4):357–366, 1980.

[6] D. P. W. Ellis and M. J. Reyes Gomez. Investigations into tandem acoustic modeling for the Aurora task. In *Proc. Eurospeech 2001*, Aalborg, Denmark, September 2001.

[7] B. Gold and N. Morgan. *Speech and audio signal processing*. John Wiley & Sons, 2000.

[8] F. Grézl. Concepts of the dissertation. Technical report, VUT Brno, Faculty of Information Technology, April 2002.

[9] F. Grézl and H. Hermansky. Local averaging and differentiating of spectral plane for trap-based asr. In *submitted to Eurospeech 2003*, Geneva, 2003.

[10] H. Hermansky. Human speech perception: Some lessons from automatic speech recognition. In V. Matoušek, P. Mautner, P. Mouček, and K. Taušer, editors, *Proc. of 4th International Conference Text, Speech,Dialogue - TSD 2001*, number 2166 in Lecture notes in artificial intelligence, pages 187–196, Železná Ruda, Czech Republic, September 2001. Springer Verlag.

[11] H. Hermansky, D. P. W. Ellis, and S. Sharma. Tandem connectionist feature extraction for conventional HMM systems. In *Proc. ICASSP 2000*, Turkey, 2000.

[12] H. Hermansky and N. Morgan. RASTA processing of speech. *Trans. on Speech & Audio Processing*, 2(4):578–589, 1994.

[13] H. Hermansky, S. Sharma, and P. Jain. Data-derived nonlinear mapping for feature extraction in HMM. In *Proc. Workshop on automatic speech recognition and understanding*, Keystone, December 1999.

[14] M. J. Hunt. A statistical approach to metrics for word and syllable recognition. *J. Acoust Soc. Am.*, 66(S1)(S35(A)), 1979.

[15] P. Jain and H. Hermansky. Improved mean and variance normalization for robust speech recognition. In *Proc. ICASSP 2001*, Salt Lake City, Utah, USA, May 2001.

[16] P. Jain, H. Hermansky, and B. Kingsbury. Distributed speech recognition using noise-robust MFCC and TRAPS-estimated manner features. In *Proc. ICSLP 2002*, Denver, Colorado, USA, 2002.

[17] J. Jan. *Digital filtering, analysis and restoration of signals (in Czech)*. Brno University of Technology, 1997.

[18] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1998.

[19] J. Kafka. Linear discriminant analysis in recognition of Czech (in Czech), Diploma thesis. Technical report, Brno University of Technology, Faculty of Electrical Engineering and Communication, June 2002.

[20] S. Kajarekar. *Analysis of Variability in Speech with Applications to Speech and Speaker Recognition*. PhD thesis, OGI School of Science and Engineering, Oregon Health & Science University, Portland, Oregon, July 2002.

[21] M. Karafiát and J. Černocký. Context dependent hidden Markov models in recognition of Czech. In *Proc. Radioelektronika 2002*, pages 37–40, Bratislava, Slovakia, May 2002.

[22] P. Matějka, P. Schwarz, M. Karafiát, and J. Černocký. Some like it Gaussian .... In P. Sojka, I. Kopeček, and K. Pala, editors, *Proc. of the 5th International Conference on Text, Speech and Dialogue—TSD 2002*, Lecture Notes in Artificial Intelligence LNCS/LNAI 2448, pages 321–324, Brno, Czech Republic, Sep 2002. Springer-Verlag.

[23] P. Motlíček and J. Černocký. All-pole modeling based feature extraction for AURORA3 DSR task. In *submitted to ICASSP 2003*, Hongkong, 2003.

[24] P. Placeway, S. Chen, Maxine Eskenazi, U. Jain, V. Parikh, B. Raj, M. Ravishankar, R. Rosenfeld, K. Seymore, M. Siegler, R. Stern, and E. Thayer. The 1996 HUB-4 SPHINX-3 system. In *Proc. 1997 ARPA Speech Recognition Workshop*, 1997.

[25] L. Rabiner and B. H. Juang. *Fundamentals of speech recognition*. Signal Processing. Prentice Hall, Engelwood Cliffs, NJ, 1993.

[26] P. Schwarz. Continuous speech recognition: spelled letters (in Czech), diploma thesis. Technical report, Brno University of Technology, Faculty of Electrical Engineering and Computer Science, Brno, 2001.

[27] S. Sharma, D. Ellis, S. Kajarekar, P. Jain, and H. Hermansky. Feature extraction using nonlinear transformation for robust speech recognition on the Aurora database. In *Proc. ICASSP 2000*, Turkey, 2000.

[28] S. R. Sharma. *Multi-stream approach to robust speech recognition*. PhD thesis, Oregon Graduate Institute of Science and Technology, October 1999.

[29] R. Singh, M. L. Seltzer, B. Raj, and R. M. Stern. Speech in noisy environments: robust automatic segmentation, feature extraction, and hypothesis combination. In *Proc. ICASSP 2001*, Salt Lake City, Utah, USA, May 2001.

[30] J. Černocký. Temporal proceesing for feature extraction in speech recogniti on. Habilitation thesis, Brno University of Technology, Faculty of INformation Technology, October 2002. http://www.fit.vutbr.cz/ cernocky/publi/2002/habil.pdf.

[31] S. Young, J. Jansen, J. Odell, D. Ollason, and P. Woodland. *The HTK book*. Entropics Cambridge Research Lab., Cambridge, UK, 1996.

Abstract

Speech recognition is a booming research field, having large number of applications in telecommunications (especially mobile), automobile industry, consumer electronics, military and security, etc. Speech recognition systems are classically built from three basic blocks: feature extraction, acoustic matching and language modeling. While the last two are trained on data (annotated databases for acoustics and large speech corpora for the LM), feature extraction block is often neglected and most often, mel-frequency cepstral coefficients (MFCC) are used. This work concentrates on two techniques that should improve the feature extraction.

The first one is temporal filtering of feature trajectories using filters designed on data using Linear Discriminant Analysis (LDA). This technique is shown to improve the recognition accuracy of isolated Czech words, confirming previous results on US-English obtained by our colleagues from OGI Portland.

The second part of the work concentrates on more revolutionary approach of feature extraction using TRAPs (temporal patterns) whose fundamentals were also laid at OGI. Several experiments were conducted on three databases during author's stay at OGI. Although we have shown that TRAPs are comparable to MFCC's only on a small vocabulary recognition task, we believe that combination of frequency-band processing and neural nets will become very important in the next decade, and that they will become standard blocks of feature extraction.

Abstrakt

Rozpoznávání řeči je rychle se rozvíjejícím oborem s množstvím aplikací v telekomunikacích (zvláště mobilních), automobilovém průmyslu, spotřební elektronice, vojenské a bezpečnostní oblasti, atd. Rozpoznávače řeči se klasicky skládají ze tří základních bloků: výpočtu příznaků (parametrizace), akustického srovnávání a jazykového modelu. Zatímco poslední dva bloky jsou trénovány na datech (akustika na anotovaných řečových databázích, LM na korpusech textových dat), parametrizace je často zanedbávána a na vstupech rozpoznávačů najdeme nejčastěji mel-frekvenční cepstrální koeficienty (MFCC). Tato práce se zaměřuje na dvě techniky, které by měly parametrizaci zkvalitnit.

První z nich je časová filtrace trajektorií parametrů pomocí LDA-filtrů. Tyto jsou získány z řečových dat pomocí Lineární diskriminační analýzy (LDA). V práci ukážeme, že tato technika zlepšuje úspěšnost rozpoznávače při rozpoznávání izolovaných českých slov. Potvrdili jsme tak předchozí výsledky na americké angličtině, získané naší partnerskou skupinou na OGI Portland.

Druhá část práce se zaměřuje na "revolučnější" přístup k parametrizaci pomocí časových trajektorií (TRAPs). Základ této metody byl rovněž položen skupinou na OGI a experimenty popsané v této práci byly provedeny během autorova sedmiměsíčního pobytu v Portlandu. I když jsme prokázali, že TRAP-příznaky jsou srovnatelné s MFCC pouze na rozpoznávání omezeného souboru slov, věříme, že kombinace zpracování v jednotlivých kmitočtových pásmech s neuronovými sítěmi nabude v následující dekádě na důležitosti a že se tyto techniky stanou standardními bloky v parametrizaci řeči.