Proceedings of the 4th International Conference on

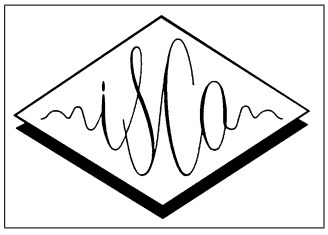
NOn-LInear Speech Processing

NOLISP 2007

Mai 22-25, 2007. Paris, France

Sponsored by:

- **ISCA** International Speech Communication Association
- **EURASIP** European Association for Signal Processing
- **IEEE** Institute of Electrical and Electronics Engineers
- **HINDAWI** Publishing corporation



Committees

Scientific Committee

Frédéric BIMBOT	IRISA, Rennes (France)
Mohamed CHETOUANI	UPMC, Paris (France)
Gérard CHOLLET	ENST, Paris (France)
Tariq DURRANI	University of Strathclyde, Glasgow (UK)
Marcos FAÚNDEZ-ZANUY	EUPMt, Barcelona (Spain)
Bruno GAS	UPMC, Paris (France)
Hynek HERMANSKY	OGI, Portland (USA)
Amir HUSSAIN	University of Stirling, Scotland (UK)
Eric KELLER	University of Lausanne (Switzerland)
Bastiaan KLEIJN	KTH, Stockholm (Sweden)
Gernot KUBIN	TUG, Graz (Austria)
Petros MARAGOS	Nat. Tech. Univ. of Athens (Greece)
Stephen Mc LAUGHLIN	University of Edimburgh (UK)
Kuldip PALIWAL	University of Brisbane (Australia)
Bojan PETEK	University of Ljubljana (Slovenia)
Jean ROUAT	University of Sherbrooke (Canada)
Jean SCHOENTGEN	Univ. Libre Bruxelles (Belgium)
Isabel TRANCOSO	INESC (Portugal)

Organizing Committee

Mohamed CHETOUANI	UPMC, Paris (FRANCE)
Bruno GAS	UPMC, Paris (FRANCE)
Amir HUSSAIN	University of Stirling, Scotland (UK)
Maurice MILGRAM	UPMC, Paris (FRANCE)
Jean-Luc ZARADER	UPMC, Paris (FRANCE)

Foreword

After the success of NOLISP'03, NOLISP'04 summer school and NOLISP'05, we are pleased to present NOLISP'07. The fourth event in a series of events related to Non-linear speech processing.

Many specifics of the speech signal are not well addressed by conventional models currently used in the field of speech processing. The purpose of NOLISP is to present and discuss novel ideas, work and results related to alternative techniques for speech processing, which depart from mainstream approach.

With this intention in mind, we provide an open forum for discussion. ALternate approaches are appreciated, although the results achieved at present may not clearly surpass results based on stat-of-the-art methods.

Please, try to establish as many contacts with your colleagues and discussions as possible: these small-size workshops make it feasible. Please, enjoy the workshop; enjoy the city, and the french meals and drinks.

We want to acknowledge our colleagues for their help and the reviewers for their very exhaustive reviews.

Best wishes for NOLISP

Mohamed Chetouani
Bruno Gas
Jean-Luc zarader

Contents

HMM-based Spanish speech synthesis using CBR as F0 estimator

Xavi Gonzalvo, Ignasi Iriondo, Joan Claudi Socoró, Francesc Alías, Carlos Monzo

Department of Communications and Signal Theory
Enginyeria i Arquitectura La Salle, Ramon Llull University, Barcelona, Spain
{gonzalvo,iriondo,jclaudi,falias,cmonzo}@salle.url.edu

Abstract

Hidden Markov Models based text-to-speech (HMM-TTS) synthesis is a technique for generating speech from trained statistical models where spectrum, pitch and durations of basic speech units are modelled altogether. The aim of this work is to describe a Spanish HMM-TTS system using CBR as a F0 estimator, analysing its performance objectively and subjectively. The experiments have been conducted on a reliable labelled speech corpus, whose units have been clustered using contextual factors according to the Spanish language. The results show that the CBR-based F0 estimation is capable of improving the HMM-based baseline performance when synthesizing non-declarative short sentences and reduced contextual information is available.

1. Introduction

One of the main interest in TTS synthesis is to improve quality and naturalness in applications for general purposes. Concatenative speech synthesis for limited domain (e.g. Virtual Weather man [1]) presents drawbacks when trying to use in a different domain. New recordings have the disadvantage of being time consuming and expensive (i.e. labelling, processing different audio levels, texts designs, etc.).

In contrast, the main benefit of HMM-TTS is the capability of modelling voices in order to synthesize different speaker features, styles and emotions. Moreover, voice transformation through concatenative speech synthesis still requires large databases in contrast to HMM which can obtain better results with smaller databases [2]. Some interesting voice transformation approaches using HMM were presented using speaker interpolation [3] or eigenvoices [4]. Furthermore, HMM for speech synthesis could be used in new systems able to unify both approaches and to take advantage of their properties [5].

Language is another important topic when designing a TTS system. HMM-TTS scheme based on contextual factors for clustering can be used for any language (e.g. English [6] or Portuguese [7]). Phonemes (the basic synthesis units) and their context attributes-values pairs (e.g. number of syllables in word, stress and accents, utterance types, etc.) are the main information which changes from one language to another. This work presents contextual factors adapted for Spanish.

The HMM-TTS system presented in this work is based on a source-filter model approach to generate speech directly from HMM itself. It uses a decision tree based on context clustering in order to improve models training and able to characterize phoneme units introducing a counterpart approach with respect to English [6]. As the HMM-TTS system is a complete technique to generate speech, this work presents objective results to measure its performance as a prosody estimator and subjective measures to test the synthesized speech. It is compared with a tested Machine Learning strategy based on case based reasoning (CBR) for prosody estimation [8].

This paper is organized as follows: Section 2 describes HMM system workflow and parameter training and synthesis. Section 3 concerns to CBR for prosody estimation. Section 4 describes decision tree clustering based on contextual factors. Section 5 presents measures, section 6 discusses results and section 7 presents the concluding remarks and future work.

2. HMM-TTS system

2.1. Training system workflow

As in any HMM-TTS system, two stages are distinguished: training and synthesis. Figure 1 depicts the classical training workflow. Each HMM represents a contextual phoneme. First, HMM for isolated phonemes are estimated and each of these models are used as a initialization of the contextual phonemes. Then, similar phonemes are clustered by means of a decision tree using contextual information and designed questions (e.g. Is right an 'a' vowel? Is left context an unvoiced consonant? Is phoneme in the 3rd position of the syllables? etc.). Thanks to this process, if a contextual phoneme does not have a HMM representation (not present in the training data, but in the test), decision tree clusters will generate the unseen model.

Figure 1: *Training workflow*

Each contextual phoneme HMM definition includes spectrum, F0 and state durations. Topology used is a 5 states left-to-right with no-skips. Each state is represented with 2 independent streams, one for spectrum and another for pitch. Both types of information are completed with their delta and delta-delta coefficients.

Spectrum is modelled by 13^{th} order mel-cepstral coefficients which can generate speech with MLSA filter [9]. Spectrum model is a multivariate Gaussian distributions [2].

Spanish corpus has been pitch marked using the approach described in [10]. This algorithm refines mark-up to get a smoothed F0 contour in order to reduce discontinuities in the generated curve for synthesis. The model is a multi-space probability distribution [2] that may be used in order to store continuous logarithmic values of the F0 curve and a discrete indicator for voiced/unvoiced.

State durations of each HMM are modelled by a Multivariate Gaussian distribution [11]. Its dimensionality is equal to the number of states in the corresponding HMM.

2.2. Synthesis process

Figure 2 shows synthesis workflow. Once the system has been trained, it has a set of phonemes represented by contextual factor (each contextual phoneme is a HMM). The first step in the synthesis stage is devoted to produce a complete contextualized list of phonemes from a text to be synthesized. Chosen units are converted into a sequence of HMM.

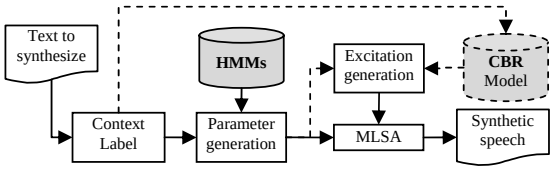


Figure 2: *Synthesis workflow*

Using the algorithm proposed by Fukada in [9], spectrum and F0 parameters are generated from HMM models using dynamic features. Duration is also estimated to maximize the probability of state durations. Excitation signal is generated from the F0 curve and the voiced and unvoiced information. Finally, in order to reconstruct speech, the system uses spectrum parameters as the MLSA filter coefficients and excitation as the filtered signal.

3. CBR system

3.1. CBR and HMM-TTS system description

As shown in figure 2, CBR system for prosody estimator can be included as a module in any TTS system (i.e. excitation signal can be created using either HMM or CBR). In a previous work it is demonstrated that using CBR approach is appropriate to create prosody even with expressive speech [8]. Despite CBR strategy was originally designed for retrieving mean phoneme information related to F0, energy and duration, this work only compares the F0 results with the HMM based F0 estimator.

Figure 3 shows the diagram of this system. It is a corpus oriented method for the quantitative modelling of prosody. Analysis of texts is carried out by SinLib library [12], an engine developed to Spanish text analysis. Characteristics extracted from the text are used to build prosody cases.

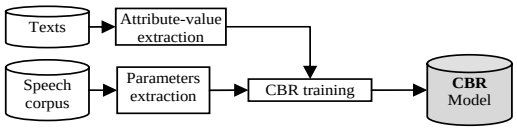


Figure 3: *CBR Training workflow*

Each file of the corpus is analysed in order to convert it into new cases (i.e. a set of attribute-value pairs). The goal is to obtain the solution from the memory of cases that best matches the new problem. When a new text is entered and converted in a set of attribute-value pairs, CBR will look for the best cases so as to retrieve prosody information from the most similar case it has in memory.

3.2. Features

There are various suitable features to characterize each intonational unit. Features extracted will form a set of attribute-value pair that will be used by CBR system to build up a memory of cases. These features (table 1) are based on accentual group (AG) and intonational group (IG) parameters. AG incorporates syllable influence and is related to speech rhythm. Structure at IG level is reached concatenating AGs. This system distinguishes IG for interrogative, declarative and exclamative phrases.

Table 1: *Attribute-value pair for CBR system*

Attributes	
Position of AG in IG	IG Position on phrase
Number of syllables	IG type
Accent type	Position of the stressed syllable

3.3. Training and retrieval

The system training can be seen as a two stages flow: selection and adaptation. In order to optimize the system, case reduction is carried out by grouping similar attributes. Once the case memory is created, the system looks for the most similar stored example. Mean F0 curve per phoneme is retrieved by firstly estimating phoneme durations, normalizing temporal axis and associating each phoneme pitch in basis on the retrieved polynomial.

4. Context based clustering

Each HMM is a phoneme used to synthesize and it is identified by contextual factors. During training stage, similar units are clustered using a decision tree [2]. Information referring to spectrum, F0 and state durations are treated independently because they are affected by different contextual factors.

As the number of contextual factors increases, the number of models will have less training data. To deal with this problem, the clustering scheme will be used to provide the HMMs with enough samples as some states can be shared by similar units.

Text analysis for HMM-TTS based decision tree clustering was carried out by Festival [13] updating an existing Spanish voice. Spanish HMM-TTS required the design of specific questions to use in the tree. Questions design concerns to unit features and contextual factors. Table 2 enumerates the main features taken into account and table 3 shows the main contextual factors. These questions represent a yes/no decision in a node of the tree. Correct questions will determine clusters to reproduce a fine F0 contour in relation to the original intonation.

Table 2: *Spanish phonetic features.*

Unit	Features	
Phoneme	Vowel	Frontal, Back, Half open, Open, Closed
	Consonant	Dental, velar, bilabial, alveolar lateral, Rhotic, palatal, labio-dental, Interdental, Prepalatal, plosive, nasal, fricative
Syllable	Stress, position in word, vowel	
Word	POS, #syllables	
Phrase	End Tone	

5. Experiments

Experiments are conducted on corpus and evaluate objective and subjective measures. On the one hand, objective measures present real F0 estimation results comparing HMM-TTS versus

Table 3: *Spanish phonetic contextual factors.*

Unit	Features
Phoneme	{Preceding, next} Position in syllable
Syllable	{Preceding, next} stress, #phonemes #stressed syllables
Word	Preceding, next POS, #syllables
Phrase	Preceding, next #syllables

CBR technique. On the other hand, subjective results validate Spanish synthesis [1]. Results are presented for various phrase types (interrogative, declarative and exclamative) and lengths (number of phonemes). Phrase classification is referenced to the corpus average length. Thus, a short (S) and a long (L) sentence are below and over the standard deviation while very short (VS) and very long (VL) exceed half the standard deviation over and below.

The Spanish female voice was created from a corpus developed in conjunction with LAICOM [8]. Speech was recorded by a professional speaker in neutral emotion and segmented and revised by speech processing researchers.

The system was trained with HTS [14] using 620 phrases of a total of 833 (25% of the corpus is used for testing purposes). Contextual factors represent around 20000 units to be trained and around 5000 are unseen units.

Firstly, texts were labelled using contextual factors described in table 3. Then, HMMs are trained and clustered. Next, decision trees for spectrum, F0 and state durations are built. These trees are different among them because spectrum, F0 and states duration are affected by different contextual factors (see figure 4). Spectrum states are basically clustered according to phoneme features while F0 questions show the influence of syllables, word and phrase contextual factors. Durations work in a similar manner to F0 as reported in [2]. In order to analyse the effect of the number of nodes in the decision trees, results are presented through two HMM configurations in basis of γ that controls the decision tree length (HMM1, $\gamma(spectrum) = 1, \gamma(f0) = 1, \gamma(duration) = 1$ and HMM2, $\gamma(spectrum) = 0.3, \gamma(f0) = 0.1, \gamma(duration) = 1$). Both systems present the best RMSE over other tested configurations and a tree length below 30% of used units.

5.1. Objective measures

Fundamental frequency estimation is crucial in a source-filter model approach. Objectives measures evaluate F0 RMSE (i.e. estimated vs. real) of the mean F0 for each phoneme (figure 5) and for a full F0 contour (figures 6 and 7).

In order to analyse the effect of phrase length figure 5 shows CBR as the best system to estimate mean F0 per phoneme. As the phrase length increases HMM improves its RMSE. F0 contour RMSE in figure 6 also shows a better HMM RMSE for long sentences than for short. However, CBR gets worse as the sentence is longer, although it presents the best results. Figure 7 demonstrates a good HMM performance for declarative phrases but low for interrogative type. Pearson correlation factor for real and estimated F0 contour is presented in table 4. While CBR presents a continuous correlation value independently of the phrase type and length, HMM presents good results when sentences are long and declarative.

[1] See http://www.salle.url.edu/∼gonzalvo/hmm, for some synthesis examples

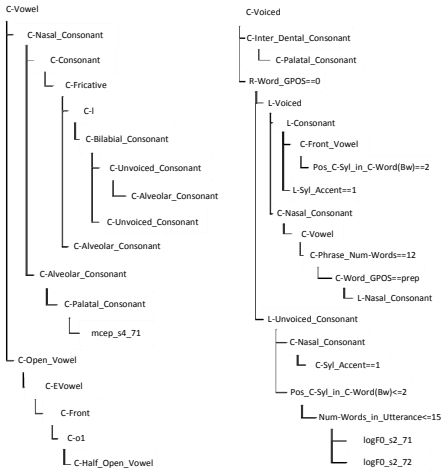


Figure 4: *Decision trees clustering for: 1) spectrum 2) F0*

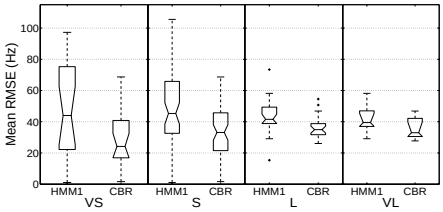


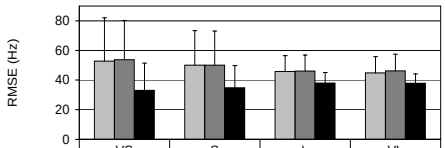
Figure 5: *Mean F0 RMSE for each phoneme and phrase length*

5.2. Subjective evaluation

The aim of the subjective measures (see figure 8) is to test synthesized speech from HMM-TTS using either CBR or HMM based F0 estimators. Figure 8(a) demonstrates that synthesis using CBR or HMM as F0 estimators is equally preferred. However, 8(b) presents CBR as the selected estimator for interrogative while HMM as the preferred for exclamative.

6. Discussion

In order to demonstrate objective results some real examples are presented. For a long and declarative phrase (figure 9) both HMM and CBR estimate a similar F0 contour. On the other hand, in figure 10, CBR reproduces fast changes better when estimating F0 in a short interrogative phrase (e.g. frames around 200). AG and IG factors become a better approach in this case.



	VS	S	L	VL
HMM1	52,87	50,01	45,81	44,87
HMM2	53,75	50,10	46,07	46,27
CBR	33,08	34,89	37,98	37,93

Figure 6: *RMSE for F0 contour and phrase length*

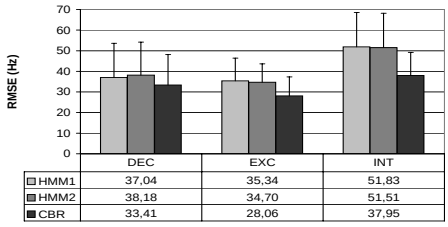


Figure 7: *F0 contour RMSE and phrase type*

Table 4: *Correlation for different length and types of phrase*

	VS	S	L	VL	ENU	EXC	INT
HMM1	0,28	0,40	0,42	**0,55**	0,52	**0,59**	**0,37**
HMM2	0,21	0,37	0,37	0,46	0,47	0,55	0,36
CBR	0,55	0,61	0,55	**0,57**	0,59	0,69	0,61

7. Conclusions and future work

This work presented a Spanish HMM-TTS and compared its performance against CBR for F0 estimation. The HMM system performance has been analysed through objective and subjective measures. Objective measures demonstrated that HMM prosody reproduction has a few dependency on the tree length but an important dependency on the type and length of the phrases. Interrogative sentences which have intense intonational variations are better reproduced by CBR approach. Subjective measures validated HMM-TTS synthesis results with HMM and CBR as F0 estimators. HMM estimates a plain F0 contour which is more suitable for declarative phrases while CBR estimation is selected for interrogatives sentences. This can be explained as CBR approach uses AG and IG attributes to retrieve a changing F0 contour which are better in non-declarative phrases and low contextual information cases.

Moreover, CBR approach presents a computational cost lower to HMM training process although modelling all parameters together in a HMM takes advantage of voice analysis and transformation. Therefore, future HMM-TTS system should include AG and IG information in its features to improve F0 estimation in cases where CBR has demonstrated a better performance.

8. Acknowledgements

This work has been developed under SALERO (IST FP6-2004-027122). This document does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of its content.

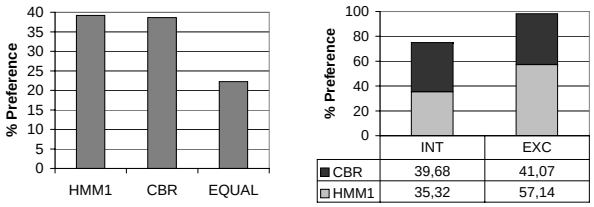


Figure 8: *a) Preference among F0 estimators b) Preference for phrase type and length*

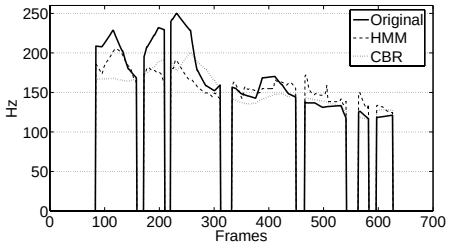


Figure 9: *Example of F0 estimation for HMM-TTS 2nd configuration ("No encuentro la informacin que necesito." translated as "I don't find the information I need.")*

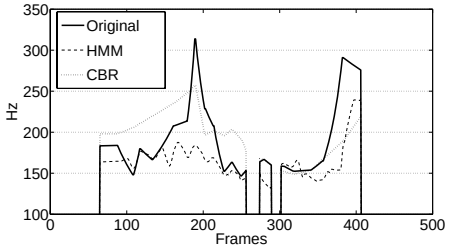


Figure 10: *Example of F0 estimation for HMM-TTS 2nd configuration ("Aburrido de ver pequeñeces?" translated as "Tired of seeing littleness?")*

9. References

[1] Alías, F., Iriondo, I., Formiga, Ll., Gonzalvo, X., Monzo, C., Sevillano, X., "High quality Spanish restricted-domain TTS oriented to a weather forecast application", INTERSPEECH, 2005

[2] Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., Kitamura, T. "Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis", Eurospeech 1999

[3] Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., Kitamura, T., "Speaker interpolation in HMM-based speech synthesis", EUROSPEECH, 1997

[4] Shichiri, K., Sawabe, A., Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., Kitamura, T., "Eigenvoices for HMM-based speech synthesis", ICSLP, 2002

[5] Taylor, P. "Unifying Unit Selection and Hidden Markov Model Speech Synthesis", Interspeech - ICSLP, 2006

[6] Tokuda, K., Zen, H., Black, A.W., "An HMM-based speech synthesis system applied to English", IEEE SSW, 2002

[7] Maia, R., Zen, H., Tokuda, K., Kitamura, T., Resende Jr., F.G., "Towards the development of a Brazilian Portuguese text-to-speech system based on HMM", Eurospeech, 2003

[8] Iriondo, I., Socoró. J.C., Formiga, L., Gonzalvo X., Alías F., Miralles P., "Modeling and estimating of prosody through CBR", JTH 2006 (In Spanish)

[9] Fukada, Tokuda, K., Kobayashi, T., Imai, S., "An adaptive algorithm for mel-cepstral analysis of speech", ICASSP 1992

[10] Alías, F., Monzo, C., Socoró, J.C. "A Pitch Marks Filtering Algorithm based on Restricted Dynamic Programming" InterSpeech - ICSLP 2006

[11] Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., Kitamura, T., "Duration modeling in HMM-based speech sytnhesis system", ICSP 1998

[12] http://www.salle.url.edu/tsenyal/english /recerca/areaparla/tsenyal_software.html

[13] Black, A. W., Taylor, P. Caley, R., "The Festival Speech Synthesis System", http://www.festvox.org/festival

[14] HTS, http://hts.ics.nitech.ac.jp

A Wavelet-Based Technique Towards a More Natural Sounding Synthesized Speech

Mehmet Ataş, Süleyman Baykut, Tayfun Akgül

Department of Electronics and Communications Engineering
Istanbul Technical University, Istanbul, Turkey
atasm@itu.edu.tr, baykut@itu.edu.tr, tayfun.akgul@itu.edu.tr

Abstract

This paper presents a wavelet-based technique to increase the quality and naturalness of LPC based synthesized speech signals. The proposed method is based on wavelet decomposition. We first obtain the wavelet coefficients, and then the variances of the wavelet coefficient at the last four scales (correspond the higher frequency region) of the synthetic speech are replaced by the original variances of the original speech. We apply the technique to synthetic speech. The results suggest that the wavelet-based technique increases the naturalness of the synthesized speech.

1. Introduction

Speech coding based on Linear Predictive Coding (LPC) is a successful and very commonly used method for years [1] and has found many application fields from mobile phones to voice mails [2, 3].

The common problem in the LPC based speech coding is to obtain more realistic speech synthesis at the receiver part. The synthesized speech segments may appear as metallic and buzz like due to the relatively smooth synthetic excitation signals and the use of insufficient number of filter coefficients. When we compare the metallic sounding speech with the natural sounding one, the richness and the naturalness are found to be detailed in the high frequency region of the voiced speech, i.e., the power of the high frequency component of natural speech is mostly higher than the metallic one.

In this study, we use wavelet decomposition to rearrange the energy of the high-frequency components to have more realistic synthesized speech.

First, as commonly used in speech coding, the LP coefficients, voiced/unvoiced decision, the gain and the pitch period (if voiced) is extracted and then they are used to synthesize the speech (at the receiver.) In our proposed method, additionally, variances of the wavelet coefficient at the last four scales (correspond to high frequency region) are extracted. These values are used for adjustment of the wavelet coefficients of the synthesized data yielding rich high frequency components. Experiments with real speech data show that the wavelet-based procedure increases the naturalness of the synthesized speech information.

This paper is organized as follows. In Section 2 we give brief background information on LPC analysis, plus wavelet-based analysis and synthesis. In Section 3, the proposed method is explained. The results are given in Section 4 before the conclusion.

2. Background

In this section LPC analysis and the wavelet–based analysis of speech are briefly explained.

2.1. LPC Analysis

The speech signals, *s(n)*, are assumed to be generated by excitation of a linear filter by a residual source, *r(n)*, as [1]:

$$s(n) = r(n) * h(n) \tag{1}$$

Here *h(n)* is the impulse response of the linear filter that models the vocal tract, * denotes convolution. The filtering procedure in z-domain can be given as:

$$S(z) = R(z)H(z) \tag{2}$$

where *H(z)* is the vocal tract transfer function which is mostly an all-pole model:

$$H(z) = \frac{G}{1 + A(z)} \tag{3}$$

$$A(z) = a_1 z^{-1} + a_2 z^{-2} + \ldots + a_p z^{-p} \tag{4}$$

where G is the gain and a_k are the LP coefficients. Then these coefficients are used for constructing the filter and the inverse filter that model the vocal tract.

2.2. Wavelet-based Analysis of Speech

Wavelet transform is an effective tool used in many signal processing applications. In this study we use the wavelet transform to obtain the variances of the wavelet coefficients which reveal the energy levels at different frequency regions. Orthonormal discrete dyadic wavelet transform (DWT) pair is given below [4]:

$$x(t) = \sum_m \sum_n x_n^m \psi_n^m(t) \tag{5}$$

$$x_n^m = \int x(t)\psi_n^m(t)dt \tag{6}$$

Here x_n^m are the wavelet coefficients, $\psi_n^m(t)$ is the normalized dilations and translations of the mother wavelet function $\psi(t)$, m and n are the dilation (scale) and translation indices, respectively. After obtaining wavelet coefficients along scales we calculate the variances of the wavelet coefficients which will be used in the wavelet-based high frequency adjustment technique.

3. Proposed Method

The block diagram of the speech analysis/synthesis method is shown in Fig. 1. The speech segments are coded and a code vector is formed in the analysis part and it is used by the synthesis part. After generating the synthesized speech, the richness of this speech is increased by changing the variances of the coefficients of the last four scales of the wavelet transform.

The analysis and the synthesis procedures are summarized below.

3.1. Analysis Part

The analysis part has the following steps:

3.1.1. Windowing and Pre-Emphasizing

The sampling frequency of the speech signals used in this study is 16 kHz. The speech signal is segmented into 32ms windows so that the windowed data is now assumed to be stationary.

3.1.2. LP Coefficients

LP analysis is applied to the speech segments and the LP coefficients are stored. In this study the order of the LP analysis is chosen as 16.

3.1.3. Wavelet-Based Analysis

The wavelet coefficients of the speech segments are obtained by equation (6) and the variances are calculated. Variances of the coefficients at the last four scales are stored for the use of the synthesis part.

In our labeling scheme, the higher scale coefficients correspond to the higher frequency regions. Therefore, we additionally store the variances of the last four scales which correspond the $\pi/8 - \pi$ [radian] frequency band for the use of synthesis part.

3.1.4. Inverse Filtering

Inverse of the vocal tract filter is modeled by using the LP coefficients and the inverse filtering is performed to extract the residual signal. The variance of the residual also stored for the synthesis part.

3.1.5. Voiced/Unvoiced Detection

Speech signals can be classified in two basic characteristics: Voiced and unvoiced. Voiced speech signals show a pseudo-periodic structure whereas unvoiced speech signals are white noise-like. The other main difference between the voiced and unvoiced speech is the short-time energy. Voiced speech signals have higher energy than unvoiced speech signals. On the other hand, zero-crossing number of unvoiced speech signals is approximately 10 times higher than voiced speech signals [3]. In this study the voiced/unvoiced decision is made according to these criteria.

3.1.6. Pitch Period Estimation

If the corresponding speech segment is labeled as voiced, the pitch period of the speech is determined by cepstral pitch detection method [2].

3.1.7. Forming the Code Vector

The code vector contains the information to synthesize the speech. LP coefficients are the most important components of the code vector. Other components of the code vector are voiced/unvoiced decision, pitch period, the variance of the speech segment, and the variances of the wavelet coefficients at the last four scales. These values are then use to adjust the high frequency components in the synthesized speech.

3.2. Synthesis Section

The synthesis of speech signals are presented in this section.

3.2.1. Unvoiced Speech Synthesis

The vocal tract filter is modeled by using the LP coefficients. Then the filter is excited by white Gaussian noise with the same variance carried in the code vector. For voiced speech segment, the filter is excited by a periodic residual signal. This signal represents the glottal flow.

3.2.1.1 Glottal Pulse Model

Glottal pulse shape selection is one of the most important aspects of speech synthesis. There are many types of models used to model glottal flow. In this study, commonly used Liljencrants-Fant (LF) model is used to represent the voice source [5].

3.3. High Frequency Adjustment in the Wavelet Domain

The synthesized speech sounds as metallic and buzz like due to the relatively smooth synthetic excitation signals [6, 7]. The high frequency components of the synthesized speech have relatively low energy compared to the original speech. In this study we adjust the high frequency components by replacing the variances of the wavelet coefficients at the last four scales with the ones in the code vector. The procedure is shown in Fig. 2. The resulting speech now sounds more natural and rich.

4. Simulation on Real Speech Data

We apply the wavelet-based technique to 32 ms long synthesized voiced speech segment (/EE/) sampled at 16 kHz. The original, the synthesized and the adjusted high frequency components speech segments are given in Fig. 3- (a), (b) and (c), respectively. Even the visual inspection of Fig. 3-(b) and (c) show that Fig. 3-(c) has more detail (high frequency component) compared to Fig. 3-(b).

We apply the high frequency adjustment technique to longer speech signal. The signal is segmented into 32ms windows. After the voiced/unvoiced decision, and the pre-emphasizing filtering, the LP coefficients are calculated. If the segment is voiced, pitch period is estimated as well. The variances of the residual signals and the variances of the last four scales of the wavelet coefficients are also determined for every segment and the code vector is formed as explained in Section 3.1. In Fig. 4 - (a), the waveform of the Turkish sentence “Sayısal İşaret İşleme” which means “Digital Signal Processing” in English, is given. The sentence is articulated by an adult female speaker. The synthesized speech is given in Fig. 4 - (b) and the speech after our proposed method is applied is given in Fig. 4 - (c). These examples will be made available to listen for the audience during the presentation of this paper in the conference.

5. Conclusions

LPC based speech synthesis makes speech communication possible at low bit rates. However, a problem of metallic and buzz like speech is faced due to the relatively smooth

synthetic excitation signals. In this study, the high frequency components of the synthesized speech is adjusted and enhanced by a wavelet-based technique to improve the naturalness of the synthesized speech. The results show that the technique gives promising results.

6. References

[1] Atal B. S., Hanauer S., "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," *The Journal of the Acoustical Society of America*, vol.50, pp. 637-655, April 1971.

[2] Deller, J. R., Hansen, J. H. I., Proakis, J. G., "Discrete-Time Processing of Speech Signals," *IEEE Press*, New York, NY, 2000

[3] Quateri, T., F., **"**Discrete–Time Speech Signal Processing Principles and Practice," *Prentice Hall Inc.*, 2002.

[4] G. W. Wornell, "Wavelet-Based Representations for the *1/f* Family of Fractal Processes," *Proc. of IEEE*, vol. 81, no. 10, pp. 1428-1450, Oct. 1993.

[5] Fant, G. and Lin, Q., "A Four Parameter Model of Glottal Flow," *STL-QPSR*, 85(2), 1-13., 1988.

[6] Lee, Y., **"**Speech Quality Enhancement by Exploring 1/*f* Nature of Speech Residual," *MS Thesis*, Drexel University, PA, 1998.

[7] Aoki, N., Ifukube, T., "Enhancing the Naturalness of Synthesized Speech by Using the Random Fractalness of Vowel Source Signals," *Electronics and Communications in Japan,* vol. 84, no. 1*, pp*. 11-20., 2001.

7. Figures

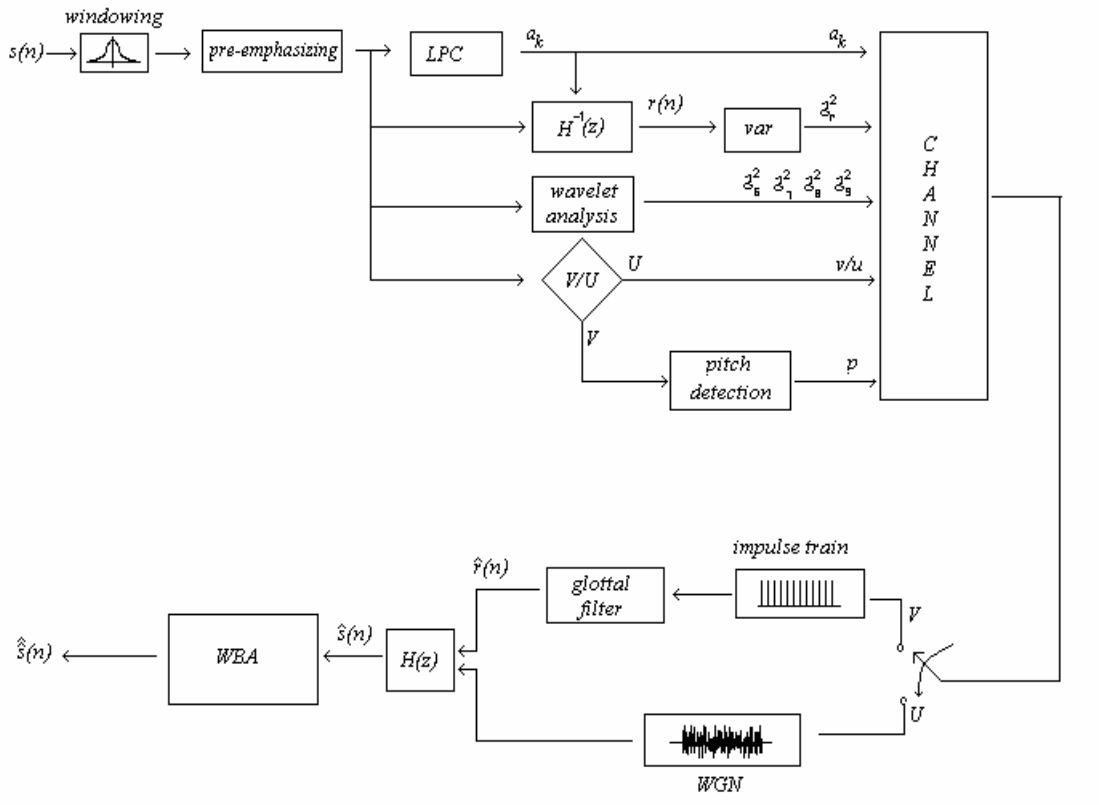


Figure 1: Block Diagram of the Speech Analysis/Synthesis Procedure.

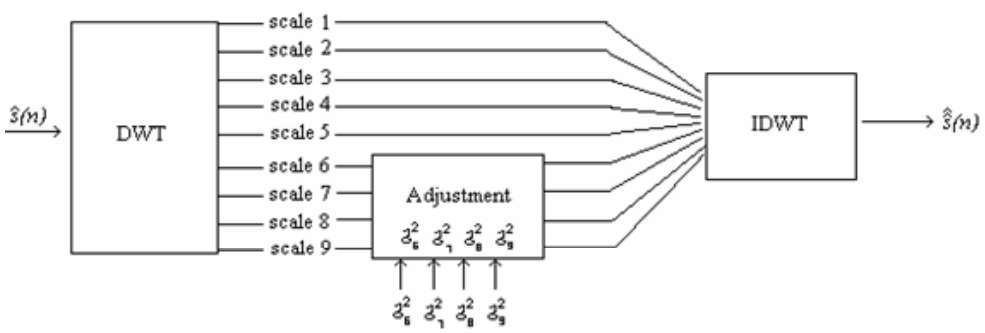


Figure 2: Wavelet-based high frequency component adjustment.

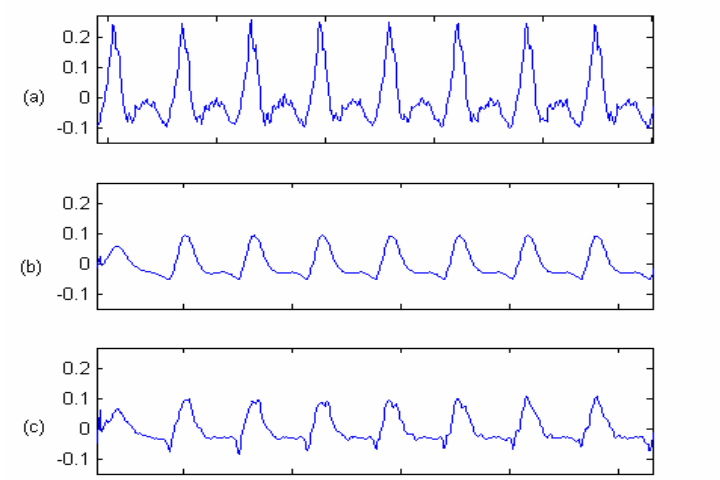


Figure 3: (a) A voiced speech segment /EE/ from the original speech data, (b) synthesized speech segment, (c) synthesized speech segment with the adjusted high frequency components.

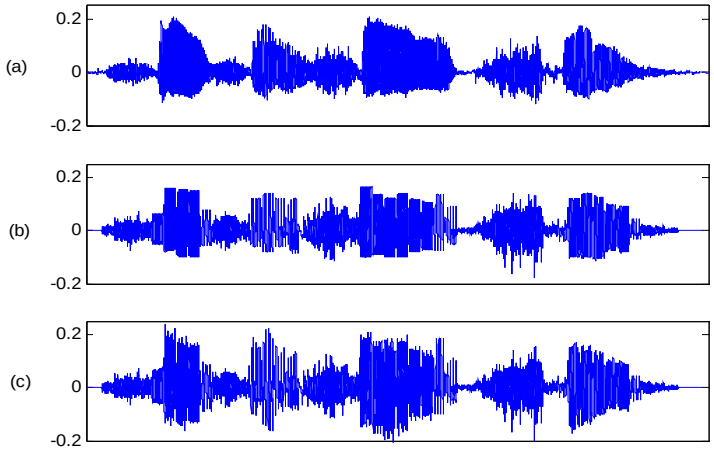


Figure 4: (a) The original speech waveform of "Sayısal İşaret İşleme" (in Turkish), (b) the synthesized speech, (c) resulting speech after the high frequency adjustment process.

Objective and Subjective Evaluation of an Expressive Speech Corpus

Ignasi Iriondo, Santiago Planet, Joan-Claudi Socoró, Francesc Alías

Department of Communications and Signal Theory
Enginyeria i Arquitectura La Salle, Ramon Llull University, Barcelona, Spain
{iriondo, splanet, jclaudi, falias}@salle.url.edu

Abstract

This paper presents the validation of the expressive content of an acted oral corpus produced to be used in speech synthesis. Firstly, objective validation has been conducted by means of automatic emotion identification techniques using statistical features obtained from the prosodic parameters of speech. Secondly, a listening test has been performed with a subset of utterances. The relationship between both objective and subjective evaluations is analysed and the obtained conclusions can be useful to improve the following steps related to expressive speech synthesis.

1. Introduction

There is a growing tendency towards the use of speech in human-machine interaction. Automatic speech recognition is used to consult information or to make managements. Speech synthesis let machines to communicate orally with users (automation of services or aid to disabled people). The incorporation of the recognition of emotional states or the synthesis of emotional speech can improve the communication by doing it more natural [1]. Therefore, one of the most important challenges in the study of the expressive speech is the development of oral corpora with authentic emotional content that enable robust analysis according to the task for which they have been developed. It is not the objective of the present work to carry out an exhaustive summary of the available databases for the study of emotional speech, since recently, complete studies have appeared in the literature. In [2], a new compilation of 48 databases is presented showing a notable increase of multimodal databases. In [3], the databases used in 14 experiments of automatic detection of the emotion are summarized. Finally, in [4] a revision of 64 databases of emotional speech is done, providing a basic description of each one and its application.

This paper describes the main aspects of the production of an expressive speech corpus in Spanish faced to synthesis and the objective and subjective evaluation of its emotional content. Section 2 introduces different aspects about the corpora of expressive speech. Section 3 explains the production of our corpus. Section 4 details the process of objective validation carried out through techniques of automatic identification of the emotion. The subjective evaluation by means of perception test is explained in Section 5, and finally, the conclusions (Section 6).

2. Building emotional speech corpora

According to [5], there are four main aspects to be considered in the development of an emotional speech corpus: *i)* the **scope** that covers the database (number of speakers, the language, dialects, genre of the speakers and types of emotional states); *ii)* the **context** in which a locution takes place (emotional significance perceived across the semantics, the prosody, the facial expression, gestures and posture); *iii)* the **descriptors** that allow to represent the linguistic, emotional and acoustic content of the speech; and *iv)* the **naturalness** of the locutions, which will depend on the strategy followed to obtain the emotional speech. With respect to the latter, the main debate is centred on the compromise between authenticity of the expressed emotion and the control on the recording. Campbell [1] and later Schröder [6] propose 4 emotional speech sources:

Natural occurrences. Spontaneous human interaction presents the most natural emotional speech although it has some drawbacks due to the lack of control on its content, the quality of sound, the difficulty of labelling, and finally, legal and ethical aspects (e.g. *The Reading-Leeds*, *The Belfast Naturalistic* and *The CREST* databases described in [5]).

Elicitation. The provocation of authentic emotions in people in the laboratory is a way of compensating some of the problems described previously, although the fullblown emotions would remain out of place [1]. In [6], five types of mood induction procedures are described.

Stimulated emotional speech. This method consists of the reading of texts with a verbal content adapted for the emotion to be expressed. The difficulty of comparing utterances with different texts should be counteracted with an increase of the corpus size so that statistical methods allow to generalize models [1]. This technique was followed in the creation of the *Belfast Structured Emotion Database* [5].

Acted emotional speech. The great advantage of this method is the control of the verbal and phonetic content of speech since all the emotional states can be produced using the same phrases. This allows direct comparisons of the phonetics, the prosody and the voice quality for the different emotions. The great objection that presents is the lack of authenticity of the expressed emotion [1].

Another important aspect to keep in mind is the purpose of the speech and emotion research. It is necessary to distinguish between processes of perception (*centred on the speaker*) and expression (*centred on the listener*) [6]. The objective of the former is to establish the relation between the speaker emotional state and quantifiable parameters of speech. Usually, they deal with the recognition of emotions from speech signal. According to [3], one of the challenges is the identification of oral indicators (prosodic, spectral and vocal quality) attributable to the emotional behavior and that are not simply own characteristics of conversational speech. The latter model the parameters of the speech with the goal to transmit a certain emotional state. The description of emotional states and the choice of speech parameters are key in the final result. There is a high consen-

This work has been partially supported by the European Commission, project SALERO FP6 IST-4-027122-IP.

sus in the scientific community for obtaining emotional speech by means of stimulated/acted speech for synthesis purposes [5] [2], although other authors argue in favour of constructing an enormous corpus gathered from recordings of the daily life of a number of voluntary speakers [7].

This work combines methods of both types of studies. On the one hand, the production of the corpus follows the guidelines of the studies *centred on the listener* since it is oriented to speech synthesis. On the other hand, we apply techniques of emotion recognition in order to validate its expressive content.

3. Our expressive speech corpus

We considered the development of a new expressive oral corpus for Spanish due to lack of availability of a corpus with the suitable characteristics within the framework of our research in expressive speech synthesis. This corpus had a double purpose: to learn the acoustic models of emotional speech and to be used as the speech unit database for the synthesizer. This section describes the steps followed in the production of the corpus.

3.1. Stimulated emotion and text design

For the recording of the present corpus, a female professional speaker has been chosen due to her capability to use the suitable expressive style to each text category (stimulted/acted speech).

For the design of texts semantically related to different expressive styles, we have made use of an existing textual database of advertisements extracted from newspapers and magazines. Based on a study of the voice in the audio-visual publicity [8], five categories of the textual corpus have been chosen and the most suitable emotion/style has been assigned to them: New technologies (neutral-mature), education (joy-elation), cosmetic (style sensual-sweet), automobiles (aggressive-hard) and trips (sad-melancholic).

A set of phrases has been selected from each category by means of a *greedy* algorithm [9] that has allowed to obtain a phonetic balance in each subcorpus. This type of algorithms take the locally optimum choice at each stage with the hope to find an adequate global solution. Therefore, the application of this algorithm to the raised problem will obtain a valid solution, although may be not the optimum one. In addition to looking for a phonetic balance, phrases that contain exceptions (e.g. foreign words, abbreviations) have been avoided due to they make difficult the automatic processes of phonetic transcription and labelling. Moreover, the selection of similar phrases to others previously selected has been penalized by the *greedy* algorithm.

3.2. Recording

The recording of the oral corpus has been carried out in a professional recording studio. Speech signals were sampled at 48 KHz and quantized using 24 bits per sample and stored in WAV files. Different recording sessions have been required and therefore a preestablished protocol has been followed in order to minimizing errors that can cause deficiencies in the corpus labelling. For the corpus segmentation in phrases, a semiautomatic process has followed by means of a forced alignment using Hidden Markov Models from the phonetic transcription and later a manual review and correction. This forced alignment also has been used to segment the phrases in phonemes. The recorded database has 4638 sentences and it is 5 h 12 min long.

4. Objective validation

The goal of the experiments described in this section was to validate the expressive content of the corpus by means of automatic emotion identification using different data mining techniques applied to statistical features computed over the prosodic parameters of speech. An exhaustive subjective evaluation of the full corpus (more than 5 hours of speech) would be a very tedious task and practically impossible. However, the whole corpus can be validated by means of these automatic techniques.

4.1. Acoustic analysis

Prosodic features of speech (fundamental frequency, energy, duration of phones and frequency of pauses) are related to vocal expression of emotion [10]. In this work, an automatic acoustic analysis of the sentences is performed using information of the previous phonetic segmentation.

4.1.1. F0 related parameters

The analysis of the fundamental frequency (F0) parameters is based on the result of the pitch marker described in [11]. This system assigns marks over the whole signal. The unvoiced segments and silences are marked using interpolated values from the neighboring voiced segments. For each phrase, three vectors of local F0 values are obtained (complete, excluding silences and unvoiced sounds, and only the stressed vowels). The information about the boundaries of voiced/unvoiced (V/UV) segments and silences is obtained from the corpus labelling. Notice that if the phonetic segmentation was not available, an automatic voice-activity detector (VAD) and a V/UV detector would be required [12]. Moreover, F0 has been calculated in both lineal and logarithmic scales.

4.1.2. Energy related parameters

For energy, speech is processed with 20-ms rectangular windows and 50% of overlap, calculating the power (linear and dBs) every 10ms. Following the same idea that for F0, three vectors per utterance are generated (complete, excluding silences, and only in the stressed vowels).

4.1.3. Rhythm related parameters

The duration of phones is an important cue for vocal expression of emotion. However, some studies omit this parameter by the difficulty to obtain it automatically [12]. In the present work we have incorporated this information (thanks to the labelling of the corpus) to generate datasets with and without this information in order to contrast its relevance. Z-scores have been employed for duration modeling in text-to-speech synthesis (TTS) to predict individual segment durations and to control the lengthening or the shortening of phones. As in [13], we take z-scores as a means to analyze the temporal structure of speech:

$$z\_score = \frac{dur(ms) - \mu}{\sigma} \quad (1)$$

where μ and σ are the mean and the standard desviation respectively of the corresponding phoneme. Therefore, the rhythm of an utterance is represented by a vector with the z-score of each phoneme. The version of this vector with only the stressed vowels is also computed.

Moreover, two pausing related parameters are added for each utterance: the number of pauses per time unit and the percentage of silence respect to the total time.

4.2. Statistical analysis and datasets

The prosody of an utterance is represented by some sequencies of values by phoneme such as F0 (lineal and logarithmic), energy (lineal and dB) and normalized durations (z-score). For each sequence, the first and the second derivative are calculated. For all these resulting sequences, the following statistics are obtained: mean, variance, maximum, minimum, range, skew, kurtosis, quartiles, and interquartilic range. Finally, 464 parameters by utterance are calculated, considering both parameters related to the pausing.

This set of parameters has been divided into different subsets according to different strategies to reduce the dimensionality (see the diagram of the figure 1). A first criterion to reduce it has been to omit the second derivative (from Data1 to Data2) in order to valorate the significance of this function. Secondly, preliminary experiments have shown that the use of the logarithmic versions of F0 and energy obtain better results. For this reason, two new datasets have been generated without the linear versions of both F0 and energy. Each one of these datasets (Data1L and Data2L) has been divided in two new sets considering all the phonemes or only the stressed vowels. Moreover, an automatic reduction of both initial datasets (with and without the 2nd derivative) has been carried out by means of the simple genetic algorithm (GA) implemeted in Weka [14] (Data1G and Data2G). This reduction is independent of the later classification algorithm and therefore all the techniques have been tryed with these datasets. Finally, two similar datasets to *Navas et al. (2006)* [12] have been generated to test the significance of omiting the timing parameters (Data1N and DATA1NG).

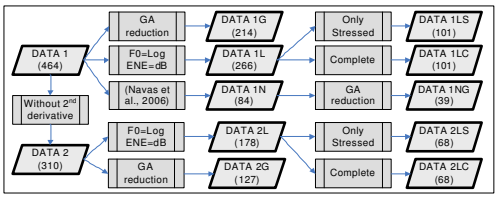


Figure 1: Generation of different datasets

4.3. Experiments and results

Numerous schemes of automatic learning can be used in a task such as classifying the style/emotion from the acoustic analysis of the speech. The objective evaluation of expressivity in our speech corpus is based on [15], where a large-scale data mining experiment about the automatic recognition of basic emotions in short utterances was conducted. After different preliminary experiments, the set of machine learning algorithms showed in table 1 has been selected in order to be tested with the different datasets. Some algorithms have been completed with their *boosted* versions that achieve better results although they present a greater computacional cost. All the experiments have been carried out using Weka software [14] by means of ten-fold cross-validation. Both tryed versions of SMO (Support Vector Machine of Weka) obtain the best results so much on average as in maximum value (see table 1). SMO algorithms achieve the highest results with Data1G, showing that the dimensionality reduction based in GA helps to these systems, although differences with Data1L and Data1LC are minimum. However, other algorithms(i.e. J48, IB1 and IBk) work better with datasets genererated by two consecutive reductions (without 2nd derivative and latter GA reduction). And finally, we can observe that there is a third group of algorithms that work better if the linear/logarithic redundance of F0 and energy is removed. Also we can observe that the boosted versions improve significantively the results respect to their corresponding algorithms. Figure 2 shows a comparison between different datasets depending on the algorithm. Notice that Data1LC obtains almost the same results than Data1G and Data1L, but with less than the half of parameters. The same effect is presented in the datasets without de 2nd derivative. Results experiment a slight loss when timing parameters are removed (Data1N and Data 1NG). However, results worsen significantly when parameters are calculated only in the stressed vowels (Data1LS and Data2LS). Table 2 shows the confusion matrix with the average results for the eleven classifiers with Data2G, that has achieved the best mean percentage of identification (97.02 % $\pm$ 1.23).

Table 1: Learning Algorithms used for the automatic recognition experiment

Name	Description	mean(95%CI)	max(Data)
J48	Decision tree based on C4.5	93.4 ± 2.0	96.4 (2G)
B.J48	Adaboosted version of J48	96.4 ± 1.4	98.3 (1L)
Part	Decision Rules (PART)	94.2 ± 2.0	96.9 (2L)
B.Part	Adaboosted version of PART	96.7 ± 1.3	98.4 (1G)
DT	Decision Table	88.7 ± 2.6	92.3 (1L)
B.T	Adaboosted version of D. T.	93.4 ± 1.6	96.1 (1L)
IB1	Instance-based (1 solution)	93.3 ± 2.8	97.5 (2G)
IBk	Instance-based (k solutions)	94.0 ± 2.3	97.9 (2G)
NB	Naive Bayes with discretization	94.6 ± 1.9	97.8 (1L)
SMO1	SVM with 2nd degree pol. Kernel	97.3 ± 1.2	99.0 (1G)
SMO2	SVM with 3rd degree pol. Kernel	97.1 ± 1.5	98.9 (1G)

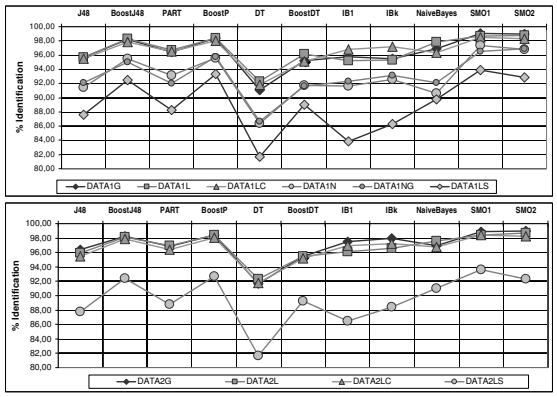


Figure 2: Identification percentage for the ten tested datasets

5. Subjective evaluation

A subjective evaluation is a tool that allows to validate the expressivity of acted speech from a point of view of the users. An exhaustive evaluation of corpus would be excessively tedious (the corpus has 4635 utterances). For each style, 96 utterances have been chosen, having done a total of 480. This test set has been divided in 4 subsets, having 120 utterances each one. An ordered pair of subsets has been assigned to each subject. Therefore, 12 different tests have been generated. The allocation of ordered pairs tries to compensate the fact that the second test could be easier to evaluate due to the previous training.

A forced answer test has been designed with the question *¿What emotional state do you recognize from the voice of the*

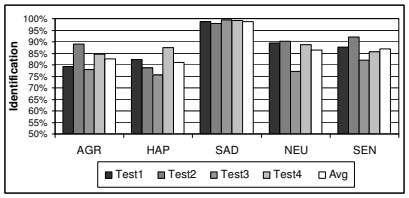


Figure 3: Percentage of identification depending on the test

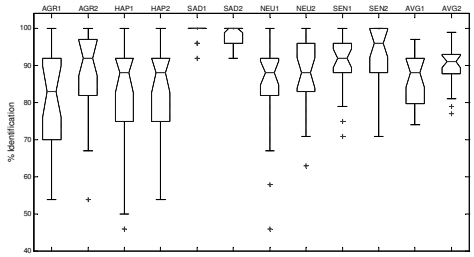


Figure 4: Boxplots of sorted pairs first-second round depending on the style and the average

speaker in this phrase?. The possible answers are the 5 styles of the corpus plus one more option *Don't know / Another*, with the objective of avoid insecure or erroneous answers for the confusing cases. Adding this option has the risk that some evaluators use excessively this answer to accelerate the end of the test [12]. However, this effect has not been considerable in this test.

The process of evaluation has been carried out on a web platform developed for this type of tests, that permits to leave the test and to resume it subsequently. The evaluators belong mainly to the staff of *Enginyeria i Arquitectura La Salle* with a quite heterogeneous profile. Only the results of the 26 volunteers who finished the two tests have been taken into account.
The results of the subjective test show that all the styles achieve a high percentage of identification. The figure 3 shows the percentage of identification by style and test, being the sad style the best rated (98.8% of average), followed by sensual (86.8%) and neutral (86.4%) styles, and finally the aggressive (82.7%) and happy (81%) ones.

The confusion matrix (table 3), shows that the main errors are in the agressive style (14.2% identified as happy) and the happy one (15.6% identified as aggressive). Moreover, neutral style is confused slightly with all and there is certain confusion of sensual with sad (5.7%). If we compare these results with the confusion matrix for the best average rated dataset (table 2), we can conclude that the algorithms confuse mainly sensual with neutral, however subjects show confusions between happy and aggressive. This difference is due to the lack of voice quality parameters because sadness and neutral have similar prosody, but sensual voice is most whispered than neutral, a difference which is clearly noticed by the subjects. Also, the influence of order has been studied. In average, the second round obtains better results than the first, especially for neutral, sensual, and aggressive styles (see figure 4).

6. Conclusion and future work

In this paper, the production of an oral corpus oriented to expressive speech synthesis has been presented. We have performed subjective (listening test) and objective (automatic emotion identification) evaluation in order to validate its expressive content showing good results. The advantage of the automatic experiments is that they are performed over the whole corpus, while the listening test comprises a subset of utterances.

In future, we will introduce voice quality parameterization in addition to prosody to minimize the confusion between sensual and neutral styles. Moreover, this work should serve to analyze the bad classified utterances in order to eliminate them and to improve the latter modelling and synthesis processes.

Table 2: Average confusion matrix for the automatic identification experiment with Data2G and the eleven algorithms

	Agr	Hap	Sad	Neu	Sen
AGR	**99.1%**	0.8%	0.1%	0.0%	0.0%
HAP	1.6%	**97.1%**	0.0%	1.2%	0.2%
SAD	0.2%	0.1%	**99.3%**	0.4%	0.1%
NEU	0.2%	0.9%	0.4%	**93.9%**	4.5%
SEN	0.0%	0.1%	0.2%	4.9%	**94.8%**

Table 3: Average confusion matrix for the subjective test

	Agr	Hap	Sad	Neu	Sen	Dk/A
AGR	**82.7%**	14.2%	0.1%	1.8%	0.1%	1.1%
HAP	15.6%	**81.0%**	0.1%	1.9%	0.2%	1.2%
SAD	0.0%	0.0%	**98.8%**	0.5%	0.6%	0.1%
NEU	5.3%	1.3%	0.7%	**86.4%**	3.6%	2.7%
SEN	0.0%	0.1%	5.7%	4.7%	**86.8%**	2.6%

7. References

[1] N. Campbell, "Databases of emotional speech," *Proceedings of the ISCA Workshop on Speech and Emotion*, pp. 34–38, September 2000.

[2] R. Cowie, E. Douglas-Cowie, and C. Cox, "Beyond emotion archetypes: databases for emotion modelling using neural networks," *Neural Networks*, vol. 18, pp. 371–388, 2005.

[3] L. Devillers, L. Vidrascu, and L. Lamel, "Challenges in real-life emotion annotation and machine learning based detection," *Neural Networks*, vol. 18, pp. 407–422, 2005.

[4] D. Ververidis and C. Kotropoulos, "Emotional speech recognition: Resources, features, and methods," *Speech Communication*, vol. 48, no. 9, pp. 1162–1181, September 2006.

[5] E. Douglas-Cowie, N. Campbell, R. Cowie, and P. Roach, "Emotional speech: towards a new generation of databases," *Speech Communication*, vol. 40, pp. 33–60, April 2003.

[6] M. Schröder, "Speech and emotion research: An overview of research frameworks and a dimensional approach to emotional speech synthesis," Ph.D. dissertation, PHONUS 7, Research Report of the Institute of Phonetics, Saarland University, 2004.

[7] N. Campbell, "Developments in corpus-based speech synthesis: Approaching natural conversational speech," *IEICE - Trans. Inf. Syst.*, vol. E88-D, no. 3, pp. 376–383, 2005.

[8] N. Montoya, "El papel de la voz en la publicidad audiovisual dirigida a los niños," *Zer. Revista de estudios de comunicación*, no. 4, pp. 161–177, 1998.

[9] H. François and O. Boëffard, "The greedy algorithm and its application to the construction of a continuous speech database," in *Proceedings of LREC*, vol. 5, May 29-31 2002, pp. 1420–1426.

[10] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. G. Taylor, "Emotion recognition in human computer interaction," *IEEE Signal Processing*, vol. 18, no. 1, pp. 33–80, January 2001.

[11] F. Alías, C. Monzo, and J. C. Socoró, "A pitch marks filtering algorithm based on restricted dynamic programming," in *International Conference on Spoken Language Processing (ICSLP)*, pp. 1698–1701.

[12] E. Navas, I. Hernáez, and I. Luengo, "An Objective and Subjective Study of the Role of Semantics and Prosodic Features in Building Corpora for Emotional TTS," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1117–1127, July 2006.

[13] A. Schweitzer and B. Möbius, "On the structure of internal prosodic models," in *Proceedings of the 15th International Congress of Phonetic Sciences (Barcelona)*, 2003, pp. 1301–1304.

[14] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco: Morgan Kaufmann, 2005.

[15] P.-Y. Oudeyer, "The production and recognition of emotions in speech: features and algorithms," *Int. Journal of Human Computer Interaction*, vol. 59, no. 1-2, pp. 157–183, 2003, special issue on Affective Computing.

ON THE USEFULNESS OF LINEAR AND NONLINEAR PREDICTION RESIDUAL SIGNALS FOR SPEAKER RECOGNITION[1]

Marcos Faundez-Zanuy

Escola Universitaria Politécnica de Mataró, UPC Barcelona, SPAIN

ABSTRACT

This paper compares the identification rates of a speaker recognition system using several parameterizations, with special emphasis on the residual signal obtained from linear and nonlinear predictive analysis. It is found that the residual signal is still useful even when using a high dimensional linear predictive analysis. On the other hand, it is shown that the residual signal of a nonlinear analysis contains less useful information, even for a prediction order of 10, than the linear residual signal. This shows the inability of the linear models to cope with nonlinear dependences present in speech signals, which are useful for recognition purposes.

Index Terms— Neural networks, speaker recognition, nonlinearities, prediction methods

1. INTRODUCTION

Several parameterization techniques exist for speech [17] and speaker [15] recognition, cepstral analysis and its related parameterizations such as Delta-Cepstral features, Cepstral Mean Subtraction, etc. being the most popular.

There are two main ways to compute the cepstral coefficients and one important drawback in both cases: relevant information is discarded, as follows.

1. LP-derived cepstral coefficients. The linear prediction analysis produces two main components, the prediction coefficients (synthesis filter) and the residue of the predictive analysis. This latter signal is usually discarded. However, experiments exist [9] where it is shown that human beings are able to recognize the identity of the speaker listening to residual signals of LP analysis. Based on this fact several authors have evaluated the usefulness of the LPC-residue and have found that although the identification rates using this kind of information alone does not perform as well as the LP-derived cepstral coefficients, a combination of both can improve the results [20,12,14,22,11].
2. Fourier Transform derived cepstral coefficients. Instead of working out a set of Linear prediction coefficients, are based on the power spectrum information, where phase information has been discarded. [19] proposed the use of new acoustic features based on the short-term Fourier phase spectrum. The results are similar to the LP-derived cepstral coefficients. Although these (phase spectrum) features cannot outperform the classical cepstral parameterization, the results are improved using a combination of both features.

In this paper we will focus on the first kind of parameterization, because they are a clear alternative to the nonlinear predictive models, which have shown an improvement over the classical linear techniques in several fields (for a recent overview about these techniques [7]).

In [4,6] we proposed a new set of features and models based on these types of nonlinear models and an improvement was also found when this information was combined with the traditional cepstral analysis, but so far, the relevance of the residual signals from linear and nonlinear predictive analysis has not been studied and compared.

In this paper we will study if the relevance of the residual signal is due to an insufficient linear predictive analysis order or because of the incapability of the linear analysis to model nonlinearities present in speech and demonstrate is usefulness for speaker recognition purposes. This important question has not been solved in previous papers that focus on a typical 8 to 16 prediction order.

2. EXPERIMENT SET UP

2.1. Database

For our experiments we have used the Gaudi database [16]. We have used one subcorpora of 49 speakers acquired with a simultaneous stereo recording with two different microphones. The speech is in wav format with a sampling frequency (f_s) = 16 kHz, 16 bit/sample and the bandwidth is 8 kHz.

From this database we have generated narrow-band signals using the potsband routine that can be downloaded from [21]. This function meets the specifications of G.151 for any sampling frequency. Thus, our study has been performed on telephone bandwidth.

2.2. Identification algorithm

In this study, we are only interested in the relative performance between linear and nonlinear analyses. Thus, we have chosen a simple algorithm for speaker recognition.

In the training phase, we compute, for each speaker, empirical covariance matrices based on feature vectors extracted from overlapped short time segments of the speech signals. As features representing short time spectra we use both linear prediction cepstral coefficients (LPCC) and mel-frequency cepstral coefficients melceps [3]. In the speaker-recognition system, the trained covariance matrices for each speaker are compared with an estimate of the covariance matrix obtained from a test sequence from a speaker. An arithmetic-harmonic sphericity measure is used in order to compare the matrices [1]:

[1] This work has been supported by FEDER and MEC TIC-2003-08382-C05-02, TEC2006-13141-C03-02/TCM

$d = \log\left(\mathrm{tr}(C_{test} C_j^{-1})\,\mathrm{tr}(C_j C_{test}^{-1})\right) - 2\log(l)$, where $\mathrm{tr}(\cdot)$ denotes the trace operator, l is the dimension of the feature vector, C_{test} and C_j is the covariance estimate from the test speaker and speaker model j, respectively.

2.3. Parameterizations

We have used the following parameterizations

1. LP-derived cepstral coefficients (LPCC)
2. Fourier transform derived cepstral coefficients (melceps)
3. LP- residue coefficients

The first two first parameterizations can be found, for instance, in [17,15,3], while the third is proposed in [11] and will be described in more detail next.

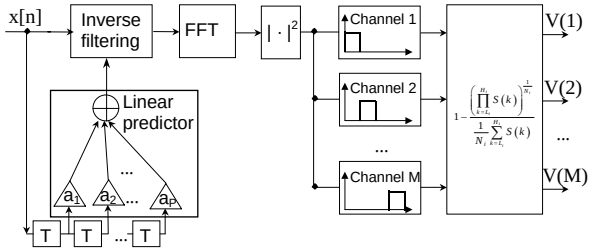


Figure 1. LP residual signal parameterization

Feature extraction from the LP-residual signal

We will use the Power Difference of Spectrum in Subband (PDSS) obtained as follows [11]:

1. Calculate the LP-residual signal using the P[th]-order linear prediction coefficients.
2. Calculate the Fast Fourier Transform (fft) of the LP-residual signal using zero padding in order to increase the frequency resolution: $S = \left| fft(residue) \right|^2$
3. Group power spectrum into P subbands.
4. Calculate the ratio of the geometric to the arithmetic mean of the power spectrum in the i[th] subband, and subtract it from 1:

$$PDSS(i) = 1 - \frac{\left(\prod_{k=L_i}^{H_i} S(k)\right)^{\frac{1}{N_i}}}{\frac{1}{N_i}\sum_{k=L_i}^{H_i} S(k)}$$, where $N_i = H_i - L_i + 1$ is the sample number of frequency points in the i[th] subband and L_i, H_i is the lower and upper limit of frequency in i[th] subband respectively. We have used the same bandwidth for all the bands.

PDSS can be interpreted as the subband version of spectral flatness measure for quantifying the flatness of the signal spectrum. Figure 1 summarizes the procedure.

3. NEW POSSIBILITIES USING NON-LINEAR PREDICTIVE ANALYSIS

Although the relevance of residual NL-predictive analysis for speaker recognition has not been studied previously, nonlinear predictive analysis has been widely studied in the context of speech coding. For instance, [5] revealed that a forward ADPCM scheme with nonlinear prediction can achieve the same Segmental Signal to Noise Ratio (SEGSNR) as the equivalent linear predictive system (same prediction order) with one less quantization bit.

We propose an analogous scheme replacing the linear predictor with a nonlinear predictor. Figure 2 shows the scheme.

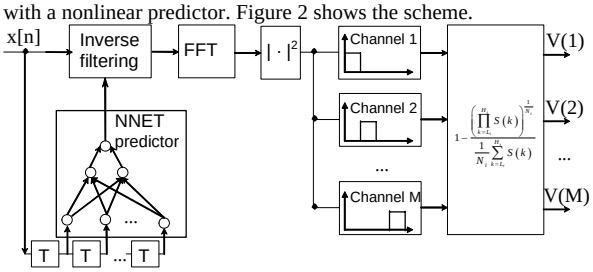


Figure 2: Block diagram used to calculate PDSS parameters from NL-prediction residual signal.

We have used a Multi-Layer Perceptron (MLP). The structure of the neural net has 10 inputs, 2 neurons in the hidden layer, and one output. The selected training algorithm was the Levenberg-Marquardt [10]. The number of epochs has been set up to 6. First layer and hidden layer transfer functions are tansig, while the output layer is linear.

4. EXPERIMENTAL RESULTS

Obviously one important question when dealing with residual LP signals is: Is the information contained in this residual signal coming from an insufficient predictive analysis order? That is, what happens when the prediction analysis order is so high that it is not possible to extract more relevant information using a linear analysis?

The experimental approach used to solve this question is to use a number of LP coefficients higher than usual. Two possible results can be obtained:

1. When the analysis order is increased, the discriminative power of the residual signal is reduced to simple chance results. This means that there is potential for speaker recognition rate improvements through extraction of the LP coefficients in a more efficient manner, probably by increasing the number of coefficients.
2. When the analysis order is increased, the residual signal still contains useful information. This means that a linear analysis is unable to extract this information, and there is room for improvement combining parameterizations defined on the LP coefficients and the residual signal. In order to obtain the optimal results, both signals should be extracted and optimized jointly.

Figure 3 shows the results obtained with the following parameterizations: Melcepstrum, LPC –P residue, LPCC, LPC-80 residue, MLP 10x2x1 and several combinations between them.

LPC-P residue is the parameterization obtained from the residual P-analysis order.

It is interesting to observe the following:

- The residual signal of an LPC-80 analysis can produce a recognition rate higher than 80% for a 15 dimensional vector extraction. Thus, it was found that the residual signal of a LP analysis contains relevant information, and this is due to the inability to extract this information using a linear analysis (80[th] order analysis is enough to model short term and long

term dependencies between samples, but if the analysis is linear, it is limited to linear dependencies).

- The residual signal of a nonlinear predictive analysis, as expected, produces the lower recognition rates, because the relevant information has been retained in the predictor coefficients. However, a maximum of 70% recognition rate is possible.

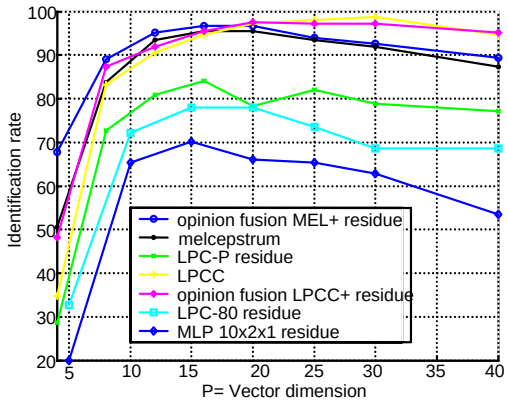


Figure 3: Identification for several parameterization algorithms.

4.1. Opinion fusion

One way to improve the results is by means of a combination of different classifiers opinion [13,8]. In our case, we will use the same classifier scheme, but different parameterizations. In order to study the complementarity of the parameterizations studied, we have computed the correlation coefficient and scatter diagrams.

Table 1 shows the correlation coefficients between distances of several parameterizations. The higher the correlation, the smaller the complementarity of both measures. Figure 2 shows a scatter diagram, which represents points on a two-dimensional space. The coordinates correspond to the obtained distance measures, which correspond to each parameterization (one in each axis). Looking at the diagram we observe that the points diverge from a strip. Thus, they have complementary information and can be combined in order to improve the results.

Table 1: Correlation coefficients between obtained distance values for P=20

	LPCC	Mel-ceps	LP-20 resid	LP-80 resid	MLP 10x2x1
LPCC		0,79	0,68	0,52	0,55
melceps	0,79		0,69	0,56	0,62
LP-20 resid	0,68	0,69		0,78	0,64
LP-80 resid	0,52	0,56	0,78		0,60
MLP 10x2x1	0,55	0,62	0,64	0,60	

When combining different measures, special care must be taken for the range of the values. If they are not commensurate, some kind of normalization must be applied. We have tested the following, based on a sigmoid function [18], $o_i' = \frac{1}{1+e^{-k_i}}$ where: $k_i = \frac{o_i - (m_i - 2\sigma_i)}{2\sigma_i}$, $o_i' \in [0,1]$, and o_i is the initial opinion of the i^{th} classifier. m_i, σ_i are the mean and standard deviation of the opinions of the i classifier, obtained with data from the authentic speakers (intra-model distances).

We have limited the combinations to the outputs of two different classifiers, and the sum and product combination rules [13].

Table 2. Identification rates (combinations with sum rule)

P / Param.	5	10	15	20	25	30	40
LPCC	46.9	90.6	93.5	97.1	98.0	98.8	94.7
Melceps	65.7	89.8	92.7	95.5	93.5	91.8	87.4
LP-P resid	44.1	75.9	84.1	78.4	82.0	78.8	77.1
LP-80 resid	32.7	72.2	78.0	78.0	73.5	68.6	68.6
MLP resid	20.0	65.3	70.2	66.1	65.3	62.9	53.5
LPCC+LP-P	64.9	89.8	94.7	97.6	97.1	97.1	95.1
LPCC+MLP	51.0	91.4	95.1	97.1	97.96	98.4	95.1

We have experimentally observed that slightly better results are obtained without normalization. Looking at figure 4 it can be seen that the distance values obtained with the residual signal parameterization have less amplitude (about 2 to 3 times). Thus, if the normalization is not done, it is equivalent to a weighted combination where the LPCC distances have more influence over the combined result than the residual signal.

Figure 3 and table 2 summarize the identification rates for several vector dimensions (*P*) and different combined parameters.

5. CONCLUSIONS

So far several papers have established that a combination between classical parameters (LPCC, melceps) with some kind of parameterization computed over the residual analysis signal can yield improvements in recognition rates. In our experiments we have found that this is only true when the analysis order ranges from 8 to 16. These values have been selected mainly because a spectral envelope can be sufficiently fitted with this amount of data, so there was no reason to increase the number of parameters. Although we consider that this is true for speech analysis, synthesis and coding, it is interesting to observe that the parameterization step for a speaker recognition system is twofold:

1. We make a dimensionality reduction, so it is easier to compute models, distances between vectors, etc.
2. We make a transformation from one space to another one. In this new domain, it can be easier to discriminate between speakers, and some parameterizations are better than others.

Thus, we are not looking for good quality representation of the speech signal (or a compromise between good representation with the smallest number of parameters). We are just looking for good discrimination capability.

In our experiments we have found that for parameter vectors of high order, although the residual signal has a significant discriminative power among speakers, this signal seems to be redundant with LPCC or melceps, and it is not useful.

If instead of using the residual signal of a linear analysis a nonlinear analysis is used, both combined signals are more uncorrelated and although the discriminative power of the NL residual signal is lower, the combined scheme outperforms the linear one for several analysis orders.

The results show that there is just a marginal improvement on the results when increasing the number of parameters (the identification rate plot saturates), but the residual signal is whiter when increasing the prediction order, especially for the nonlinear analysis. This is a promising result, because although a good parameterization based on nonlinear analysis has not yet been established, this paper reveals that the NL analysis can extract more relevant information with the same prediction order as a linear analysis. Thus, it opens a new way for investigation that has started to provide successful results [2].

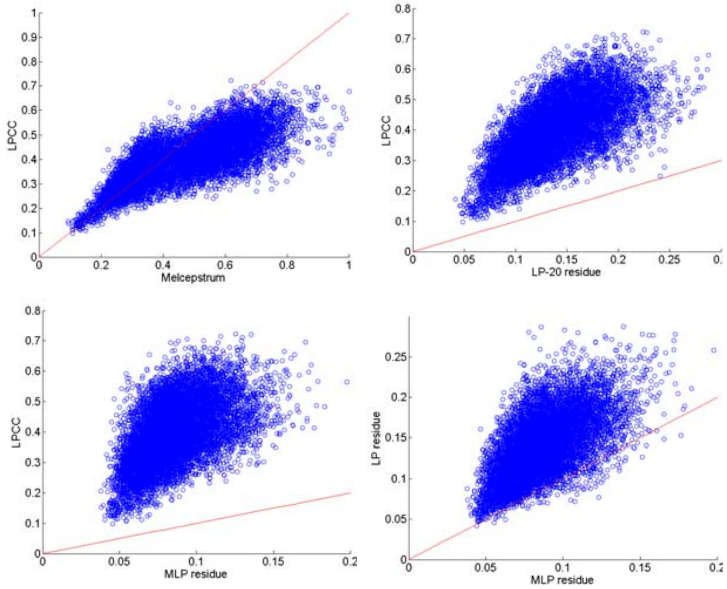


Figure 4: Scatter diagram of distances for observing the correlation between parameters.

6. REFERENCES

[1] F. Bimbot, L. Mathan "Text-free speaker recognition using an arithmetic-harmonic sphericity measure." pp.169-172, Eurospeech 1993.

[2] M. Chetouani, M. Faundez, B. Gas, J. L. Zarader "A New Nonlinear speaker parameterization algorithm for speaker identification". ISCA Speaker Odyssey Workshop·, 2004.

[3] J. Deller et al. "Discrete-Time Processing of Speech Signals," Prentice-Hall, 1993.

[4] M. Faundez and D. Rodriguez "Speaker recognition using residual signal of linear and nonlinear prediction models". Vol.2 pp.121-124. ICSLP'98, Sidney.

[5] M. Faundez, F. Vallverdú, E. Monte, "Nonlinear prediction with neural nets in adpcm" IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1998, Vol I, pp.345-348.Seattle.

[6] M. Faundez "Speaker recognition by means of a combination of linear and nonlinear predictive models" EUROSPEECH'99, Budapest. Vol. 2 pp. 763-766.

[7] M. Faundez et al.. "nonlinear speech processing: overview and applications". Control and intelligent systems, Vol. 30 Nº 1, pp.1-10, 2002, ACTA Press

[8] M. Faundez "Data fusion in Biometrics". IEEE Aerosp. Electron. Syst. Mag. Vol.20 nº 1, pp.34-38, January 2005

[9] T. C. Feustel & G. A. Velius "Human and machine performance on speaker identity verification". Speech Tech 1989, pp.169-170.

[10] F. D. Foresee and M. T. Hagan, "Gauss-Newton approximation to Bayesian regulariza-tion", proceedings of the 1997 International Joint Conference on Neural Networks, pp.1930-1935, 1997

[11] S. Hayakawa, K. Takeda & F. Itakura "Speaker identification using harmonic structure of LP-Residual spectrum" Audio Video Biometric personal autentification 1997, pp. 253-260 LNCS-1206.

[12] J. He, L. Liu & G. Palm "On the use of features from prediction residual signals in speaker identification". EUROSPEECH'1995 pp.313-316.

[13] J. Kittler, M. Hatef, R. P. W. Duin & J. Matas "On combining classifiers". IEEE Trans. On PAMI, Vol. 20, Nº 3, pp. 226-239, 1998

[14] L. Liu et al. "Signal modelling for speaker identification". Proceedings of the IEEE ICASSP 1996, Vol.2, pp. 665 - 668

[15] R. J. Mammone, X. Zhang & R. Ramachandran "Robust speaker recognition" IEEE Sig. Proc. magazine, 1996, pp.58-70.

[16] J. Ortega et al. "Ahumada: a large speech corpus in Spanish for speaker identification and verification". ICASSP 1998 Seattle, Vol. 2, pp. 773-776.

[17] J. W. Picone "Signal Modeling techniques in speech recognition" Proceedings of the IEEE, Vol. 79, Nº 4, April 1991, pp.1215-1247

[18] C. Sanderson "Information fusion and person verification using speech & face information". IDIAP Research Report 02-33, pp. 1-37. September 2002

[19] R. Schlüter, H. Ney "Using phase spectrum information for improved speech recognition performance" Proceedings of the IEEE ICASSP 2001, Vol.1, pp.133-136

[20] P. Thévenaz, H. Hügli "Usefulness of the LPC-residue in text-independent speaker verification" Speech Communication 17 (1995) pp. 145-157

[21] http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

[22] B. Yegnanarayana et al. "Source and system features for speaker recognition using AANN models" IEEE ICASSP 2001, Vol.1, pp.409-412

Multi filter bank approach for speaker verification based on genetic algorithm

Christophe Charbuillet, Bruno Gas, Mohamed Chetouani, Jean Luc Zarader

Université Pierre et Marie Curie-Paris6, FRE2507
Institut des Systèmes Intelligents et Robotique (ISIR), Ivry sur Seine, F-94200 France
Christophe.Charbuillet@lis.jussieu.fr, Gas@ccr.jussieu.fr,
Mohamed.Chetouani@upmc.fr, jean-luc.zarader@upmc.fr

Abstract

Speech recognition systems usually need a feature extraction stage which aims at obtaining the best signal representation. State of the art speaker verification systems are based on cepstral features like MFCC, LFCC or LPCC. In this article, we propose a feature extraction system based on the combination of three feature extractors adapted to the speaker verification task. A genetic algorithm is used to optimize the features complementarities. This optimization consists in designing a set of three non linear scaled filter banks. Experiments are carried out using a state of the art speaker verification system. Results show that the proposed method improves significantly the system performances on the 2005 Nist SRE Database. Furthermore, the obtained feature extractors show the importance of some specific spectral information for speaker verification.

1. Introduction

Speech feature extraction plays a major role in speaker verification systems. State of the art speaker verification systems front end are based on the estimation of the spectral envelope of the short term signal, e.g., Mel-scale Filterbank Cepstrum Coefficients (MFCCs), Linear-scale Filterbank Cepstrum Coefficients (LFCCs), or Linear Predictive Cepstrum Coefficients (LPCCs). Even if these extraction methods achieve good performances on speaker verification, they do not take into account specific information about the task to achieve. To avoid this draw back, several approaches have been proposed to optimize the feature extractor to a specific task. These methods consist to simultaneously learn the parameters of both the feature extractor and the classifier [1]. This procedure consists in the optimization of a criterion, which can be the Maximization of the Mutual Information (MMI) [2] or the Minimization of the Classification Error (MCE) [3]. In this paper we proposed to used a genetic algorithm to design a feature extraction system adapted to the speaker verification task.

Genetic algorithms (GA) were first proposed by Holland in 1975 [4] and became widely used in various disciplines as a new means of complex systems optimization. In recent years their have been successfully applied to the speech processing domain. Chin-Teng Lin and al. [5] proposed to apply a GA to the feature transformation problem for speech recognition and M. Zamalloa and al. [6] worked on a GA based feature selection for speaker recognition. GAs most attractive quality is certainly their aptitude to avoid local minima. However, our study relies on another quality which is the fact that GAs are unsupervised optimization methods. So they can be used as an exploration tool, free to find the best solution without any constraint. In a previous work [7] we used this approach to show the importance of specific spectral information for the speaker diarization task. State of the art speaker verification systems are based on a cepstral feature extraction front end (LFCC, MFCC, LPCC) follow by a GMM [8] or an hybrid GMM/SVM classifier [9]. Nowadays, an alternative an increasingly used approach consists in fusing different systems. This technique can be divided in two main categories depending on the source of this difference. The systems based on a classifier's variety [10] and the systems based on different features. Our study deals with the second principle. We can quote the work of M. Zhiyou and al. [11] which consist of combining the LFCC and MFCC features, or the study of Poh Hoon Thian & al. [12] who proposed to complete the LFCC's with spectral centroids subands features.

In this paper we proposed to fuse three systems based on different feature extractors. A genetic algorithm is used to optimize the feature extractor's complementarities. Figure 1 describes this approach. In the second section, a description of the feature extraction method is given. Afterward, we describe the genetic algorithm we used, followed by its application to complementary feature extraction. Then, the experiments we made and the obtained results are presented.

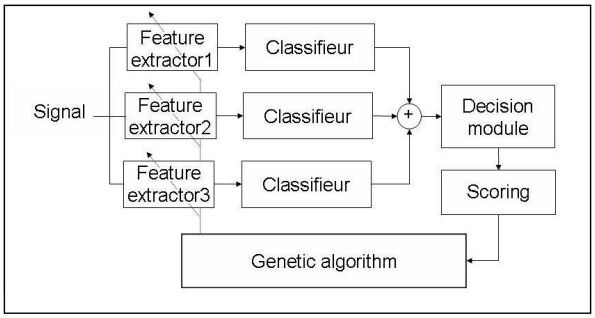


Figure 1: *Feature extraction optimization*

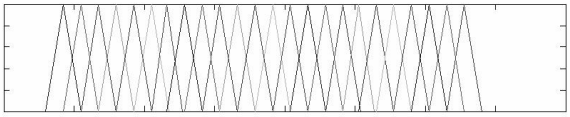


Figure 2: *Linear scaled filter bank*

2. Filter bank based feature extractors

The conventional MFCC and LFCC feature extractor process mainly consists of modifying the short-term spectrum by a filter bank. This process has four steps:

- Compute the power spectrum of the analyzed frame;
- Sum the power spectrum for each triangular filter of the bank;
- Apply the log operator to the obtained coefficients;
- Compute the Discrete Cosine Transform (DCT).

Figure 2 presents the linear scaled filter bank used for the LFCC's computation. This feature extractor is known to be the most robust for the short band signals representation. The purpose of our study is to find a set of three cepstrum based feature extractors design for high level fusion. To this end, we propose to use a genetic algorithm to optimize, the number of filters on the bank, the scaled of the filter bank and the number of cepstral output coefficients.

3. Genetic algorithm

A genetic algorithm is an optimization method. Its aim is to find the best values of the system's parameters in order to maximize its performance. The basic idea is that of "natural selection", i.e. the principle of "the survival of the fittest". A GA operates on a population of systems. In our application, each individual of the population is a feature extractor defined by its genes. Genes consists in a condensed an adapted representation of the feature extractor's operational parameters.

3.1. Gene encoding

Parameter's encoding plays a major role in a genetic algorithm. By an adapted parameter representation, this method can strongly increases the speed convergence of the algorithm. Moreover it reduces the over fitting effect by reducing the parameters dimention. The parameters we chose to optimize are:

- Nf: Number of filters in the bank;
- Nc: Number of cepstral coefficients;
- C_i: Center frequency of the i^{th} filter in the bank;
- B_i: Band width of the i^{th} filter in the bank.

Parameters C and B are encoded with two polynomial functions described by the equations (1) and (2). This encoding method reduces the parameter's dimention from 50 to 12 (in the average case) and guaranties the filter bank's regularity. The parameter Nf and Nc are not encoded and will be directly muted.

$$C_i = gc_0 + gc_1 \cdot \frac{i}{Nf} + gc_2 \cdot (\frac{i}{Nf})^2 + ... + gc_N \cdot (\frac{i}{Nf})^N \quad (1)$$

$$B_i = gb_0 + gb_1 \cdot \frac{i}{Nf} + gb_2 \cdot (\frac{i}{Nf})^2 + ... + gb_N \cdot (\frac{i}{Nf})^N \quad (2)$$

Where $\{gc_0,...,gc_N\}$ and $\{gb_0,...,gb_N\}$ are the genes relative to the parameters $\{C_0,...,C_{Nf}\}$ and $\{B_0,...,B_{Nf}\}$; N is the polynomial order; Nf represent the number of filter.

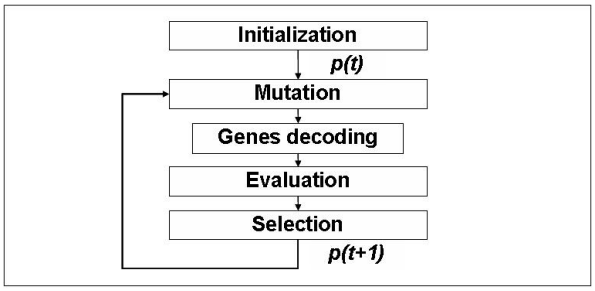


Figure 3: *Genetic algorithm*

3.2. Genetic algorithm description

The algorithm we used is made of four operators: **M**utation, **D**ecoding, **E**valuation and **S**election (M, D, E, S). These operators are applied to the current population $p(t)$ to produce a new generation $p(t+1)$ by the relation:

$$p(t+1) = S \circ E \circ D \circ M(p(t)) \quad (3)$$

Figure 3 represent this algorithm. The first step consists on a random initialization of the feature extractor's genes. Then, the operators are iteratively applied.
The *Mutation* operator consists in a short random variation of the genes.
The *Decoding* operator aim at decode the genes to obtain the operational feature extractor's parameters.
The *Evaluation* operator's goal is to evaluate each feature extractor performances. The evaluation criterion we used is defined on the section 3.3.
The *Selection* operator selects the Ns better feature extractors of the current population. These individuals are then cloned according to the evaluation results to produce the new generation $p(t+1)$ of Np feature extractors. As a consequence of this selection process, the average of the performance of the population tends to increase and in our application adapted feature extractors tend to emerge.

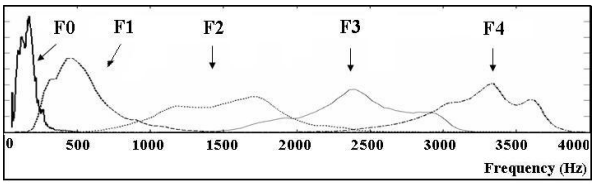
3.3. Application to complementary feature extraction

The objective is to obtain a set of three complementary feature extractors. The main idea is to evolve three isolated populations of feature extractors and to select the best combination. At each generation, the fusion is done for all combination of feature extractors and the resulting Equal Error Rate (EER) is memorized. At the end of this process, the fitness of an individual is defined as the lower EER obtained (e.i. the EER corresponding to the best combination including this feature extractor). As a consequence of this process, each population tends to specialize on specific feature, complementary with the others.

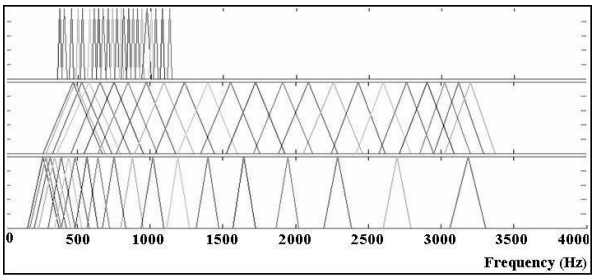
4. Experiments and results

4.1. Data bases

The databases used for the evolution phase and for the test are extracted from the 2005 Nist SRE corpus [13]. This corpus is composed of conversational telephone speech signals passed through different channels, (landline, cordless or cellular) and sampled to 8 kHz. We used 10 male and 10 female with one utterance of 2 min 30s per speaker for the evolution phase. The number of tests between model and test signal involved for each



(a) Formant and fondamental frequency distributions



(b) Obtained filter banks for C1 (top) C2 (midle) and C3 (bottom)

Figure 4: Spectral analyse and obtained solutions

feature extractor evaluation was of 2052. For the test database, we used 50 males and 50 females whose not appear on the training base. The number of tests involved was of 116942.

4.2. Speaker verification system

All experiments we made are based on a state of the art GMM-UBM speaker verification system. This system, called LIA SpkDet [14] was provided by the University of Avignon, France. We used a system with 16 gaussian per mixture, with diagonal covariance matrix.

4.3. Genetic algorithm parameters

The genes $\{gc_0,...,gc_N\}$ and $\{gb_0,...,gb_N\}$ which code for the centres frequencies and the band widths are initialized with a Gaussian normalized random. The parameter Nf are initialized to 24, and Nc to 16.
The parameter we used for the feature extractor's evolution are:

- Population size Np : 20;
- Number of selected individuals Ns : 5;
- Polynomial order for the genes encoding N : 5;
- Mutation method for the polynomials coefficients: Gaussian random variation of $\pm$ 0.1;
- Mutation method for Nf : uniform random variation of $\pm$ 5;
- Mutation method for Nc : uniform random variation of $\pm$ 3;

4.4. Results

In this section, obtained feature extractors are presented and analysed. Figure 4.b presents the obtained filter banks. In order to interpret the obtained solution, a statistical analysis of the fundamental frequency and formants was done on a database composed of 20 male and 20 females. Figure 4.a presents the probability distributions of these mesures.

Table 1: *Comparative results*

Feature extractor	Nf	Nc	F_{min}	F_{max}	EER%
LFCC	24	16	300	3400	**14.44**
MFCC	24	16	300	3400	**14.88**
C1	23	15	360	1145	22.90
C2	25	20	266	3372	14.79
C3	19	19	156	3309	16.07
C1+C2+C3	- -	- -	- -	- -	**12.69**

Table 2: *Fusion analysis*

Feature extractor	Correlation	EER obtained by fusion
C1+C2	0.51	13.21%
C2 + C3	0.83	13.45%
C1 + C3	0.64	15.39%

Table 1 details both the feature extractor's characteristics and the results obtained on the test base. The combination method used is an arithmetic fusion, as illustrated by the figure 1.

Table 2 presents the correlation coefficients between the compared system and the EER obtained by fusion. The correlation is based on the log-likelihood outputs of the compared systems for the whole tests of the test database. A test consists to measure the log-likehood between a speaker model and test signal. The r correlation coefficient is defined by:

$$r = \frac{\sum_{i=1}^{Nt}(S1_i - \bar{S1}) \cdot (S2_i - \bar{S2})}{\sqrt{\sum_{i=1}^{Nt}(S1_i - \bar{S1})^2} \cdot \sqrt{\sum_{i=1}^{Nt}(S2_i - \bar{S2})^2}} \quad (4)$$

Where $S1_i$ represent the log-likelihood obtained by the system 1 on i^{th} test; Nt is the number of test.

The correlation coefficient, which takes value in [-1;1], is a measure of the system's decision similarity. In our application, the classifiers are identical. As a consequence, this measure can be interpreted as the similarity between the information provied by the feature extractors. A correlation of 1 means that the information supplied by the feature extractors are equivalent (i.e. they lead to the same decision). A correlation of 0 means that the information supplied are independent.

Taking into account these different information, we can notice that:

- Information relative to the fundamental frequencies is not used;
- C2 covers a large spectral zone and obtained results similar are to the LFCC or MFCC feature extractors;
- C1 seems to focus exclusively on the first formant;
- C3 presents a high filter density centred on the first formant, while keeping the whole spectre information;
- The de-correlation of the obtained systems are significant.
- The final combination of the three feature extractors improve the system performance of 12% compare to the baseline system.

These results show that the proposed method is reliable. The correlation between the diferent systems and the improvement supplied by the fusion show that the obtained feature extractors are complementary. This improvement seems to be related to the information provied by the first formant. In the final article, a more detailed analyse of these results will be presented.

5. Conclusion

In this paper, we proposed to use a genetic algorithm to optimize a feature extraction system adapted to the speaker verification task. The proposed system is based on a combination of three complementary feature extractors. Obtained results show that the proposed method improves significantly the system performance. Furthermore, the obtained feature extractors reveal the importance of specific spectral information relatives to the first formant.
Our future work will consist in study the robustness of the obtained solutions according to both the initial conditions and the base used for the evolution phase.

6. References

[1] B. G. Mohamed Chetouani, Marcos Faundez-Zanuy and J.-L. Zarader, *Nonlinear Speech Modeling and Applications*. Springer, 2005, ch. Non-linear Speech Feature Extraction for Phoneme Classification and Speaker Recognition, pp. 344–350.

[2] K. Torkkola, "Feature extraction by non parametric mutual information maximization," *The Journal of Machine Learning Research*, vol. 3, pp. 1415–1438, 2003.

[3] C. Miyajima, H. Watanabe, K. Tokuda, T. Kitamura, and S. Katagiri, "A new approach to designing a feature extractor in speaker identification based on discriminative feature extraction," *Speech Communication*, vol. 35, no. 3-4, pp. 203–218, Oct. 2001.

[4] J. H. Holland, "Adaptation in natural and artificial systems," *University of Michigan Press*, 1975.

[5] L. Chin-Teng, N. Hsi-Wen, and H. Jiing-Yuan, "Ga-based noisy speech recognition using two-dimensional cepstrum," in *IEEE Transactions on Speech and Audio Processing*, vol. 8, 2000, pp. 664–675.

[6] M. Zamalloa, G. Bordel, J. L. Rodriguez, and M. Penagarikano, "Feature selection based on genetic algorithms for speaker recognition," in *IEEE Odyssey*, vol. 1, 2006, pp. 1–8.

[7] C. Charbuillet, B. Gas, M. Chetouani, and J. L. Zarader, "Filter bank design for speaker diarization based on genetic algorithms," in *Acoustics, Speech, and Signal Processing, 2006. Proceedings. (ICASSP '06). IEEE International Conference on*, vol. 1, 2006, pp. 673–676.

[8] D. Reynolds and R. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 1, pp. 72–83, 1995.

[9] S. Fine, J. Navratil, and R. Gopinath, "A hybrid gmm/svm approach to speaker identification," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 1, 2001, pp. 417–420 vol.1.

[10] K. Farrell, R. Ramachandran, and R. Mammone, "An analysis of data fusion methods for speaker verification," in *Acoustics, Speech, and Signal Processing, 1998. ICASSP '98. Proceedings of the 1998 IEEE International Conference on*, vol. 2, 1998, pp. 1129–1132 vol.2.

[11] M. Zhiyou, Y. Yingchun, and W. Zhaohui, "Further feature extraction for speaker recognition," *IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, pp. 4153–4158, 2003.

[12] N. Poh Hoon Thian, C. Sanderson, S. Bengio, D. Zhang, and K. Jain Anil, "Spectral subband centroids as complementary features for speaker authentication," *Lect. notes comput. sci.*, vol. 3072, pp. 631–639, 2004.

[13] "2005 nist speaker recognition evaluation site." [Online]. Available: http://www.nist.gov/speech/tests/spk/2005/

[14] "Lia spkdet web site." [Online]. Available: http://www.lia.univ-avignon.fr/heberges/ALIZE/LIA RAL

Speaker Recognition Via Nonlinear Discriminant Features

Lara Stoll[1,2], *Joe Frankel*[1,3], *Nikki Mirghafori*[1]

[1]International Computer Science Institute, Berkeley, CA, USA
[2]University of California at Berkeley, USA
[3]Centre for Speech Technology Research, Edinburgh, UK
{lstoll,nikki}@icsi.berkeley.edu, joe@cstr.ed.ac.uk

Abstract

We use a multi-layer perceptron (MLP) to transform cepstral features into features better suited for speaker recognition. Two types of MLP output targets are considered: phones (Tandem/HATS-MLP) and speakers (Speaker-MLP). In the former case, output activations are used as features in a GMM speaker recognition system, while for the latter, hidden activations are used as features in an SVM system. Using a smaller set of MLP training speakers, chosen through clustering, yields system performance similar to that of a Speaker-MLP trained with many more speakers. For the NIST Speaker Recognition Evaluation 2004, both Tandem/HATS-GMM and Speaker-SVM systems improve upon a basic GMM baseline, but are unable to contribute in a score-level combination with a state-of-the-art GMM system. It may be that the application of normalizations and channel compensation techniques to the current state-of-the-art GMM has reduced channel mismatch errors to the point that contributions of the MLP systems are no longer additive.

1. Introduction

The speaker recognition task is that of deciding whether or not a (previously unseen) test utterance belongs to a given target speaker, for whom there is only a limited amount of training data available. The traditionally successful approach to speaker recognition uses low-level cepstral features extracted from speech in a Gaussian mixture model (GMM) system. Although cepstral features have proven to be the most successful choice of low-level features for speech processing, discriminatively trained features may be better suited to the speaker recognition problem. We utilize multi-layer perceptrons (MLPs), which are trained to distinguish between either phones or speakers, as a means of performing a feature transformation of cepstral features.

There are two types of previous work that are directly related to our research, both involving the development of discriminative features. In the phonetically discriminative case, the use of features generated by one or more MLPs trained to distinguish between phones has been shown to improve performance for automatic speech recognition (ASR). At ICSI, Zhu and Chen, et al. developed what they termed Tandem/HATS-MLP features, which incorporate longer term temporal information through the use of MLPs whose outputs are phone posteriors [1, 2].

In the area of speaker recognition, Heck and Konig, et al. focused on extracting speaker discriminative features from MFCCs using an MLP [3, 4]. They used the outputs from the middle layer of a 5-layer MLP, which was trained to discriminate between speakers, as features in a GMM speaker recognition system. The MLP features, when combined on the score-level with a cepstral GMM system, yielded consistent improvement when the training data and testing data were collected from mismatched telephone handsets [3]. A similar approach was followed by Morris and Wu, et al.[5]. They found that speaker identification performance improved as more speakers were used to train the MLP, up to a certain limit [6].

In the phonetic space, we use the Tandem/HATS-MLP features in a GMM speaker recognition system. The idea is that we can use the phonetic information of a speaker in order to distinguish that speaker from others.

In the speaker space, we train 3-layer Speaker-MLPs of varying sizes to discriminate between a set of speakers, and then use the hidden activations as features for a support vector machine (SVM) speaker recognition system. The intuition behind this method is that the hidden activations from the Speaker-MLP represent a nonlinear mapping of the input cepstral features into a general set of speaker patterns. Our Speaker-MLPs are on a larger scale than any previous work: we use more training speakers, training data, and input frames of cepstral features, and larger networks.

To begin, Section 2 outlines the experimental setup. The results of our experiments are reported in Section 3. Finally, we end with discussion and conclusions in Section 4.

2. Experiments

2.1. Overall Setup

The basic setups of the Tandem/HATS-GMM and Speaker-SVM systems are shown in Figures 1 and 2, respectively. Frames of perceptual linear prediction (PLP) coefficients, as well as frames of critical band energies in the former case, are the inputs to the MLPs. A log is applied to either the output or hidden activations, and after either dimensionality reduction or calculation of mean, standard deviation, histograms, and percentiles, the final features are used in a speaker recognition system (GMM or SVM).

2.2. Baseline GMM Systems

We make use of two types of GMM baselines for purposes of comparison. The first is a state-of-the-art GMM system, which was developed by our colleagues at SRI, and which we will refer to as SRI-GMM [7]. It utilizes 2048 Gaussians, CMS, T-norm, H-norm, and channel mapping to improve its results. We use this system for score-level combinations, in which the scores from SRI's GMM system are combined with the scores from our MLP features systems. For more details, see Section 2.6.

The second system, on the other hand, is a very basic GMM system, with 256 Gaussians, and which includes only CMS, without any other normalizations. This system, which we will

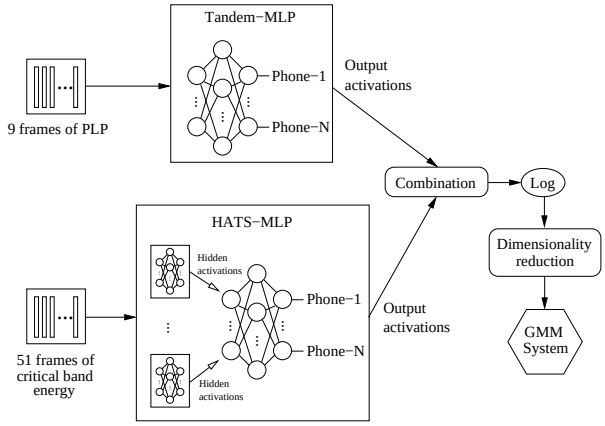


Figure 1: *Tandem/HATS-GMM System*

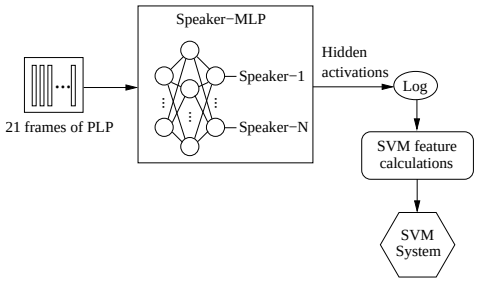


Figure 2: *Speaker-SVM System*

refer to as Basic-GMM, is useful for the purpose of feature-level combination (where we use MFCC features augmented with MLP features as features in the GMM system), as well as for score-level combination.

2.3. Tandem/HATS-MLP Features

There are two components to the Tandem/HATS-MLP features, namely the Tandem-MLP and the HATS-MLP. The Tandem-MLP is a single 3-layer MLP, which takes as input 9 frames of PLPs (12th order plus energy) with deltas and double-deltas, contains 20,800 units in its hidden layer, and has 46 outputs, corresponding to phone posteriors. The hidden layer applies the sigmoid function, while the output uses softmax.

The HATS-MLP is actually two stages of MLPs that perform phonetic classification with long-term (500-1000 ms) information. The first stage MLPs take as input 51 frames of log critical band energies (LCBE), with one MLP for each of the 15 critical bands; each MLP has 60 hidden units (with sigmoid applied), and the output layer has 46 units (with softmax) corresponding to phones. For the HATS (Hidden Activation TRAPS) features, the hidden layer outputs are taken from each first-stage critical band MLP, and then input to the second-stage merger MLP, which contains 750 hidden units, and 46 output units.

The Tandem-MLP and HATS-MLP features are then combined using a weighted sum, where the weights are a normalized version of inverse entropy. The log is applied to the output, and a Karhunen-Loeve Transform (KLT) dimensionality reduction is applied to reduce the output feature vector to an experimentally determined optimal length of 25. This process is illustrated in Figure 1.

The Tandem/HATS-MLP system is trained on roughly 1800 hours of conversational speech from the Fisher [8] and Switchboard [9] corpora.

2.4. Speaker-MLP Features

2.4.1. Speaker Target Selection Through Clustering

As a contrast to using all speakers with enough training data available (with the idea that including more training speakers will yield better results), we also implemented MLPs trained using only subsets of specifically chosen speakers. These speakers were chosen through clustering in the following way. First, a background GMM model was trained using 286 speakers from the Fisher corpus. Then, a GMM was adapted from the background model with the data from each MLP-training speaker. These GMMs used 32 Gaussians, with input features of 12th order MFCCs plus energy and their first order derivatives. The length-26 mean vectors of each Gaussian were concatenated to form a length-832 feature vector for each speaker. Principal component analysis was performed, keeping the top 16 dimensions of each feature vector (accounting for 68% of the total variance). In this reduced-dimensionality speaker space, k-means clustering was done, using the Euclidean distance between speakers, for $k = 64$ and $k = 128$. Finally, the sets of 64 and 128 speakers were chosen by selecting the speaker closest to each of the (64 or 128) cluster centroids.

2.4.2. MLP Training

A set of 64, 128, or 836 speakers was used to train each Speaker-MLP, with 6 conversation sides per speaker used for training, and 2 for cross-validation (CV). The training speaker data came from the Switchboard-2 corpus [9]. The set of 836 speakers included all speakers in the Switchboard2 corpus with at least 8 conversations available. The smaller sets of speakers, selected through clustering, used training and CV data that was balanced in terms of handsets.

ICSI's QuickNet MLP training tool [10] was used to train the Speaker-MLPs. The input to each Speaker-MLP is 21 frames of PLPs (12th order plus energy) with first and second order derivatives appended. The hidden layer applies a sigmoid, while the output uses softmax.

Table 2 shows the sizes of MLPs (varying in the number of hidden units) trained for each set of speakers.

2.5. SVM Speaker Recognition System

The GMM system is well suited to modeling features with fewer than 100 dimensions. However, problems of data sparsity and singular covariance matrices soon arise in trying to estimate high dimensional Gaussians. Previous work in speech recognition (HATS) has shown that there is a great deal of information in the hidden structure of the MLP. Preliminary experiments also showed that reducing the dimensionality of the hidden activations using principal component analysis (PCA) or linear discriminant analysis (LDA), so that the features could be used in a GMM system, yielded poor results. In order to take advantage of the speaker discriminative information in the hidden activations of the Speaker-MLPs, we use an SVM speaker recognition system, which is better suited to handle the high dimensional sparse features, is naturally discriminative in the way it is posed, and has proven useful in other approaches to speaker verification.

Since the SVM speaker recognition system requires the same length feature vector for each speaker (whether a target, an impostor, or a test speaker), we produce a set of statistics

to summarize the information along each dimension of the hidden activations. These statistics (mean, standard deviation, histograms of varying numbers of bins, and percentiles) are then used as the SVM features for each speaker. For our experiments, the set of impostor speakers used in the SVM system is a set of 286 speakers from the Fisher corpus designed to be balanced in terms of gender, channel, and other conditions.

2.6. System Combinations Using LNKnet

In order to improve upon the baseline of the SRI-GMM system, we choose to combine our various systems on the score-level with the SRI-GMM, using LNKnet software [11]. We use a neural network with no hidden layer and sigmoid output nonlinearity, which takes two or more sets of likelihood scores as input. We use a round-robin approach and divide our test data into two subsets for development and evaluation.

3. Results

3.1. Testing Database

In order to compare the performance of our systems, we use the database released by NIST for the 2004 Speaker Recognition Evaluation (SRE) [12]. This database consists of conversational speech collected in the Mixer project, and includes various languages and various channel types. We use only telephone data, containing a variety of handsets and microphones.

One conversation side (roughly 2.5 minutes) is used for both the training of each target speaker model and the testing of each test speaker. As performance measures, we use the detection cost function (DCF) of the NIST evaluation and the equal error rate (EER). The DCF is defined to be a weighted sum of the miss and false alarm error probabilities, while the EER is the rate at which these error probabilites are equal.

3.2. Tandem/HATS-GMM

For NIST's SRE2004, the DCF and EER results are given in Table 1 for the Basic-GMM system, the Tandem/HATS-GMM system, and their score- and feature-level combinations, as well as for the SRI-GMM system and its combination with the Tandem/HATS-GMM. Changes relative to each baseline (where a positive value indicates improvement) are shown in parentheses.

Alone, the Tandem/HATS-GMM system performs slightly better than the Basic-GMM system. Feature-level combination of MFCC and Tandem/HATS features in a GMM system, as well as score-level combination of the Tandem/HATS-GMM system with the Basic-GMM, both yield significant improvements. When the Tandem/HATS-GMM system is combined on the score-level with the SRI-GMM system, there is no gain in performance over the SRI-GMM alone.

	DCF×10	EER (%)
Basic-GMM	0.724	18.48
Tandem/HATS-GMM	0.713 (2%)	18.48 (0%)
Score-level fusion	0.618 (15%)	16.26 (12%)
Feature-level fusion	0.601 (17%)	16.35 (12%)
SRI-GMM	0.374	9.01
Tandem-GMM	0.713	18.48
Score-level fusion	0.378 (-1%)	9.09 (-1%)

Table 1: *Tandem/HATS-GMM system improves upon Basic-GMM system, especially in combination, but there is no improvement for SRI-GMM system.*

3.3. Speaker-SVM

Both the cross-validation and SRE2004 results for the Speaker-MLPs are shown in Table 2 for each size MLP. It is clear that the CV accuracy increases with respect to the number of hidden units, for each training speaker set. The accuracy increase on adding further hidden units does not appear to have reached a plateau at 2500 hidden units for the 836 speaker net, though for the purposes of the current study the training times became prohibitive. With the computation shared between 4 CPUs, it took over 4 weeks to train the MLP with 2500 hidden units.

Similar to the CV accuracy, the speaker recognition results improve with an increase in the size of the hidden layer when considering a given number of training speakers.

# spkrs	Hid. units	CV acc.	DCF×10	EER (%)
64	400	37.8%	0.753	21.04
64	1000	47.8%	0.715	20.41
128	1000	39.4%	0.702	20.45
128	2000	44.5%	0.691	19.70
836	400	20.5%	0.756	22.88
836	800	25.5%	0.734	21.37
836	1500	32.0%	0.711	20.45
836	2500	35.5%	0.689	19.91

Table 2: *Speaker-SVM results improve as the number of hidden units, as well as the CV accuracy, increase.*

In Table 3, the results are given for the score-level combination of the 64 speaker, 1000 hidden unit, Speaker-SVM system with the Basic-GMM and SRI-GMM systems. For the SRI-GMM, the best combination is yielded when the Speaker-MLP is trained with 64 speakers and 1000 hidden units (although the 128 speakers with 2000 hidden units does somewhat better in combination with the Basic-GMM). There is a reasonable gain made when combining the Speaker-SVM system with the Basic-GMM, but there is no significant improvement for the combination of the Speaker-SVM and SRI-GMM systems.

	DCF×10	EER (%)
Basic-GMM	0.724	18.48
Speaker-SVM	0.715	20.41
Score-level fusion	0.671 (7%)	17.52 (5%)
SRI-GMM	0.374	9.01
Speaker-SVM	0.715	20.41
Score-level fusion	0.373 (0%)	9.01 (0%)

Table 3: *System combination with 64 speaker, 1000 hidden unit, Speaker-SVM improves Basic-GMM results, but not the SRI-GMM.*

3.4. Mismatched Train and Test Conditions

We now consider matched (same gender and handset) and mismatched (different gender or handset) conditions between the training and test data. Such a breakdown is given in Table 4 for both the Tandem/HATS-GMM and Speaker-SVM systems and their score-level combinations with the Basic-GMM and SRI-GMM. For each combination, changes relative to the appropriate baseline system are given in parentheses.

When considering a score-level fusion with the Basic-GMM system, gains are made in the matched and especially the mismatched conditions for both the Tandem/HATS-GMM and Speaker-SVM. For the SRI-GMM baseline, combination with the Tandem/HATS-GMM and Speaker-SVM systems has marginal impact in either the matched or mismatched case.

	System Alone		Fusion with Basic-GMM		Fusion with SRI-GMM	
	Matched EER (%)	Mismatched EER (%)	Matched EER (%)	Mismatched EER (%)	Matched EER (%)	Mismatched EER (%)
Basic-GMM	9.13	22.65	–	–	–	–
SRI-GMM	5.74	10.71	–	–	–	–
Tandem/HATS-GMM	12.53	21.54	8.67 (5%)	19.84 (12%)	5.74 (0%)	10.77 (-1%)
Speaker-SVM (1000hu, 64ou)	13.93	23.56	8.78 (4%)	20.95 (7%)	5.74 (0%)	10.64 (1%)

Table 4: *Breakdown of results for matched and mismatched conditions for the MLP-based systems and their score-level fusions with the Basic-GMM and SRI-GMM.*

4. Discussion and conclusions

For the first time, phonetic Tandem/HATS-MLP features were tested in a speaker recognition application. Although developed for ASR, the Tandem/HATS-MLP features still yield good results for a speaker recognition task, and in fact perform better than a basic cepstral GMM system; even more improvement comes from score- and feature-level combinations of the two.

Prior related work used discriminative features from MLPs trained to distinguish between speakers. Motivated by having a well-established infrastructure for neural network training at ICSI, we felt that there was potential for making greater gains by using more speakers, more hidden units, and a larger contextual window of cepstral features at the input. Even though preliminary experiments confirmed this, ultimately, however, a smaller subset of speakers chosen through clustering proved similar in performance and could be trained in less time.

Although the MLP-based systems do not improve upon the SRI-GMM baseline in combination, this result could be explained by considering the difference in the performance between the two types of systems: standalone, each MLP-based system performs much more poorly than the SRI-GMM. The addition of channel compensating normalizations, like T-norm [13], to an MLP-based system should help reduce the performance gap between the MLP-based system and the SRI-GMM. It may then be possible for the MLP-based system to improve upon the state-of-the-art cepstral GMM system in combination, in the event that the performance gap is narrowed sufficiently.

Similar to results observed in prior work, the Speaker-SVM system improved speaker recognition performance for a cepstral GMM system lacking sophisticated normalizations (such as feature mapping [14], speaker model synthesis (SMS) [15], and T-norm); such a result was also true for the Tandem/HATS-GMM system. However, no gains were visible in addition with the SRI-GMM, which is significantly improved from the Basic-GMM (as well as the GMM systems of Wu and Morris, et al. and Heck and Konig, et al.) by the addition of feature mapping, T-norm, as well as increasing the number of Gaussians to 2048.

As shown in Table 4, combinations of the Basic-GMM with the phonetic- and speaker-discriminant MLP-based systems of this paper do yield larger improvements for the mismatched condition (which refers to the training data and test data being different genders or different handset types). However, such a result does not hold for combinations of the MLP-based systems with the SRI-GMM. The previous work of Heck and Konig, et al., completed prior to the year 2000, showed that the greatest strength of an MLP-based approach was for the case when there is a handset mismatch between the training and test data, however, the state-of-the-art has since advanced significantly in normalization and channel compensation techniques. As a result, the contributions of the MLP-based systems, without any normalizations applied, to a state-of-the-art cepstral GMM system are no longer significant for the mismatched condition.

5. Acknowledgements

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship and upon work supported by the National Science Foundation under grant number 0329258. This work was also made possible by funding from the EPSRC Grant GR/S21281/01 and the AMI Training Programme. We would also like to thank our colleagues at ICSI and SRI.

6. References

[1] B. Chen, Q. Zhu, and N. Morgan, "Learning long-term temporal features in LVCSR using neural networks," in *ICSLP*, 2004.

[2] Q. Zhu, B. Chen, N. Morgan, and A. Stolcke, "On using MLP features in LVCSR," in *ICSLP*, 2004.

[3] L. P. Heck, Y. Konig, M. K. Sönmez, and M. Weintraub, "Robustness to telephone handset distortion in speaker recognition by discriminative feature design," *Speech Communications*, vol. 31, no. 2-3, pp. 181–192, 2000.

[4] Y. Konig, L. Heck, M. Weintraub, and K. Sönmez, "Nonlinear discriminant feature extraction for robust text-independent speaker recognition," in *Proceedings of RLA2C - Speaker Recognition and Its Commercial and Forensic Applications*, Avignon, France, 1998.

[5] A. C. Morris, D. Wu, and J. Koreman, "MLP trained to separate problem speakers provides improved features for speaker identification," in *IEEE Int. Carnahan Conf. on Security Technology*, 2005.

[6] D. Wu, A. Morris, and J. Koreman, "MLP internal representation as discriminative features for improved speaker recognition," in *Proc. NOLISP*, 2005, pp. 25–33.

[7] S. Kajarekar, L. Ferrer, E. Shriberg, K. Sönmez, A. Stolcke, A. Venkataraman, and J. Zheng, "SRI's 2004 NIST speaker recognition evaluation system," in *ICASSP*, vol. 1, 2005, pp. 173–176.

[8] C. Cieri, D. Miller, and K. Walker, "The Fisher corpus: a resource for the next generations of speech to text," in *LREC*, 2004, pp. 69–71.

[9] Linguistic Data Consortium, "Switchboard-2 corpora," http://www.ldc.upenn.edu.

[10] D. Johnson, "QuickNet3," http://www.icsi.berkeley.edu/Speech/qn.html, 2004.

[11] MIT Lincoln Labs, "LNKNet," http://www.ll.mit.edu/IST/lknet, 2005.

[12] National Institute of Standards and Technology, "The NIST year 2004 speaker recognition evaluation plan," http://www.nist.gov/speech/tests/spk/2004/SRE-04_evalplan-v1a.pdf, 2004.

[13] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," in *Digital Signal Processing*, vol. 10, 2000, pp. 42–54.

[14] D. Reynolds, "Channel robust speaker verification via feature mapping," in *ICASSP*, 2003.

[15] R. Teunen, B. Shahshahani, and L. Heck, "A model-based transformational approach to robust speaker recognition," in *ICSLP*, 2000.

Bispectrum Mel-frequency Cepstrum Coefficients for Robust Speaker Identification

Ufuk Ülüğ[1], Tolga Esat Özkurt[2], Tayfun Akgül[1]

[1]Department of Electronics and Communications Engineering, Istanbul Technical University, Istanbul, Turkey
[2]Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA, USA

ulug@itu.edu.tr, tolga@neuronet.pitt.edu, tayfun.akgul@itu.edu.tr

Abstract

In this paper, we introduce the use of bispectrum slice for mel-frequency cepstrum coefficients as robust text-independent speaker identification. The main advantage of using the bispectrum is to be able to suppress additive Gaussian noise while preserving the phase information of the signal. In order to obtain cepstral coefficients, features of the speech signal are extracted by mel-frequency filter banks, the cosine transform and the logarithm operator. Under various noisy test utterances, we compare and present the performances of the methods which use the bispectrum and the classical mel-frequency cepstrum coefficients.

1. Introduction

Speech signals yield information about the identity of the speaker as well as the content of the speech. Speaker recognition methods have found various applications such as security, voiced internet applications and telephone banking. Such systems consist mainly two parts: Speaker identification and verification. While speaker identification determines the identity of the speaker among a group of people from text dependent or independent speech signal, speaker verification is utilized as a second step to ensure the validity of the resultant speaker obtained by the identification process.

In speaker identification systems, speech signals are recorded and saved into a database. Training sets, which consist of the feature vectors of previously archived speech signals, are compared with the test sets. The conditions for obtaining the training set and the test set can be tremendously different. While the former can usually be obtained in noiseless environments, the test set may not. This may lead to an important decrease in the performance of a speaker identification system. Various methods have been proposed in literature in order to prevent this problem. They mainly include robust feature extraction, speech enhancement techniques and noise compensation [1].

The main advantage of using higher order statistics is to be able to suppress Gaussian noise unlike the classical autocorrelation-based (power spectrum-based) methods. For example, in [2] a particular part of bispectrum is suggested for feature extraction for speaker identification which is shown to be robust to additive Gaussian noise compared to the classical cepstrum.

In this study, we propose to use a bispectrum slice for the computation of mel-frequency cepstrum coefficients as robust features in a text-independent speaker identification system. The organization of the paper is as follows: Section 2 is for a brief explanation of feature extraction, bispectrum and sum-of-cumulants. In Section 3, the speaker identification systems and Gaussian mixture models are summarized. Simulation results are presented in Section 4. Conclusion is given in Section 5.

2. Feature Extraction Using Bispectrum Slice

Feature extraction of a speech signal is mostly based on the spectrum because any information about the characteristics of the vocal track can be obtained from the spectrum [3]. Although the spectrum of a speech signal can be defined by different models, using filter banks instead of linear prediction analysis provides more robust speech features. In this study, we extend to use bispectrum slice instead of the classical spectrum cepstrum coefficients obtained by the mel-frequency filter banks. The brief introduction of the bispectrum slice is given below.

2.1. Bispectrum Slice

If the autotripplecorrelation of any discrete signal $x(n)$ is

$$c(\tau_1, \tau_2) = E[x(n)x(n+\tau_1)x(n+\tau_2)] \quad (1)$$

then the bispectrum $B(\omega_1, \omega_2)$ is defined as the 2-D Fourier transform of its autotripplecorrelation [4]:

$$B(\omega_1,\omega_2) = F\{c(\tau_1,\tau_2)\} \quad (2)$$

where $E[.]$ is the expected value and $F\{.\}$ is Fourier transform. 1-D inverse Fourier transform of the

bispectrum, *q(n),* on the $\omega_1 = \omega_2$ line is defined as the sum of cumulants [4]:

$$q(n) = x(n)*x(n)*k(n) \quad (3)$$

where * denotes convolution operator and

$$k(n) = \begin{cases} x(N-1-N/2) & n \text{ odd} \\ 0 & n \text{ even} \end{cases} \quad (4)$$

for the signal with *n=0,1,…,N-1*. The sequence, *q(n),* has *4N-3* samples with *n=-2(N-1),…,-1,0,1,…,2(N-1).*

2.2. Feature Extraction

Before the analysis, speech signal is divided into segments of 16 ms length and 10 ms overlap. After the estimation of the sum of cumulants using (3) for these frames, Mel-Frequency Cepstrum Coefficients (MFCC) are obtained by utilizing mel-frequency filter banks. Figure 1 shows the block diagram of the steps for the extraction of features.

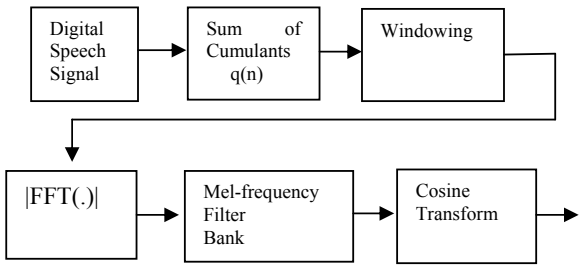


Figure 1: Feature Extraction Block Diagram

3. Speaker Identification System

Gaussian Mixture Model (GMM) is used for speaker identification in this study. In text-independent speaker identification systems, the performance of GMM is known to be relatively high. Let *M* be the order, then GMM is expressed as:

$$p(\mathbf{x} \mid \lambda) = \sum_{i=1}^{M} w_i b_i(\mathbf{x}) \quad (5)$$

where

$$b_i(\mathbf{x}) = \frac{1}{(2\pi)^{D/2} \mid \mathbf{C_i} \mid^{1/2}} \exp(-0.5(\mathbf{x}-\boldsymbol{\mu}_\mathbf{i})^T \mathbf{C_i}^{-1}(\mathbf{x}-\boldsymbol{\mu}_\mathbf{i})) \quad (6)$$

In that expression $\boldsymbol{x}$ is a *D* dimensional random vector, w_i *(i=1,…,M)* are weight coefficients and $b_i(\boldsymbol{x})$ are component density functions. The mixture weight coefficients satisfy the below equation:

$$\sum_{i=1}^{M} w_i = 1 \quad (7)$$

GMM is defined with $\lambda = \{w_i, \boldsymbol{\mu_i}, \boldsymbol{C_i}\}$, where $\boldsymbol{\mu_i}$ is the mean value vectors and $\boldsymbol{C_i}$ is the covariance matrix. For speaker identification systems, λ can be represented to model the speaker. Then the model parameters are estimated by expectation maximization method, which maximizes the model likelihood iteratively [6].

4. Simulation Results

For the training and test sets we use TIMIT database. The training set contains 50 male speech excerpts with the same dialect for various lengths. Test data contains 5 different sentences from different speech segments for each training member. The order of GMM is chosen as *M=40.* The performance of the proposed bispectrum-based mel-frequency cepstrum coefficients are compared with the classical mel-frequency cepstrum coefficients.

The evaluation is done by the normalized total score which is the logarithmic extraction of the actual speaker's likelihood from the maximum likelihood of the other speakers apart from the actual speaker:

$$\log(L(X)) = \log p(X \mid S = S_c, \lambda) - \max(p(X \mid S \neq S_c, \lambda)) \quad (9)$$

Here, *X* is the features in the test set, *S* is the speakers in the training set and S_c is the actual speaker with λ model parameters. If the normalized score is positive, the speaker is estimated correctly.

First, we simulated our speaker identification system with noiseless training sets and noise-free test sets and reached almost %100 correct identification with both spectral and bispectral methods. Then, in order to compare the robustness of bispectrum to spectrum, we add white Gaussian noise with SNR = 40, 20, 15, 10, 5 dBs to the speech signal. Speaker identification performances for both methods are given in Figure 2. Below 10 dB, the performance results of both the spectrum and the bispectrum slice is low but above 10 dB, the speaker identification rate is better when the bispectral method is used.

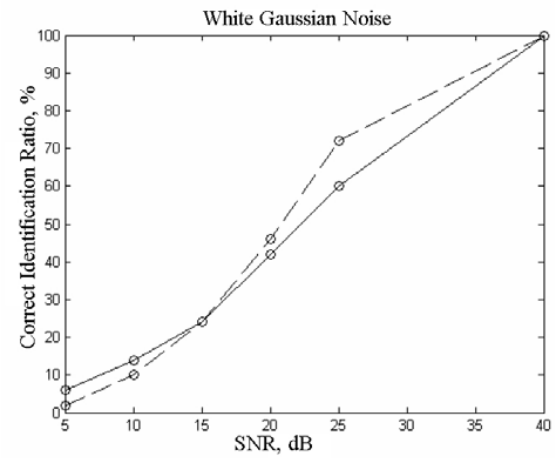


Figure 2: Correct identification ratios vs. SNR for spectral (straight-line) and bispectral methods (dashed-line) are used when white Gaussian noise added.

Various noises are also added to the speech signals in test set to show the effect of the proposed methods. Real noise samples, such as babble, car and factory noises, are collected from NOISEX database of NATO Speech Signal Workgroup [8]. We down sampled that noise samples from 19.98 kHz to 16 kHz and added them to speech signals with SNR = 30, 25, 15, 10, 5, 0 dBs. Although, the distributions of these noises may not be Gaussian, they are symmetric since their skewnesses are closer to zero. The histograms of the noises can be seen in Figure 3. Test results for real noise experiment are provided in Figures 4, 5, 6, which show that, at any SNR above 0 dB, the performance is higher when the bispectrum slice is utilized for feature extraction in our speaker identification system.

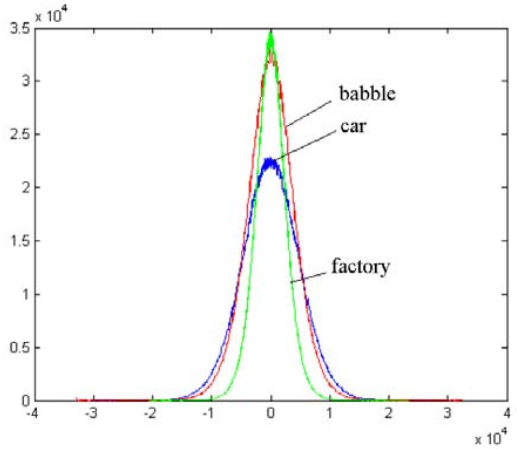


Figure 3: Noise histograms for babble, car and factory

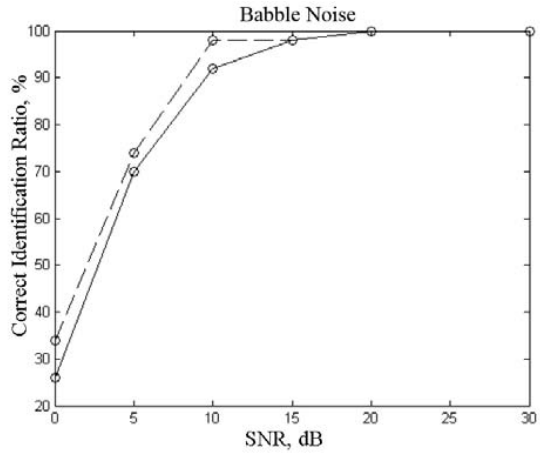


Figure 4: Correct identification ratios vs. SNR for spectral (straight-line) and bispectral methods (dashed-line) are used when babble noise added.

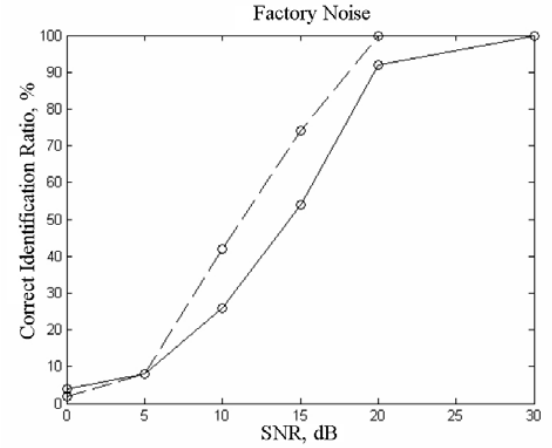


Figure 5: Correct identification ratios vs. SNR for spectral (straight-line) and bispectral methods (dashed-line) are used when factory noise added.

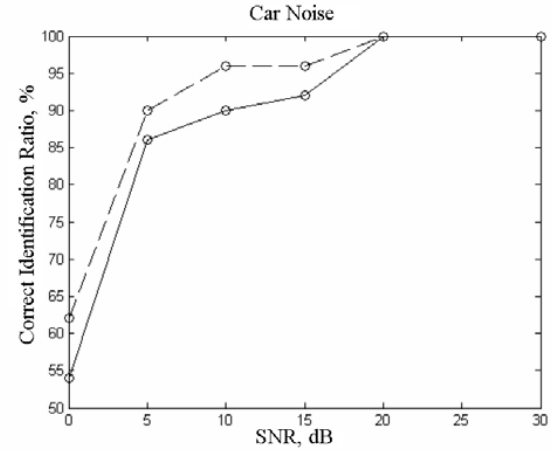


Figure 6: Correct identification ratios vs. SNR for spectral (straight-line) and bispectral methods (dashed-line) are used when car noise added.

The next step in our study is to compare the robustness of spectral and bispectral methods to colored noise produced by wavelet-based methods. We will present the results and the comparisons during the presentation and in the final version of the paper.

5. Conclusion

It is known that additive Gaussian noise deteriorates the identification rate seriously in speaker identification systems. We propose to use bispectrum slice mel-frequency cepstrum features as robust and efficient features for text-independent speaker identification systems. We show the comparisons between the classical and bispectrum based methods.

6. References

[1] **Gong, Y.**, "Speech Recognition In Noisy Environments: A Survey," *Speech Communication*, vol. 16, pp. 261-291, 1995.

[2] **Wenndt, S., Shamsunder, S.,** "Bispectrum Features for Robust Speaker Identification," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1095-1098, 1997.

[3] **Campbell J.,** "Speaker Recognition: A Tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, 1437-1462, 1997.

[4] **Akgül T., El-Jaroudi A.,** "Reconstruction of Mixedphase Signals From Sum-of-Autotripple-correleations Using Least Squares," *IEEE Transactions on SignalProcessing*, vol.46 , no.1, 250-254, 1998.

[5] **Reynolds D.,** "Robust Text-Independent Speaker Identification Using Gaussian Mixture Models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no.1, 4072-4075, 1995.

[6] **Bilmes J.,** "A Gentle Tutorial of the EM Algorithm and Its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," International Computer Science Institute, 1998.

[7] **Furui S.,** *Digital Speech Processing, Synthesis and Recognition*, Marcel Dekker, New York, 2001.

[8] **Varga M. G., Steeneken H. J. M.,** "Assessment for Automatic Speech Recognition: II. NOISEX-92:A Database and an Experiment to Study the Effect of Additive Noise on Speech Recognition Systems," *Comput. Speech Lang.*, vol. 12, pp. 247-251, 1993.

Perceptron-based Class Verification

Michael Gerber, Tobias Kaufmann and Beat Pfister

Speech Processing Group
Computer Engineering and Networks Laboratory
{gerber,kaufmann,pfister}@tik.ee.ethz.ch

Abstract

We present a method to use multilayer perceptrons (MLPs) for a verification task, i.e. to verify whether two vectors are from the same class or not. In tests with synthetic data we could show that the verification MLPs are almost optimal from a Bayesian point of view. With speech data we have shown that verification MLPs generalize well such that they can be deployed as well for classes which were not seen during the training.

1. Introduction

Multilayer perceptrons (MLPs) are successfully used in speech processing. For example they are used to calculate the phoneme posterior probabilities in hybrid MLP/HMM speech recognizers (see for example [1]). In this case their task is to output for every phoneme the posterior probability that a given input feature vector is from this phoneme. They are thus used to *identify* a feature vector with a given phoneme. Expressed in more general terms the MLPs are used for the *identification* of input vectors with a class from within a closed set of classes.

There are applications however, where the identification of input vectors is not necessary but it has to be *verified* whether two given input vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are from the same class or not. In Section 2 we present two verification tasks in the domain of speech processing. In this work we show that MLPs have the capability to optimally solve verification problems. Furthermore we have observed in a task with real-world data that the verification MLPs can even be used to discriminate between classes which were not present in the training set. This is an especially useful property for two reasons:

- The verification MLP is usable for an open set of classes.
- Since we do not need training data from the classes present in the application but can collect training data from other classes which have the same classification objective (e.g. classifying speakers). Therefore we can build a training set of a virtually unlimited size.

In Section 2 we present the motivation for our approach to class verification and outline how MLPs can be used for that purpose. The structure and training of our verification MLPs is described in Section 3. Our evaluation methods are described in Section 4. In order to test whether verification MLPs are capable of performing the verification task in an optimal way from a Bayesian point of view we made experiments with synthetic data. These experiments and their results are described in Section 5. The results of experiments with speech data are shown in Section 6. Finally, our conclusions are summarized in Section 7.

2. Motivation

Our method to decide whether two speech signals are spoken by the same speaker or not includes the following 3 steps: First equally worded segments are sought in the two speech signals. This results in a series of frame pairs where both frames of a pair are from the same phoneme. In a second step for each frame pair the probability that the two frames come from the same speaker is computed. Finally, the global indicator that the two speech signals were spoken by the same speaker can be calculated from these frame-level probabilities. See e.g. [2] for a more detailed description of the speaker-verification approach. We used the verification MLPs for the following two tasks:

- In order to seek phonetically matching segments in two speech signals with a method based on dynamic programming we need a phonetic probability matrix. This matrix is spanned by the two signals and every element $P_{ij}(x_{1i}, x_{2j})$ gives the probability that frame i of signal 1 given as feature vector x_{1i} and frame j of signal 2 given as feature vector x_{2j} are from the same phoneme. The probabilities $P_{ij}(x_{1i}, x_{2j})$ are calculated by an appropriately trained verification MLP.
- For every frame pair we use a verification MLP to calculate a score which stands for the probability that the two phonetically matching frames are from the same speaker. In this case we use a MLP which was trained with data from speakers which are not present in the test. Therefore we make use of the generalization property of the verification MLP.

3. Verification MLP

Since the MLP has to decide whether two given input vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are from the same class the MLP has to process vector pairs rather than single vectors. The target output of the MLP is o_s if the two vectors of the pair are from the same class and o_d if they are from different classes. The vectors are decided to belong to the same class if the output is closer to o_s and to different classes otherwise.

The sizes of the 3-layer perceptrons used for the experiments described in Sections 5 and 6 are as follows:

dataset	input size	1st hidden layer	2nd hidden layer	output layer
synthetic data	$2 \cdot 2...5$	20 (tanh)	10 (tanh)	1 (tanh)
phoneme verification	$2 \cdot 26$	80 (tanh)	35 (tanh)	1 (tanh)
speaker verification	$2 \cdot 16$	70 (tanh)	18 (tanh)	1 (tanh)

The verification MLPs are trained by means of the backpropagation algorithm. For a hyperbolic tangent output neuron a good choice for the output targets is $o_s = 0.75$ and $o_d = -0.75$ such that the weights are not driven towards infinity (see for example [3]).

4. Performance evaluation

In order to evaluate a verification MLP, we measure its verification error rate for a given dataset and compare it to a reference error rate which is optimal in a certain sense. By formulating our verification task as a classification problem, we can use the Bayes error as a reference. The Bayes error is known to be optimal for classification problems given the distribution of the data.

To reformulate a verification task as a classification problem, each pair of vectors is assigned one of the following two groups:

G_S group of all vector pairs where the two vectors are from the same class

G_D group of all vector pairs where both vectors are from different classes.

In the case of synthetic data it is possible to calculate the Bayes verification error since the data distributions are given in a parametric form. For real-world problems the data distributions are not given in a parametric form and hence the Bayes verification error can't be computed directly. In this case we can use a k nearest neighbor (KNN) classifier to asymptotically approach the Bayes error as described below.

The KNN approach is a straightforward means of classification. The training set for the KNN algorithm consists of training vectors with known classification $(a_{tr,i}, b_{tr,i})$ where $a_{tr,i}$ is the training vector and $b_{tr,i}$ is its associated class. A test vector $a_{tst,j}$ is classified by seeking the k nearest training vectors $a_{tr,i}$ and it is assigned to the class which is most often present among the k nearest neighbors. The KNN classifier is known to reach the Bayes error when an infinite number of training vectors is available (see e.g. [4]) and is therefore a means to approximate the Bayes error if the data distributions are not known in a parametric form.

5. Experiments with synthetic data

The aim of the experiments with synthetic datasets, i.e. datasets with known data distributions, was to test if the verification MLP achieves the lowest possible verification error from a Bayesian point of view. The data sets had 2 to 4 classes and were 2- to 5-dimensional. We illustrate these investigations by means of an experiment with a 2-dimensional dataset with 2 classes that are distributed as shown in Figure 2.

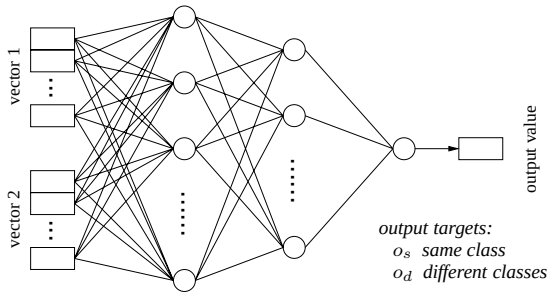


Figure 1: Structure of the verification MLPs.

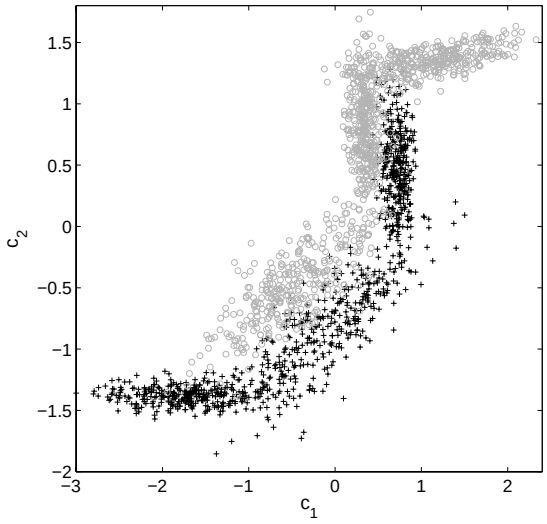


Figure 2: Synthetic data: 2 classes with 2-dimensional non-Gaussian distributions.

The number of training epochs which were necessary to train the verification MLP depended largely on the type of the dataset. We observed the following dependencies:

- If only a few features carried discriminating information and all other features were just random values the verification MLP learned quickly which features were useful and which ones could be neglected.
- The shape of the distributions strongly influenced the number of epochs that were necessary for the training. For example, two classes distributed in two parallel stripes or classes that had a non-linear Bayes decision boundary, such as those shown in Figure 2, required many epochs.

Figure 3 shows the error rates of different verification methods for data distributed as shown in Figure 2. It can be seen that the error of the verification MLP is almost as low as the Bayes error. Note that the MLP was trained with a fixed number of 20'000 vector pairs. We are only interested in the best possible verification error for a given task and not in the verification error in function of the number of training vectors (see Section 1). Therefore the MLP training set was chosen as large as necessary.

For all investigated datasets the verification error achieved with the verification MLP was not significantly higher than the Bayes verification error.

6. Experiments with speech data

6.1. Data description and feature extraction

For a speaker verification task with single speech frames we used speech signals from 48 male speakers recorded from different telephones. From short speech segments (32 ms frames) the 13 first Mel frequency cepstral coefficients (MFCCs) were extracted and used as feature vectors for our experiments. For

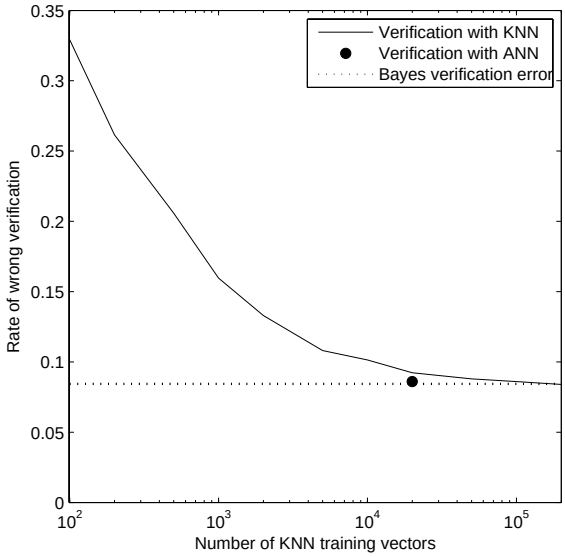


Figure 3: Class verification error for the test set as shown in Figure 2: the KNN verification error is shown in function of the training set size. As expected, with increasing size it approximates the Bayes limit which is indicated by the dotted line. The error rate of the verification MLP is close to the Bayes error.

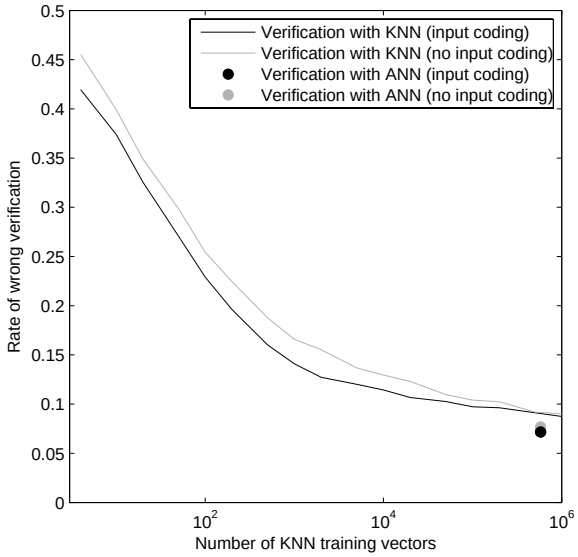


Figure 4: Phoneme verification task: The KNN error rates decrease with increasing number of KNN training vectors. The error rates of the verification MLPs are shown as dots. The error rates for both, KNN and MLP are given for coded and uncoded input vectors.

the phoneme verification task the first derivatives of the MFCCs were used in addition to the static MFCCs. The data from all speakers was divided into 3 disjoint sets (i.e. no speaker was present in more than one set). The MLP and KNN training vector pairs were extracted from the *training set* (26 speakers). The *validation set* (10 speakers) was used to stop the MLP training at the optimal point and to find the optimal k of the KNN classifier. The test vector pairs were taken from the *test set* (12 speakers). Within all sets vector pairs were formed in a way that the two vectors of a pair were always from the same phoneme. In every set the number of pairs with vectors of the same speaker and the number of pairs with vectors from different speakers were equal.

6.2. Phoneme Verification

In this task the objective was to decide whether two speech feature vectors originate from the same phoneme. In this task the same classes (phonemes) are present in all 3 datasets since all signals have similar phonetic content. Yet all sets are extracted from different speakers as is described in Section 6.1.

Because of the slow convergence of the KNN for this verification problem we used two types of input, namely pairs of concatenated vectors $\boldsymbol{p}_{in} = (\boldsymbol{x}, \boldsymbol{y})$ as mentioned above and coded vector pairs $\boldsymbol{p}_{in} = (|\boldsymbol{x} - \boldsymbol{y}|, \boldsymbol{x} + \boldsymbol{y})$ (see [5] for details about the input coding). This input coding sped up the training of the MLPs and led to a faster convergence of the KNN.

The verification error of an MLP trained with 580000 vector pairs is shown in Figure 4. For comparison also the KNN error rate in function of the training set size is drawn. It can be seen that with this data the verification KNN converges much slower than with synthetic data. This was expected because of the more complex nature of the problem. It can only be guessed where the asymptote and therefore the Bayes error will be. It seems that the verification error of the MLP is close to the Bayes verification error however. Since we did not have enough training data we could not prove this assumption. Furthermore it does not seem that the input coding had a big effect on the optimal verification result - the verification MLP which used coded input vectors was even a bit better.

6.3. Speaker Verification

In this task the objective was to decide whether two speech frames are from speech signals of the same speaker or not. In this case all 3 sets of classes (speakers) were disjoint. Therefore a good generalization of the verification MLP is required.

The experiment results are shown in Figure 5. It can be seen that the KNN verification error in function of the training set size decreases much slower than in the experiments done with synthetic data and does not even reach the verification error of the MLP. This is possible since the training and test set have some mismatch because the speaker sets are disjoint. Here it can be seen very well that the KNN which is based on coded vector pairs converged much quicker. In this case the verification ANN which used coded vector pairs was a bit worse however.

The MLP has a quite low verification error if it is considered that the feature vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ were extracted from single speech frames only. If all phonetically matching frame pairs of two equally worded speech segments of about 1 s length are fed separately into the MLP and the output values of the MLP are averaged, the verification error rate is about 6 %.

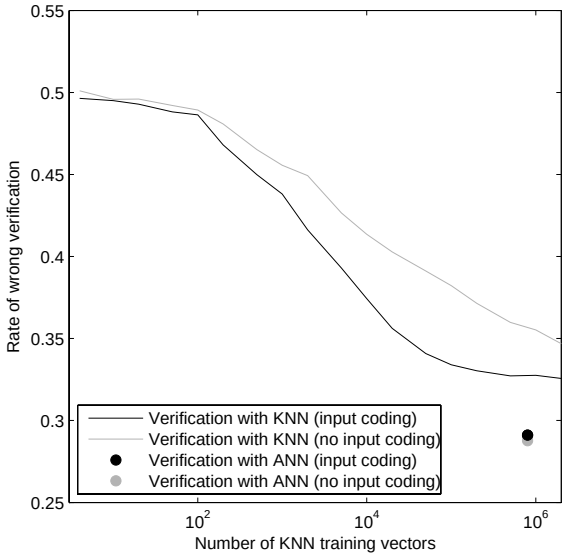


Figure 5: Speaker verification: The KNN error rates decrease with increasing number of KNN training vectors. The error rates of the verification MLPs are shown as dots. The error rates for both, KNN and MLP are given for coded and uncoded input vectors.

7. Conclusions

By means of experiments we have shown that the error rate of an appropriately configured and trained verification MLP is close to the Bayes error rate. Depending on the class distributions, the training can be fairly time-consuming, however. This is not critical in our application since the MLP is class independent and does not need to be trained whenever new classes are added to the application.

For speech data with a virtually unlimited set of classes, as is for example the case in speaker verification, MLP-based class verification has shown to be very efficient not only in terms of verification error but also with respect to computational complexity. For a speaker-verification task the good generalization property of the verification MLP could be shown. Thus the verification MLPs are able to learn a general rule to distinguish between classes rather than class-specific features.

8. Acknowledgements

This work was partly funded by the Swiss National Center of Competence in Research IM2.

9. References

[1] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proceedings of the ICASSP 2000*, vol. 3, 2000, pp. 1635–1638.

[2] M. Gerber and B. Pfister, "Quasi text-independent speaker verification with neural networks," MLMI'05 Workshop, Edinburgh (United Kingdom), July 2005.

[3] S. Haykin, *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice-Hall, 1999.

[4] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley Interscience, 2001.

[5] U. Niesen and B. Pfister, "Speaker verification by means of ANNs," in *Proceedings of ESANN'04, Bruges (Belgium)*, April 2004, pp. 145–150.

Manifold Learning-based Feature Transformation for Phone Classification

Andrew Errity, John McKenna and Barry Kirkpatrick

School of Computing
Dublin City University, Dublin 9, Ireland
{andrew.errity, john.mckenna, barry.kirkpatrick}@computing.dcu.ie

Abstract

This paper investigates approaches for low dimensional speech feature transformation using manifold learning. It has recently been shown that speech sounds may exist on a low dimensional manifold nonlinearly embedded in high dimensional space. A number of techniques have been developed in recent years that attempt to discover the geometric structure of the underlying low dimensional manifold. The manifold learning techniques locally linear embedding and Isomap are considered in this study. The low dimensional feature representations produced by these techniques are applied to several phone classification tasks on the TIMIT corpus. Classification accuracy is analysed and compared to conventional MFCC features and PCA, a linear dimensionality reduction method, transformed features. It is shown that features resulting from manifold learning are capable of yielding higher classification accuracy than these baseline features. The best phone classification accuracy in general is demonstrated by feature transformation with Isomap.

1. Introduction

Feature transformation is an important part of the speech recognition process and can be viewed as a two step procedure. Firstly, relevant information is extracted from short time segments of the acoustic speech signal using a procedure such as Fourier analysis, cepstral analysis or some other perceptually motivated analysis. The resulting D-dimensional parameter vectors are then transformed to a feature vector of lower dimensionality d ($d \leq D$). The aim of dimensionality reduction is to produce features which are concise low dimensional representations that retain the most discriminating information for the intended application and are thus more suitable for pattern classification. Dimensionality reduction also decreases the computational cost associated with subsequent processing.

Physiological constraints on the articulators limit the degrees of freedom of the speech production apparatus. As a result humans are only capable of producing sounds occupying a subspace of the acoustic space. Thus, speech data can be viewed as lying on or near a low dimensional manifold embedded in the original acoustic space. The underlying dimensionality of speech has been the subject of much previous research including classical dimensionality reduction analysis [1, 2], nonlinear dynamical analysis [3] and manifold learning [4]. The consensus of this work is that some speech sounds, particularly voiced speech, are inherently low dimensional.

Dimensionality reduction methods aim to discover this underlying low dimensional structure. These methods can be categorised as linear or nonlinear. Linear methods are limited to discovering the structure of data lying on or near a linear subspace of the high dimensional input space. The most widely used linear dimensionality reduction methods include principal component analysis (PCA) [5] and linear discriminant analysis (LDA) [6]. These methods have been successfully applied to feature transformation in speech processing applications [7, 8] in the past.

However if speech data occupies a low dimensional submanifold nonlinearly embedded in the original space, as proposed previously [2, 4], linear methods will fail to discover the low dimensional structure. A number of manifold learning, also referred to as nonlinear dimensionality reduction, algorithms have been developed [9–11] which overcome the limitations of linear methods. Manifold learning algorithms have recently been shown to be useful in a number of speech processing applications including low dimensional visualization of speech [4,11–14] and limited phone classification tasks [14,15].

In this paper, we build upon previous work and apply two manifold learning algorithms, locally linear embedding (LLE) [9] and isometric feature mapping (Isomap) [10], to extract features from speech data. These features are evaluated in phone classification experiments using a support vector machine (SVM) [16] classifier. The classification performance of these features is compared to baseline Mel-frequency cepstral coefficients (MFCC) and those resulting from the classical linear method, PCA.

The remainder of this paper is structured as follows. In Section 2, the manifold learning algorithms LLE and Isomap are briefly described. Section 3 details the experimental procedure, data set, parameter extraction, feature transformation and classification technique used. Results are examined and discussed in Section 4, with conclusions presented in Section 5. Finally, possibilities for future work are outlined in Section 6.

2. Manifold learning algorithms

2.1. Locally linear embedding

LLE [9] is an unsupervised learning algorithm that computes low dimensional embeddings of high dimensional data. The principle of LLE is to compute a low dimensional embedding with the property that nearby points in the high dimensional space remain nearby and similarly co-located with respect to one another in the low dimensional space. In other words, the embedding is optimised to preserve local neighbourhoods.

The LLE algorithm can be summarised in three steps:

1. For each data point X_i, compute its k nearest neighbours (based on Euclidean distance or some other appropriate definition of 'nearness').
2. Compute weights W_{ij} that best reconstruct each data point X_i from its neighbours, minimising the reconstruction error E:

$$E(W) = \sum_i \left| X_i - \sum_j W_{ij} X_j \right|^2 \qquad (1)$$

3. Compute the low dimensional embeddings Y_i, best reconstructed by the weights W_{ij}, minimising the cost function Ω:

$$\Omega(W) = \sum_i \left| Y_i - \sum_j W_{ij} Y_j \right|^2 \quad (2)$$

In step 2, the reconstruction error is minimised subject to two constraints: first, that each input is reconstructed only from its nearest neighbours, or $W_{ij} = 0$ if X_i is not a neighbour of X_j; second, that the reconstruction weights for each data point sum to one, or $\sum_j W_{ij} = 1 \; \forall i$. The optimum weights for each input can be computed efficiently by solving a constrained least squares problem.

The cost function in step 3 is also based on locally linear reconstruction errors, but here the weights W_{ij} are kept fixed while optimising the outputs Y_i. The embedding cost function in Equation (2) is a quadratic function in Y_i. The minimisation is performed subject to constraints that the outputs are centered and have unit covariance. The cost function has a unique global minimum solution for the outputs Y_i. This is the result returned by LLE as the low dimensional embedding of the high dimensional data points X_i.

2.2. Isomap

The Isomap algorithm [10] offers a differently motivated approach to manifold learning. Isomap is a nonlinear generalisation of multidimensional scaling (MDS) [6] that seeks a mapping from high dimensional space $\mathbf{X}$ to low dimensional feature space $\mathbf{Y}$ that preserves geodesic distances between pairs of data points—that is, distances on the manifold from which the data is sampled.

While Isomap and LLE have similar aims, Isomap is based on a different principle than LLE. In particular, Isomap attempts to preserve the global geometric properties of the manifold while LLE attempts to preserve the local geometric properties of the manifold.

As with LLE, the Isomap algorithm consists of three steps:

1. Construct a neighbourhood graph - Determine which points are neighbours on the manifold based on distances $l(i, j)$ between pairs of points i, j in the input space (as in step 1 of LLE). These neighbourhood relations are then represented as a weighted graph over the data points with edges of weight $l(i, j)$ between neighbouring points.
2. Compute the shortest path between all pairs of points among only those paths that connect nearest neighbours using a technique such as Dijkstra's algorithm.
3. Use classical MDS to embed the data in a d-dimensional Euclidean space so as to preserve these geodesic distances.

3. Experiments

3.1. Classification tasks

The objective of these experiments is to perform phone classification using four different feature types: baseline MFCC vectors and features produced by applying PCA, Isomap and LLE to MFCC vectors. Each feature type was evaluated in three phone classification experiments. The first experiment involves distinguishing between a set of five vowels ('aa', 'iy', 'uw', 'eh', and 'ae'). Phones are labeled using TIMIT symbols [17]. In the second test, a further five vowels ('ah', 'ay', 'oy', 'ih' and 'ow') were added to the previous vowel set, forming a more complex ten class vowel classification problem. The final test involves classifying a set of 19 phones into their associated phone classes. The phone classes and phones used were: vowels (listed above), fricatives ('s', 'sh'), stops ('p', 't', 'k'), nasals ('m', 'n') and, semivowels and glides ('l', 'y').

3.2. Data

The speech data used in this study was taken from the TIMIT corpus [17]. This corpus contains 6300 utterances, 10 spoken by each of 630 American English speakers. The speech recordings are provided at a sampling frequency of 16 kHz.

3.3. Parameter extraction

Based on the phonetic transcriptions and associated phone boundaries provided in TIMIT all units of a subset of phones, listed in Section 3.1, were extracted from the corpus. One 40 ms frame was extracted from the middle of each phone unit (units of duration less than 100 ms were discarded). The raw speech frames were amplitude normalised, preemphasized with the filter $H(z) = 1 - 0.98z^{-1}$ and Hamming windowed. Following this preprocessing, 19-dimensional MFCC vectors were computed for each frame. These MFCC vectors serve as both a baseline feature and high dimensional input for PCA, Isomap and LLE methods.

3.4. Feature transformation

For each of the three phone classification experiments, 250 units representing each of the required phones were chosen at random from those extracted above to make up the data set. PCA, Isomap and LLE were applied to the equivalent set of MFCC vectors.

In order to examine the ability of the feature transformation methods to compute concise representations of the input vectors retaining discriminating information, the dimensionality of the resulting feature vectors was varied from 1 to 19. A separate classifier was subsequently trained and tested using feature vectors with each of the 19 different dimensionalities. Thus the ability of these feature transformation methods to produce useful low dimensional features could be evaluated and changes in performance with varying dimension analysed. As a baseline the original MFCC vectors were used, also varying in dimensionality from 1 to 19.

The number of nearest neighbours, k, used in Isomap and LLE was set equal to 14 and 6 respectively. These values were chosen empirically by varying k and examining classification performance. The performance of both methods was found to be sensitive to the choice of k.

3.5. Support vector machine classification

SVM [16], a powerful classification tool, was used in these experiments. SVM is a binary pattern classification algorithm. For our experiments it is necessary to construct a multiclass classifier. This was achieved using a one-against-one training scheme, training one classifier for every possible pair of classes. The final classification result was determined by majority voting.

It is also necessary to choose an appropriate kernel function to be used in the SVM. In order to select an effective

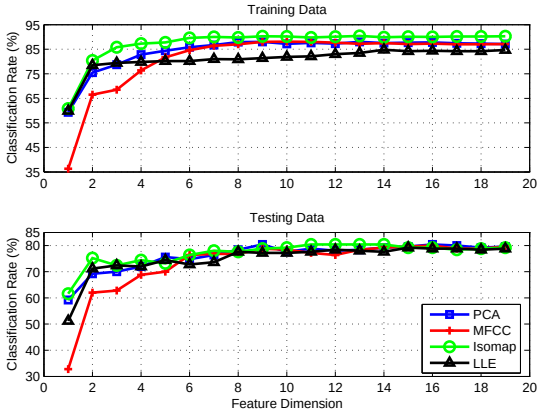


Figure 1: *Five vowel classification results for baseline MFCC, PCA, Isomap and LLE features on the TIMIT database.*

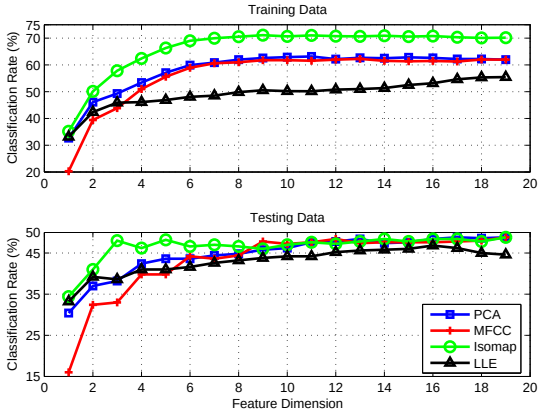


Figure 2: *Ten vowel classification results for baseline MFCC, PCA, Isomap and LLE features on the TIMIT database.*

kernel, different SVM models using linear, polynomial and radial basis function (RBF) kernels were evaluated in a number of phone classification tasks. SVM with RBF kernel demonstrated the best classification accuracy and is used for classification throughout this work. The RBF kernel used is given in Equation (3) below, with x and x' feature vectors and d the feature vector dimensionality.

$$K(x, x') = exp(-\frac{1}{d}\left\|x - x'\right\|^2) \tag{3}$$

In all classification experiments 80% of the data was assigned as training data with the remaining 20% withheld and used as unseen testing data.

4. Results

In each experiment the classifier was evaluated on each of the four feature types. The dimensionality of the feature vectors used in the experiment vary from 1 to 19—the original, full dimension. Results are presented for evaluation on both the training data and testing data.

Fig. 1 shows the results of the five vowel classification task using the baseline MFCC, PCA, Isomap and LLE features. The percentage of phones correctly classified is given on the vertical axis. The horizontal axis represents the dimensionality of the feature vector. The results in Fig. 1 can be summarized as follows:

- The performance of the baseline MFCC vectors improves with increasing dimensionality, plateauing at a dimensionality of approximately 8.
- PCA features offer improvements over baseline MFCC for low dimensions, 1 to 7.
- For the training data, maximum classification accuracy in all dimensions is demonstrated with Isomap features, outperforming all other features including the original full 19-dimensional MFCC vectors.
- Isomap features also offer performance comparable to, and in some dimensions better than, other features on the testing data.
- Accuracy with LLE features is better than both MFCC and PCA in low dimensions, ($d < 3$). However in higher dimensions LLE features do not offer a performance increase over other methods.

Results for ten vowel classification are given in Fig. 2. The results are similar to those of the task above, with reduced classification accuracy due to increased complexity and increased possibility of phone confusion. The important findings are as follows:

- Again, Isomap performs best for the training data, and also for testing data in low dimensions ($d < 8$).
- Isomap, PCA and MFCC performance reach a flat performance level from approximately 10 dimensions.
- A classification accuracy of 48.2% is achieved on the testing data with 5-dimensional Isomap features. This performance is only exceeded by much higher dimensional, ($d > 12$), MFCC and PCA features.

The mean classification accuracy results for each feature type in the ten vowel classification task are presented in Table 1. The mean accuracy scores were computed for the testing data evaluation. Averages are computed for three dimensionality ranges. It can be seen that Isomap gives the highest average accuracy in all ranges. LLE is shown to perform better than PCA and MFCC in low dimensions.

Dimensions	MFCC	PCA	Isomap	LLE
1–5	32.2000	38.3200	43.5600	38.6000
6–19	47.0000	47.0143	47.5286	44.6286
1–19	43.1053	44.7263	46.4842	43.0421

Table 1: Mean classification accuracy in the ten vowel classification task for MFCC, PCA, Isomap and LLE features.

Phone class classification results are presented in Fig. 3. The following is evident:

- Best accuracy is achieved in all dimensions with Isomap features.
- PCA and MFCC features yield similar performance, with PCA features offering improved accuracy for low dimensional features.

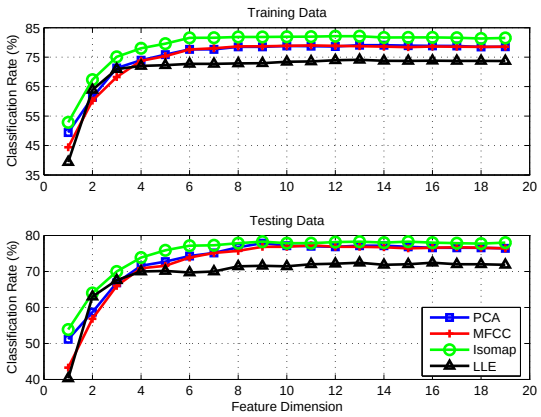


Figure 3: *Phone class classification results for baseline MFCC, PCA, Isomap and LLE features on the TIMIT database.*

- LLE features give the lowest classification rates, except for 2 and 3 dimensional features where they are second only to Isomap.

5. Conclusions

In this paper a phone classification system based on nonlinear manifold learning was proposed and evaluated against a baseline linear dimensionality reduction method, PCA, and conventional MFCC features. All of the dimensionality reduction methods presented outperform the baseline MFCC features for low dimensions. This illustrates the capability of these methods to extract discriminating information from the original 19-dimensional MFCC features.

Higher classification accuracy is shown for manifold learning derived features compared to baseline MFCC and PCA features for low dimensions. Also, in general Isomap yields superior performance to both MFCC and PCA features. This indicates that nonlinear manifold learning algorithms are more capable of retaining information required for discriminating between phones, especially in low dimensional space.

Comparing the manifold learning methods, Isomap demonstrates better classification accuracy than LLE. This indicates that preserving global structure rather than local relationships may be more important for speech feature transformation.

6. Future Work

Possible future work includes the application of the manifold learning feature transformation procedure presented here to continuous ASR. The manifold learning methods described above are batch processing algorithms. A number of out-of-sample extensions have been proposed to overcome this limitation. In the future these out-of-sample approaches could be developed for use with speech data.

7. Acknowledgments

Andrew Errity would like to acknowledge the support of the Irish Research Council for Science, Engineering and Technology; grant number RS/2003/114.

8. References

[1] W. Klein, R. Plomp, and L. C. W. Pols, "Vowel spectra, vowel spaces, and vowel identification," *J. Acoust. Soc. Amer.*, vol. 48, no. 4, pp. 999–1009, 1970.

[2] R. Togneri, M. Alder, and J. Attikiouzel, "Dimension and structure of the speech space," *IEE Proceedings-I*, vol. 139, no. 2, pp. 123–127, 1992.

[3] M. Banbrook, S. McLaughlin, and I. Mann, "Speech characterization and synthesis by nonlinear methods," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 1, pp. 1–17, January 1999.

[4] A. Jansen and P. Niyogi, "Intrinsic Fourier analysis on the manifold of speech sounds," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2006.

[5] I. Jolliffe, *Principal Component Analysis*, ser. Springer Series in Statistics. New York: Springer-Verlag, 1986.

[6] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[7] X. Wang and K. Paliwal, "Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition," *Pattern Recognition*, vol. 36, no. 10, pp. 2429–2439, October 2003.

[8] P. Somervuo, "Experiments with linear and nonlinear feature transformations in HMM based phone recognition based phone recognition," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, April 2003, pp. 52–55.

[9] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, December 2000.

[10] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[11] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14. Cambridge, MA: MIT Press, 2002, pp. 585–591.

[12] R. M. Hegde and H. A. Murthy, "Cluster and intrinsic dimensionality analysis of the modified group delay feature for speaker classification," *Lecture Notes in Computer Science*, vol. 3316, pp. 1172–1178, January 2004.

[13] V. Jain and L. Saul, "Exploratory analysis and visualization of speech and music by locally linear embedding," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 3, 2004, pp. 984–987.

[14] A. Errity and J. McKenna, "An investigation of manifold learning for speech analysis," in *Proc. of the Int. Conf. on Spoken Language Processing (ICSLP)*, Pittsburgh PA, USA, September 2006, pp. 2506–2509.

[15] M. Belkin and P. Niyogi, "Semi-supervised learning on Riemannian manifolds," *Machine Learning*, vol. 56, no. 1 - 3, pp. 209–239, July 2004.

[16] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer, N.Y., 1995.

[17] J. S. Garofalo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, *The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CDROM*, NIST, 1990.

Word Recognition with a Hierarchical Neural Network

Xavier Domont[1,2]*, Martin Heckmann*[1]*, Heiko Wersing*[1]*,*
Frank Joublin[1]*, Stefan Menzel*[1]*, Bernhard Sendhoff* [1]*, Christian Goerick*[1]

[1]Honda Research Institute Europe, 63073 Offenbach/Main, Germany
{firstname.lastname}@honda-ri.de

[2]Technische Universität Darmstadt, 64283 Darmstadt, Germany
xavier.domont@rtr.tu-darmstadt.de

Abstract

In this paper we propose a feedforward neural network for syllable recognition. The core of the recognition system is based on a hierarchical architecture initially developed for visual object recognition. We show that, given the similarities between the primary auditory and visual cortexes, such a system can successfully be used for speech recognition. Syllables are used as basic units for the recognition. Their spectrograms, computed using a Gammatone filterbank, are interpreted as images and subsequently feed into the neural network after a preprocessing step that enhances the formant frequencies and normalizes the length of the syllables. The performance of our system has been analyzed on the recognition of 25 different monosyllabic words. The parameters of the architecture have been optimized using an evolutionary strategy. Compared to the Sphinx-4 speech recognition system, our system achieves better robustness and generalization capabilities in noisy conditions.

1. Introduction

The aim of the proposed speech recognition architecture is to overcome the limitations of conventional, HMM-based, systems which substantially lack robustness against noise.

It has recently been shown that the time-frequency receptive fields in the primary auditory cortex of ferrets have strong similarities to those of the visual cortex [1]. They are selective to modulations in the time-frequency domain and have Gabor-like shapes. A mathematical model of these receptive fields was given in [2] and has already been used for source separation [3] and speech detection [4]. As Gabor-like filters are extensively used in object recognition systems [5, 6], we decided to develop a system for speech recognition by adapting the feedforward neural network initially developed by Wersing and Körner for object recognition [6].

Syllables are the basic units for speech production and show less co-articulatory effects across their boundaries. Therefore, we believe that they are the adequate speech units for our biologically-inspired system. Moreover, the syllable segmentation required for the training of the system seems biologically plausible for speech acquisition.

The building blocks of the system (Fig. 1) are detailed in the following sections. After explaining how we optimized the parameters of the architecture using an evolutionary strategy, we will compare our results to a state of the art speech recognition system and conclude with a discussion of the obtained results.

2. Preprocessing of the spectrogram

The preprocessing mainly aims at transforming a previously segmented speech signal, corresponding to one syllable, into an "image" that is fed into the hierarchical recognition architecture. A two-dimensional representation of a signal is obtained by computing its spectrogram. In addition to the phonetic information, the speech signal also contains many speaker and recording specific information. As the phonetic information is chiefly conveyed by the formant trajectories, we enhance them in the spectrograms prior to recognition.

We used a Gammatone filterbank to compute the spectrogram of the signal. It models the response of the basilar membrane in the human inner ear and is, therefore, adapted to a biology-inspired system. The signal's sampling frequency is $16\,\mathrm{kHz}$. The filterbank has 128 channels ranging from $80\,\mathrm{Hz}$ to $8\,\mathrm{kHz}$. The left part of Fig. 2 shows the response of the Gammatone filterbank after rectification and low-pass filtering. To compensate for the influence of the speech excitation signal, the high frequencies are emphasized by $+6\,\mathrm{dB}$ per octave resulting in a flattened spectrogram (Fig. 2 center). Next, the formant frequencies are enhanced by filtering along the channel axis using mexican-hat filters (Fig. 2 right), only the positive values are kept. For the filtering the size of the kernel is channel-dependent, varying from $90\,\mathrm{Hz}$ for low frequencies to $120\,\mathrm{Hz}$ for high frequencies. This takes the logarithmic arrangement of the center frequencies in the Gammatone filterbank into account.

Finally, the length of the spectrogram is scaled using linear interpolation so that all the spectrograms feeding the recognition hierarchy have the same size. The sampling rate is then reduced to $100\,\mathrm{Hz}$. By doing so syllables of different lengths are scaled to the same length. This relies on the assumption that a linear scaling can handle variations in the length of the same syllable uttered at different speaking rates. However, these are known to be non-linear. In particular, some parts of the signal, like vowels, are more affected by variation in the speech rate than other parts, e.g. plosives. The generalization over these variations is a main challenge in the recognition task. In order to also assess the performance of the recognition hierarchy independent of this non-linear scaling, we applied the Dynamic Time Warping (DTW) method to the spectrograms. For each syllable, we selected one single repetition as reference template and aligned the other by DTW.

Afterwards the syllables were again scaled to the same length and downsampled. At the output of the preprocessing stage the spectrograms feeding the recognition hierarchy have

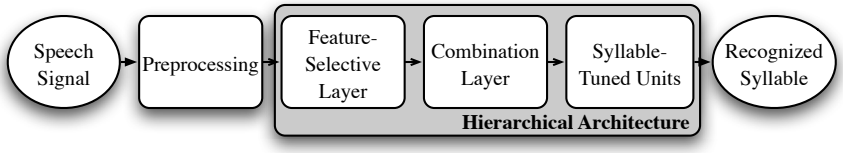


Figure 1: Overview of the system.

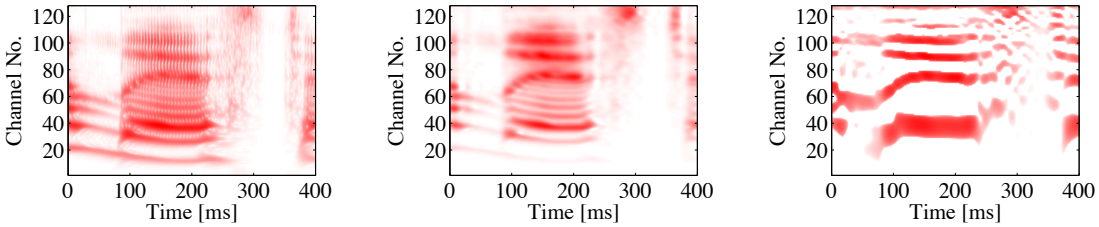


Figure 2: Overview of the preprocessing step for the word "list" spoken by a female American speaker. The 128 channels logarithmically span the frequency range from 80 Hz to 8 kHz. Left: Response of the basilar membrane. Center: After a low-pass filtering over time and a preemphasis has been applied. Right: The harmonic structure has been removed using a filtering along the frequency axis.

all the size of 128×128, i.e. 128 time frames over 128 frequency channels. Note, however, that the application of DTW requires that a hypothesis for the syllable is available. Thus, it cannot easily be applied to a real recognition test.

3. The recognition hierarchy

The preprocessed two-dimensional spectrogram is from now on considered to be an image and feeds into a feedforward architecture initially aimed at visual object recognition. However, the structure of spectrograms differs from the structure of images taken from objects and, while keeping the overall layout of the network described in [6], the receptive fields and the parameters of the neurons were retrained for the task of syllable recognition. The recognition hierarchy is illustrated in Fig. 3.

3.1. Feature-Selective Layer

The first feature-matching stage consists of a linear receptive field summation, a Winner-Take-Most (WTM) and a pooling mechanism. The preprocessed spectrogram is first filtered by eight different Gabor-like filters. The purpose of these filters is to extract local features from the spectrogram. In [6] the receptive fields were chosen as four first-order even Gabor filters. For syllable recognition, 8 receptive fields were learned using independent component analysis on 3500 randomly selected local patches of preprocessed spectrograms.

The WTM competition mechanism between features at the same position introduces nonlinearity into the system. The value $r_l(t,f)$ of the spectrogram in the lth neuron of the feature-selective layer after the WTM competition is given at the position (t,f) by the following equation:

$$r_l(t,f) = \begin{cases} 0, & \text{if } \frac{q_l(t,f)}{M(t,f)} < \gamma_1 \text{ or } M(t,f) = 0 \\ \frac{q_l(t,f) - \gamma_1 M(t,f)}{1-\gamma_1}, & \text{else} \end{cases} \quad (1)$$

where $q_l(t,f)$ is the value of the spectrogram before the WTM competition, $M(t,f) = \max_k q_k(t,f)$ the maximal value at position (t,f) over the eight neurons and $0 \leq \gamma_1 \leq 1$ is a parameter controlling the strength of the competition. A threshold θ_1 is applied to the activity $r_l(t,f)$. This threshold is common for all the neurons in the layer. The pooling performs a downsampling of the spectrogram by four in both time and frequency direction. It is done by a Gaussian receptive field with width σ_1. The feature-selective layer transforms the 128×128 original spectrogram to eight 32×32 spectrogram feature maps.

3.2. Combination Layer

The goal of the combination layer is to detect relevant local feature combinations in the first layer. Similar to the previous layer it consists of a linear receptive field summation, a Winner-Take-Most and a pooling mechanism. These combination cells are learned using the non-negative sparse coding method (NNSC) as in [6], however no invariance transformations have been implemented at this stage. Similarly to Non-Negative Matrix Factorization (NMF), the NNSC method decomposes data vectors $\mathbf{I}^p$ into linear combinations (with non-negative weights s_i^p) of non-negative features $\mathbf{w}_i$ by minimizing the following cost function:

$$E = \sum_p \|\mathbf{I}^p - \sum_i s_i^p \mathbf{w}_i\|^2 + \beta \sum_p \sum_i |s_i^p|.$$

NNSC differs from NMF by the presence of a sparsity enforcing term in the cost function, controlled by the parameter β, which aims at limiting the number of non-zero coefficients required for the reconstruction. Consequently, if a feature appears often in the data, it will be learned, even if it can be obtained by a combination of two or more other features. Therefore, the NNSC is expected to learn complex and global features appearing in the data. An comprehensive description of this method can be found in [7].

For the proposed syllable recognition system 50 complex features $\mathbf{w}_i$ have been learned from image patches extracted from the output of the feature-selective layer. At last, a WTM competition (γ_2, θ_2) and pooling (σ_2) are applied to the 50 neurons and their size is reduced to 16×16.

3.3. Syllable-Tuned Units

In the last stage of the architecture, linear discriminant classifiers are learned based on the output of the combination layer. A classical gradient descent is used for this supervised learning including an early stopping mechanism to avoid overfitting. The obtained classifiers are called Syllable-Tuned Units (STUs) in reference to the View-Tuned Units used in [5] and [6].

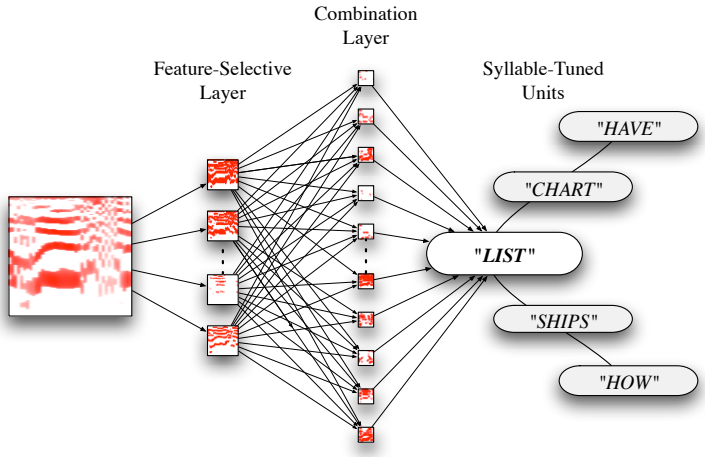


Figure 3: The system is based on a feedforward architecture with weight-sharing and a succession of feature sensitive matching and pooling stages. It comprises three stages arranged in a processing hierarchy.

4. Optimization of the architecture

The performance of the recognition highly depends on the choice of the non-linearities present in the hidden layers of the architecture, i.e. the coefficients and the thresholds of the WTM competitions (Eq. 1) and the width of the poolings. The six parameters ($\gamma_{1,2}$, $\theta_{1,2}$ and $\sigma_{1,2}$) have to be tuned simultaneously and the receptive field of the combination layer as well as the Syllable-Tuned Units have to be learned at each iteration, similarly to the method used in [8].

Practically, this tuning of the model parameter set has been realized within an evolutionary optimization aiming at maximizing the recognition performance in a clean speech scenario. Due to the stochastic components and the use of a population of solutions evolutionary algorithms need more quality evaluations than other algorithms, but on the other hand they allow for a global search and are able to overcome local optima. In the present context, an evolutionary strategy with global step size adaptation (GSA-ES) has been applied relying on similar ranges of the object variables. Initially, standard values, see [9, 10], have been used and then tuned in some test experiments to this specific task. Based on these experiments we have chosen a population size of 32 individuals. Each generation, the two individuals with the best performance have been chosen as parents for the next generation. The optimization parameters have been scaled and the initial global step size was set to 0.003.

Although the evolutionary optimization used a clean scenario for the performance evaluation of each individual we will show that the optimized parameters are robust with respect to noisy signals.

5. Recognition performance

In order to evaluate the performance of the system, a database was built using 25 very frequent monosyllabic words extracted from the DARPA Resource Management (RM) database. Isolated monosyllabic words have been chosen in lack of a syllable segmented database with sufficient size. The words were segmented using forced-alignment. For each of the monosyllabic words we selected 140 occurrences from 12 different speakers (6 males and 6 females) from the speaker dependent part of the database. 70 repetitions of each word were used for training, 20 for the early stopping validation of the Syllable-Tuned Units and 50 for testing.

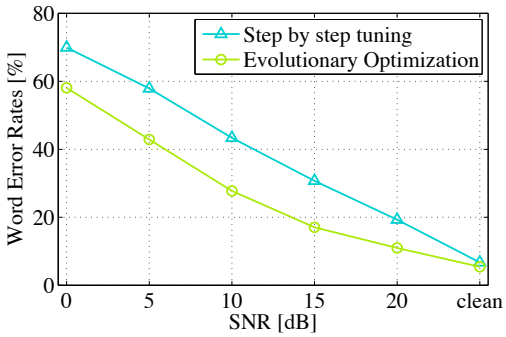


Figure 4: Improvement of the recognition performance using an evolutionary algorithm to tune the parameters, compared to manual tuning one layer after the other. The spectrograms are scaled using a linear interpolation.

The performance of our system has been compared to the Sphinx-4 speech recognition system, an open source speech recognition system that performs well on the whole RM corpus [11]. The Hidden Markov Models for Sphinx were trained only on the segmented monosyllabic words. The robustness towards noise has been investigated by adding babble noise to the test database at different signal to noise ratios (SNR) while training was still performed on clean data.

Figure 4 illustrates the gain in performance obtained using the evolutionary algorithm, compared to a manual tuning of the parameters one layer after the other. Following the notation introduced in [6], the optimal parameters given by the evolution strategy are $\gamma_1 = 0.82$, $\theta_1 = 2.66$, $\sigma_1 = 3.16$ for the first layer and $\gamma_2 = 0.84$, $\theta_2 = 2.78$, $\sigma_2 = 1.87$ for the second layer, when linear interpolation is used to scale the signals. Using a DTW, the optimal set of parameters is $\gamma_1 = 0.99$, $\theta_1 = 0.32$, $\sigma_1 = 4$ for the first layer and $\gamma_2 = 0.89$, $\theta_2 = 0.99$, $\sigma_2 = 1.93$. As can be seen, the performance increased due to the optimization at all SNR levels. With clean speech we ob-

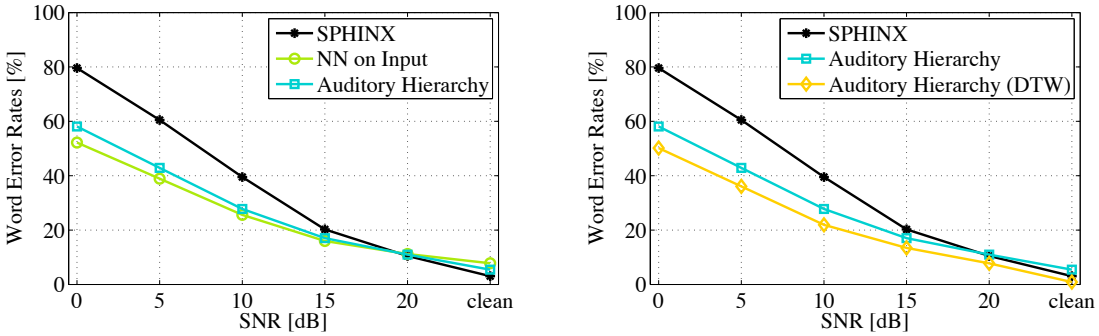


Figure 5: Comparison of the Word Error Rates (WER) between the proposed system and Sphinx-4 in the presence of babble noise. Left: The spectrograms are scaled using a linear interpolation. Comparison between Sphinx-4, a nearest neighbor classifier on the preprocessed spectrograms and the proposed hierarchy. Right: Improvement of the performance when a Dynamic Time Warping method is used to scale the signals.

serve an improvement from 6.72% to 5.44% (19% relative). The largest improvement was achieved at $15\,\mathrm{dB}$ SNR from 30.72% to 17.04% (44.5% relative).

Fig. 5 summarizes the performance of both Sphinx-4 and the proposed system. To measure the baseline similarities of the image ensemble, we also give the performance of a nearest neighbor classifier (NN) that matches the test data against all available training "views". An exhaustive storage of examples is, however, not a viable model for auditory classification. With clean signals, the STUs show better generalization capabilities and perform better than a nearest neighbor on the input layer (Fig. 5 left). For noisy signals, the STUs are slightly worse, however, at a strong reduction of representational complexity.

With a simple linear time scaling our system only outperforms Sphinx-4 in noisy conditions but shows inferior performance on clean data. When Dynamic Time Warping is used to properly scale the signals, the STUs improve the already good performance obtained directly after the preprocessing in all the cases and our system outperforms Sphinx-4 even for clean signals (Fig. 5 right). With clean data Sphinx obtains a 3.1% Word Error Rate (WER), our system achieves 0.9% WER with the DTW and 5.4% without the DTW.

6. Discussion

In this paper, we presented a novel approach to speech recognition interpreting spectrograms as images and deploying a hierarchical object recognition system. To optimize the main free parameters of the system, we used an evolutionary algorithm which allows us to quickly change the system without the need for manual parameter tuning.

We could show that our system performs better than a state of the art system in noisy conditions even when we applied a simplistic linear scaling of the input for time alignment. When we aligned the current utterance with the DTW to a known representation in an optimal non-linear way, we obtained better than state of the art results for all cases tested. However, in its current form the DTW makes use of information not available in real situations.

From this we conclude that our architecture and the underlying features are more robust against noise than the commonly used mel frequency cepstral coefficients (MFCCs). This robustness against noise is very important for real world scenarios which are usually characterized by significant background noise and variations in the recording conditions. A similar robustness was also observed for visual recognition in clutter scenes [6].

Our comparison between the linear scaling and the DTW shows that the performance of the model could be significantly improved by better temporal alignment. We therefore consider methods for improving this alignment as interesting future research directions.

7. References

[1] S. Shamma, "On the role of space and time in auditory processing," *Trends Cogn. Sci.*, vol. 5, no. 8, pp. 340–348, 2001.

[2] T. Chih, P. Ru, and S. Shamma, "Multiresolution spectrotemporal analysis of complex sounds," *J. Acoust. Soc. Am.*, vol. 118, pp. 887–906, 2005.

[3] M. Elhilali and S. Shamma, "A biologically-inspired approach to the cocktail party problem," in *Proc. ICASSP Conf.*, vol. 5, 2006, pp. V–637–640.

[4] N. Mesgarani, M. Slaney, and S. Shamma, "Discrimination of speech from non-speech based on multiscale spectro-temporal modulations," *IEEE Trans. Speech and Audio Process.*, pp. 920–930, 2006.

[5] M. Riesenhuber and T. Poggio, "Hierachical models of object recognition in cortex," *Nature Neuroscience*, vol. 2, pp. 1019–1025, 1999.

[6] H. Wersing and E. Körner, "Learning optimized features for hierarchical models of invariant recognition," *Neural Computation*, vol. 15, no. 7, pp. 1559–1588, 2003.

[7] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[8] G. Schneider, H. Wersing, B. Sendhoff, and E. Körner, "Evolutionary optimization of a hierarchical object recognition model," *IEEE Trans. Syst., Man and Cybern. B, Cybern.*, vol. 35, no. 3, pp. 426–437, 2005.

[9] H.-P. Schwefel, *Evolution and Optimum Seeking*. John Wiley and sons, New York, 1995.

[10] T. Bäck, *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.

[11] W. Walker, P. Lamere, and P. Kwok, "Sphinx-4: A flexible open source framework for speech recognition," Sun Microsystems Inc., Tech. Rep., 2004.

Discriminative Keyword Spotting

Joseph Keshet[1]*, David Grangier*[2] *and Samy Bengio*[2]

[1] School of Computer Science & Engineering, The Hebrew University, Jerusalem, Israel
jkeshet@cs.huji.ac.il
[2] IDIAP Research Institute, Martigny, Switzerland
{grangier,bengio}@idiap.ch

Abstract

This paper proposes a new approach for keyword spotting, which is not based on HMMs. The proposed method employs a new discriminative learning procedure, in which the learning phase aims at maximizing the area under the ROC curve, as this quantity is the most common measure to evaluate keyword spotters. The keyword spotter we devise is based on non-linearly mapping the input acoustic representation of the speech utterance along with the target keyword into an abstract vector space. Building on techniques used for large margin methods for predicting whole sequences, our keyword spotter distills to a classifier in the abstract vector-space which separates speech utterances in which the keyword was uttered from speech utterances in which the keyword was not uttered. We describe a simple iterative algorithm for learning the keyword spotter and discuss its formal properties. Experiments with the TIMIT corpus show that our method outperforms the conventional HMM-based approach.

1. Introduction

Keyword (or word) spotting refers to a proper detecting of any occurrence of a given word in a speech signal. Most previous work on keyword spotting has been based on hidden Markov models (HMMs). See for example [1, 2, 3] and the references therein. Despite their popularity, HMM-based approaches have several known drawbacks such as convergence of the training algorithm (EM) to a local maxima, conditional independence of observations given the state sequence and the fact that the likelihood is dominated by the observation probabilities, often leaving the transition probabilities unused. However, the most acute weakness of HMMs for keyword spotting is that they do not aim at maximizing the detection rate of the keywords.

In this paper we propose an alternative approach for keyword spotting that builds upon recent work on discriminative supervised learning and overcomes some of the inherent problems of the HMM-based approaches. The advantage of discriminative learning algorithms stems from the fact that the objective function used during the learning phase is tightly coupled with the decision task one needs to perform. In addition, there is both theoretical and empirical evidence that discriminative learning algorithms are likely to outperform generative models for the same task (see for instance [4, 5]). One of the main goals of this work is to extend the notion of discriminative learning to the task of keyword spotting.

Our proposed method is based on recent advances in kernel machines and large margin classifiers for sequences [6, 7], which in turn build on the pioneering work of Vapnik and colleagues [4, 5]. The keyword spotter we devise is based on non-linear mapping the speech signal along with the target keyword into a vector-space endowed with an inner-product. Our learning procedure distills to a classifier in this vector-space which is aimed at separating the utterances in which the keyword was uttered from those in which the keyword was not uttered. On this aspect, our approach is hence related to support vector machine (SVM), which has already been successfully applied in speech applications [8, 9]. However, the model proposed in this paper is different from a classical SVM since we are not addressing a simple decision task such as binary classification or regression. Our algorithm is in fact closer to recent work on kernel machine methods for sequence prediction, such as [10, 6, 11], with the main difference that we avoid the costly optimization problems introduced by such models. Instead, we propose an efficient iterative algorithm for learning a discriminative keyword spotter by traversing the training set a single time.

This paper is organized as follows. In Sec. 2 we formally introduce the keyword spotting problem. We then present an iterative algorithm for keyword spotting in Sec. 3. The implementation details of our learning approach and the non-linear set of feature functions we use are presented in Sec. 4. Next, we present experimental results in Sec. 5. Finally, concluding remarks and future directions are discussed in Sec. 6.

2. Problem Setting

Any keyword (or word) is naturally composed of a sequence of phonemes. In the keyword spotting task, we are provided with a speech utterance and a keyword and the goal is to decide whether the keyword is uttered or not, namely, whether the sequence of phonemes was articulated in the given utterance.

Formally, we represent a speech signal as a sequence of acoustic feature vectors $\bar{\mathbf{x}} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$, where $\mathbf{x}_t \in \mathcal{X} \subset \mathbb{R}^d$ for all $1 \leq t \leq T$. We denote a keyword by $k \in \mathcal{K}$, where $\mathcal{K}$ is a lexicon of words. Each keyword k is composed of a sequence of phonemes $\bar{p}^k = (p_1, \ldots, p_L)$, where $p_l \in \mathcal{P}$ for all $1 \leq l \leq L$ and $\mathcal{P}$ is the domain of the phoneme symbols. We denote by $\mathcal{P}^*$ the set of all finite length sequences over $\mathcal{P}$. Our goal is to learn a *keyword spotter*, denoted f, which takes as input the pair $(\bar{\mathbf{x}}, \bar{p}^k)$ and returns a real value expressing the confidence that the targeted keyword k is uttered in $\bar{\mathbf{x}}$. That is, f is a function from $\mathcal{X}^* \times \mathcal{P}^*$ to the set $\mathbb{R}$. The confidence score outputted by f for a given pair $(\bar{\mathbf{x}}, \bar{p}^k)$ can then be compared to a threshold b to actually determine whether $\bar{p}^k$ was uttered in $\bar{\mathbf{x}}$. Let us further define the *alignment* between a keyword phoneme sequence and a speech signal. We denote by $s_l \in \mathbb{N}$ the start time of phoneme p_l (in frame units), and by $e_l \in \mathbb{N}$ the end time of phoneme p_l. We assume that the start time of phoneme p_{l+1} is equal to the end time of phoneme p_l, that is,

$e_l = s_{l+1}$ for all $1 \leq l \leq L-1$. The alignment sequence $\bar{s}^k$ corresponding to the phonemes sequence $\bar{p}^k$ is a sequence of start-times and an end-time, $\bar{s}^k = (s_1, \ldots, s_L, e_L)$, where s_l is the start-time of phoneme p_l and e_L is the end-time of the last phoneme p_L.

Our construction is based on a set of predefined non-linear feature functions $\{\phi_j\}_{j=1}^n$. Each feature function is of the form $\phi_j : \mathcal{X}^* \times \mathcal{P}^* \times \mathbb{N}^* \to \mathbb{R}$. That is, each feature function takes as input an acoustic representation of a speech utterance $\bar{\mathbf{x}} \in \mathcal{X}^*$, together with a keyword phoneme sequence $\bar{p}^k \in \mathcal{P}^*$, and a candidate alignment sequence $\bar{s}^k \in \mathbb{N}^*$, and returns a scalar in $\mathbb{R}$ which represents the confidence in the suggested alignment sequence given the keyword $\bar{p}^k$. For example, one element of the feature function can sum the number of times phoneme p comes after phoneme p', while other elements of the feature function may extract properties of each acoustic feature vector $\mathbf{x}_t$ provided that phoneme p was pronounced at time t. The complete set of the non-linear feature functions we use is described in Sec. 4.

As mentioned above, our goal is to learn a keyword spotter f, which takes as input a sequence of acoustic features $\bar{\mathbf{x}}$, a keyword $\bar{p}^k$, and returns a confidence value in $\mathbb{R}$. The form of the function f we use is

$$f(\bar{\mathbf{x}}, \bar{p}^k) = \max_{\bar{s}} \ \mathbf{w} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}, \bar{p}^k, \bar{s}) \ , \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^n$ is a vector of importance weights that should be learned and $\boldsymbol{\phi} \in \mathbb{R}^n$ is a vector function composed out of the feature functions ϕ_j. In other words, f returns a confidence prediction about the existence of the keyword in the utterance by maximizing a weighted sum of the scores returned by the feature function elements over all possible alignment sequences. The maximization defined by Eq. (1) is over an exponentially large number of all possible alignment sequences. Nevertheless, as in HMMs, if the feature functions $\boldsymbol{\phi}$ are decomposable, the maximization in Eq. (1) can be efficiently calculated using a dynamic programming procedure.

The performance of a keyword spotting system is often measured by the Receiver Operating Characteristics (ROC) curve, that is, a plot of the true positive (spotting a keyword correctly) rate as a function of the false positive (mis-spotting a keyword) rate (see for example [1, 3, 2] and the references therein). The points on the curve are obtained by sweeping the decision threshold b from the most positive confidence value outputted by the system to the most negative one. Hence, the choice of b represents a trade-off between different operational settings, corresponding to different cost functions weighing false positive and false negative errors. Assuming a flat prior over all these cost functions, a criterion to identify a good keyword spotting system that would be good on average for all these settings could be to select the one maximizing the area under the ROC curve (AUC). In the following we propose an algorithm which directly aims at maximizing the AUC.

Recall that we would like to obtain an algorithm that maximizes the AUC on unseen data. In order to do so, we will maximize the AUC over a large set of training examples. Let us consider two sets of examples. Denote by $\mathcal{X}_k^+$ a set of speech utterances in which the the keyword k was uttered. Similarly, denote by $\mathcal{X}_k^-$ a set of speech utterances in which the keyword k was not uttered. The AUC for the keyword k can be written in the form of the Wilcoxon-Mann-Whitney statistics [12] as

$$A_k = \frac{1}{|\mathcal{X}_k^+||\mathcal{X}_k^-|} \sum_{\substack{\bar{\mathbf{x}}^+ \in \mathcal{X}_k^+ \\ \bar{\mathbf{x}}^- \in \mathcal{X}_k^-}} \mathbb{1}_{\{f(\bar{\mathbf{x}}^+, \bar{p}^k) > f(\bar{\mathbf{x}}^-, \bar{p}^k)\}},$$

where $\mathbb{1}_{\{\cdot\}}$ refers to the indicator function and $\bar{p}^k$ refers to the phoneme sequence corresponding to keyword k. Thus, A_k estimates the probability that the score assigned to an utterance in which the keyword was uttered is greater than the score assigned to an utterance in which the keyword was not uttered. The average AUC over the set of keywords $\mathcal{K}$ can be written as $A = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} A_k$. In the next section we describe an iterative algorithm for learning the weight vector $\mathbf{w}$, which aims at maximizing the average AUC.

3. An Iterative Algorithm

We now describe a simple iterative algorithm for learning the weight vector $\mathbf{w}$ based on a training set of examples. Each example in the training set S is composed of a keyword phoneme sequence $\bar{p}^k$, an utterance in which the keyword k was uttered $\bar{\mathbf{x}}^+ \in \mathcal{X}_k^+$, an utterance in which the keyword k was not uttered $\bar{\mathbf{x}}^- \in \mathcal{X}_k^-$, and an alignment sequence $\bar{s}^k$ that corresponds to the location of the keyword in $\bar{\mathbf{x}}^+$. The algorithm receives as input a set of training examples $S = \{(\bar{p}^{k_i}, \bar{\mathbf{x}}_i^+, \bar{\mathbf{x}}_i^-, \bar{s}_i^{k_i})\}_{i=1}^m$. and examines each of them sequentially. Initially, we set $\mathbf{w} = \mathbf{0}$. At each iteration i, the algorithm updates $\mathbf{w}$ according to the current example $(\bar{p}^{k_i}, \bar{\mathbf{x}}_i^+, \bar{\mathbf{x}}_i^-, \bar{s}_i^{k_i})$ as we now describe. Denote by $\mathbf{w}_{i-1}$ the value of the weight vector before the ith iteration. Let $\bar{s}'$ be the predicted alignment for the negative utterance, $\bar{\mathbf{x}}_i^-$, according to $\mathbf{w}_{i-1}$,

$$\bar{s}' = \arg\max_{\bar{s}} \ \mathbf{w}_{i-1} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i^-, \bar{p}^{k_i}, \bar{s}) \ . \tag{2}$$

Let us define the difference between the feature functions of the acoustic sequence in which the keyword was uttered and the feature functions of the acoustic sequence in which the keyword was not uttered as $\Delta\boldsymbol{\phi}_i$, that is,

$$\Delta\boldsymbol{\phi}_i = \frac{1}{|\mathcal{X}_{k_i}^+||\mathcal{X}_{k_i}^-|} \left(\boldsymbol{\phi}(\bar{\mathbf{x}}_i^+, \bar{p}^{k_i}, \bar{s}^{k_i}) - \boldsymbol{\phi}(\bar{\mathbf{x}}_i^-, \bar{p}^{k_i}, \bar{s}') \right) \ .$$

We set the next weight vector $\mathbf{w}_i$ to be the minimizer of the following optimization problem,

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{R}^n, \xi \geq 0} \quad & \frac{1}{2}\|\mathbf{w} - \mathbf{w}_{i-1}\|^2 + C\,\xi \\ \text{s.t.} \quad & \mathbf{w} \cdot \Delta\boldsymbol{\phi}_i \geq 1 - \xi \ , \end{aligned} \tag{3}$$

where C serves as a complexity-accuracy trade-off parameter (see [13]) and ξ is a non-negative slack variable, which indicates the loss of the ith example. Intuitively, we would like to minimize the loss of the current example, i.e., the slack variable ξ, while keeping the weight vector $\mathbf{w}$ as close as possible to our previous weight vector $\mathbf{w}_{i-1}$. The constraint makes the projection of the utterance in which the keyword was uttered onto $\mathbf{w}$ higher than the projection of the utterance in which the keyword was not uttered onto $\mathbf{w}$ by at least 1. It can be shown (see [13]) that the solution to the above optimization problem is

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \alpha_i \Delta\boldsymbol{\phi}_i \ . \tag{4}$$

The value of the scalar α_i is based on the different scores that $\bar{\mathbf{x}}^+$ and $\bar{\mathbf{x}}^-$ received according to $\mathbf{w}_{i-1}$, and a parameter C. Formally,

$$\alpha_i = \min \left\{ C, \ \frac{[1 - \mathbf{w}_{i-1} \cdot \Delta\boldsymbol{\phi}_i]_+}{\|\Delta\boldsymbol{\phi}_i\|^2} \right\} \ . \tag{5}$$

The optimization problem given in Eq. (3) is based on ongoing work on online learning algorithms appearing in [13].

Input: training set $S = \{(\bar{p}^{k_i}, \bar{\mathbf{x}}_i^+, \bar{\mathbf{x}}_i^-, \bar{s}^{k_i})\}_{i=1}^m$; validation set S_{val}; parameter C

Initialize: $\mathbf{w}_0 = \mathbf{0}$

For $i = 1, \ldots, m$

Predict: $\bar{s}' = \arg\max_{\bar{s}} \ \mathbf{w}_{i-1} \cdot \boldsymbol{\phi}(\bar{\mathbf{x}}_i^-, \bar{p}^{k_i}, \bar{s})$

Set: $\Delta\boldsymbol{\phi}_i = \frac{1}{|\mathcal{X}_{k_i}^+||\mathcal{X}_{k_i}^-|} \left(\boldsymbol{\phi}(\bar{\mathbf{x}}_i^+, \bar{p}^{k_i}, \bar{s}^{k_i}) - \boldsymbol{\phi}(\bar{\mathbf{x}}_i^-, \bar{p}^{k_i}, \bar{s}')\right)$

If $\mathbf{w}_{i-1} \cdot \Delta\boldsymbol{\phi}_i < 1$

Set: $\alpha_i = \min\left\{C \ , \ \frac{1 - \mathbf{w}_{i-1} \cdot \Delta\boldsymbol{\phi}_i}{\|\Delta\boldsymbol{\phi}_i\|^2}\right\}$

Update: $\mathbf{w}_i = \mathbf{w}_{i-1} + \alpha_i \cdot \Delta\boldsymbol{\phi}_i$

Output: The weight vector $\mathbf{w}^*$ which achieves best AUC performance on the validation set S_{val}.

Figure 1: An iterative algorithm.

Based on that work, it can be shown that, under some mild technical conditions, the cumulative performance of the iterative procedure, i.e., $\frac{1}{m}\sum_{i=1}^m \mathbb{1}_{\{\mathbf{w}_i \cdot \Delta\boldsymbol{\phi}_i > 0\}}$ is likely to be high. Moreover, it can further be shown that if the cumulative performance of the iterative procedure is high, there exists at least one weight vector among the vectors $\{\mathbf{w}_1, \ldots, \mathbf{w}_m\}$ which attains high averaged performance on the test examples as well, that is, there exists a vector which attains high averaged AUC over a set of test examples. To find this weight vector, we simply calculate the averaged loss attained by each of the weight vectors on a validation set. A pseudo-code of our algorithm is given in Fig. 1.

In the case the user would like to select a threshold b that would ensure a specific requirement in terms of true positive rate or false negative rate, a simple cross-validation procedure (see [14]) would consist in selecting the confidence value given by our model at the point of interest over the ROC curve plotted for some validation utterances of the targeted keyword.

4. Non-Linear Feature Functions

In this section we present the implementation details of our learning approach for the task of keyword spotting. Recall that our construction is based on a set of non-linear feature functions, $\{\phi_j\}_{j=1}^n$, which maps an acoustic-phonetic representation of a speech utterance as well as a suggested alignment sequence into an abstract vector-space. In order to make this section more readable we omit the keyword index k.

We introduce a specific set of feature functions, which is highly adequate for the keyword spotting problem. We utilize seven different feature functions ($n = 7$). These feature functions are used for defining our keyword spotting function $f(\bar{\mathbf{x}}, \bar{p})$ as in Eq. (1).

Our first four feature functions aim at capturing transitions between phonemes. These feature functions are the distance between frames of the acoustic signal at both sides of phoneme boundaries as suggested by an alignment sequence $\bar{s}$. The distance measure we employ, denoted by d, is the Euclidean distance between feature vectors. Our underlying assumption is that if two frames, $\mathbf{x}_t$ and $\mathbf{x}_{t'}$, are derived from the same phoneme then the distance $d(\mathbf{x}_t, \mathbf{x}_{t'})$ should be smaller than if the two frames are derived from different phonemes. Formally, our first four feature functions are defined as

$$\phi_j(\bar{\mathbf{x}}, \bar{p}, \bar{s}) = \sum_{i=2}^{|\bar{p}|-1} d(\mathbf{x}_{-j+s_i}, \mathbf{x}_{j+s_i}), \ \ j \in \{1, 2, 3, 4\} \ . \quad (6)$$

If $\bar{s}$ is the correct timing sequence then distances between frames across the phoneme change points are likely to be large. In contrast, an incorrect phoneme start time sequence is likely to compare frames from the same phoneme, often resulting in small distances. Note that the first four feature functions described above use only the start time of the ith phoneme and do not use the values of s_{i-1} and s_{i+1}.

The fifth feature function we use is built from a frame-wise phoneme classifier described in [15]. Formally, for each phoneme event $p \in \mathcal{P}$ and frame $\mathbf{x} \in \mathcal{X}$, there is a confidence, denoted $g_p(\mathbf{x})$, that the phoneme p is pronounced in the frame $\mathbf{x}$. The resulting feature function measures the cumulative confidence of the complete speech signal given the phoneme sequence and their start-times,

$$\phi_5(\bar{\mathbf{x}}, \bar{p}, \bar{s}) = \sum_{i=1}^{|\bar{p}|} \sum_{t=s_i}^{s_{i+1}-1} g_{p_i}(\mathbf{x}_t) \ . \quad (7)$$

The fifth feature function uses both the start time of the ith phoneme and the $(i+1)$th phoneme but ignores s_{i-1}.

Our next feature function scores timing sequences based on phoneme durations. Unlike the previous feature functions, the sixth feature function is oblivious to the speech signal itself. It merely examines the length of each phoneme, as suggested by $\bar{s}$, compared to the typical length required to pronounce this phoneme. Formally,

$$\phi_6(\bar{\mathbf{x}}, \bar{p}, \bar{s}) = \sum_{i=1}^{|\bar{p}|} \log \mathcal{N}(s_{i+1} - s_i; \hat{\mu}_{p_i}, \hat{\sigma}_{p_i}) \ , \quad (8)$$

where $\mathcal{N}$ is a Normal probability density function with mean $\hat{\mu}_p$ and standard deviation $\hat{\sigma}_p$. In our experiments, we estimated $\hat{\mu}_p$ and $\hat{\sigma}_p$ from the training set (see Sec. 5).

Our last feature function exploits assumptions on the speaking rate of a speaker. Intuitively, people usually speak in an almost steady rate and therefore a timing sequence in which speech rate is changed abruptly is probably incorrect. Formally, let $\hat{\mu}_p$ be the average length required to pronounce the pth phoneme. We denote by r_i the relative speech rate, $r_i = (s_{i+1} - s_i)/\hat{\mu}_{p_i}$. That is, r_i is the ratio between the actual length of phoneme p_i as suggested by $\bar{s}$ to its average length. The relative speech rate presumably changes slowly over time. In practice the speaking rate ratios often differ from speaker to speaker and within a given utterance. We measure the local change in the speaking rate as $(r_i - r_{i-1})^2$ and we define the feature function ϕ_7 as the local change in the speaking rate,

$$\phi_7(\bar{\mathbf{x}}, \bar{p}, \bar{s}) = \sum_{i=2}^{|\bar{p}|} (r_i - r_{i-1})^2 \ . \quad (9)$$

5. Experimental Results

To validate the effectiveness of the proposed approach we performed experiments with the TIMIT corpus. We divided the training portion of TIMIT (excluding the SA1 and SA2 utterances) into three disjoint parts containing 500, 80 and 3116 utterances. The first part of the training set was used for learning the functions g_p (Eq. (7)), which define the feature function ϕ_5. Those functions were learned by the algorithm described in [15] using the MFCC+Δ+$\Delta\Delta$ acoustic features and a Gaussian kernel ($\sigma = 6.24$ and $C = 5.0$). The second set of 80 utterances formed the validation set needed for our keyword spotting algorithm. The set was built out of a set of 40 keywords randomly

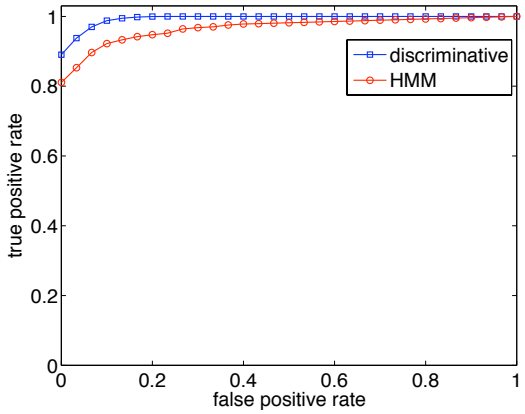


Figure 2: ROC curves of the discriminative algorithm and the HMM approach. The AUC of the ROC curves is 0.99 and 0.96 for the discriminative algorithm and the HMM algorithm, respectively.

chosen from the TIMIT lexicon. The 80 utterances were chosen by pairs: one utterance in which the keyword was uttered and another utterance in which the keyword was not uttered. Finally, we ran our iterative algorithm on the rest of the utterances in the training set. The value of C was set to be 1.

We compared the results of our method to the HMM approach, where each phoneme was represented by a simple left-to-right HMM of 5 emitting states with 40 diagonal Gaussians. These models were enrolled as follows: first the HMMs were initialized using K-means, and then enrolled independently using EM. The second step, often called embedded training, re-enrolls all the models by relaxing the segmentation constraints using a forced alignment. Minimum values of the variances for each Gaussian were set to 20% of the global variance of the data. All HMM experiments were done using the *Torch* package [16]. All hyper-parameters including number of states, number of Gaussians per state, variance flooring factor, were tuned using the validation set.

Keyword detection is performed with a new HMM composed of two sub HMM models, the keyword model and the garbage model. The keyword model is an HMM, which estimates the likelihood of an acoustic sequence given that this sequence represents the keyword phoneme sequence. The garbage model is an HMM composed of phoneme HMMs fully connected with each others, which estimates the likelihood of any acoustic sequence. The overall HMM fully connects the keyword model and the garbage model and the best path found by Viterbi decoding on this overall HMM either passes through the keyword model (in which case the keyword is said to be uttered) or not (in which case the keyword is not in the acoustic sequence).

The test set was composed of 80 keywords, distinct from the keywords of the training and validation set. For each keyword, we randomly picked at most 20 utterances in which the keyword was uttered and at most 20 utterances in which it was not uttered. The number of test utterances in which the keyword was uttered was not always 20, since some keywords were uttered less than 20 times in the whole TIMIT test set. Both the discriminative algorithm and the HMM based algorithm have been evaluated against this data and their results are reported as averaged ROC curves in Fig. 2. The AUC of the ROC curves is 0.99 and 0.96 for the discriminative algorithm and the HMM algorithm, respectively. In order to check whether the advantage over the averaged AUC could be due to a few keyword, we ran the Wilcoxon test. At the 95% confidence level, the test rejected this hypothesis, showing that our model indeed brings a consistent improvement on the keyword set.

6. Conclusions

In this work, we introduced a discriminative approach to keyword spotting. We adopted a large-margin formulation of the problem and proposed a model relying on an objective function related the area under the ROC curve, i.e., the most common measure for keyword spotter evaluation. Compared to state-of-the-art approaches which mostly rely on generative HMM models, the proposed model has shown to yield an improvement over the TIMIT corpus.

Acknowledgments: This research was partly conducted while Joseph Keshet was visiting the IDIAP Research Institute. This research was supported by the PASCAL European Network of Excellence and the DIRAC project.

7. References

[1] M.-C. Silaghi and H. Bourlard, "Iterative posterior-based keyword spotting without filler models," in *ASRU*, 1999.

[2] Y. B. Ayed, D. Fohr, J.-P. Haton, and G. Chollet, "Confidence measure for keyword spotting using support vector machines," in *ICASSP*, 2004.

[3] H. Ketabdar, J. Vepa, S. Bengio, and H. Bourlard, "Posterior based keyword spotting with a priori thresholds," in *Interspeech*, 2006.

[4] V. N. Vapnik, *Statistical Learning Theory*. Wiley, 1998.

[5] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge Univ. Press, 2000.

[6] B. Taskar, C. Guestrin, and D. Koller, "Max-margin markov networks," in *NIPS*, 2003.

[7] S. Shalev-Shwartz, J. Keshet, and Y. Singer, "Learning to align polyphonic music," in *ISMIR*, 2004.

[8] J. Keshet, D. Chazan, and B.-Z. Bobrovsky, "Plosive spotting with margin classifiers," in *Eurospeech*, 2001.

[9] J. Salomon, S. King, and M. Osborne, "Framewise phone classification using support vector machines," in *Interspeech*, 2002.

[10] M. Collins, "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms," in *EMNLP*, 2002.

[11] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *ICML*, 2004.

[12] C. Cortes and M. Mohri, "Confidence intervals for the area under the roc curve," in *NIPS*, 2004.

[13] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive aggressive algorithms," *Journal of Machine Learning Research*, vol. 7, Mar 2006.

[14] S. Bengio, J. Maréthoz, and M. Keller, "The expected performance curve," in *ICML*, 2005.

[15] O. Dekel, J. Keshet, and Y. Singer, "Online algorithm for hierarchical phoneme classification," in *MLMI*, 2004.

[16] R. Collobert, S. Bengio, and J. Mariéthoz, "Torch: a modular machine learning library," IDIAP-RR 46, IDIAP, 2002.

Hybrid models for automatic speech recognition: a comparison of classical ANN and kernel based methods

Ana I. García-Moral, Rubén Solera-Ureña, Carmen Peláez-Moreno and Fernando Díaz-de-María

Department of Signal Theory and Communicationos
University Carlos III Madrid, Leganés (Madrid), Spain
rsolera@tsc.uc3m.es

Abstract

Support Vector Machines (SVM) are state-of-the-art methods for machine learning but share with more classical Artificial Neural Networks (ANN) the difficulty of their application to temporally variable input patterns. This is the case in Automatic Speech Recognition (ASR). In this paper we have recalled the solutions provided in the past for ANN and applied them to SVMs performing a comparison between them. Preliminary results show a similar behaviour which results encouraging if we take into account the novelty of the SVM systems in comparison with classical ANNs. The envisioned ways of improvement are outlined in the paper.

1. Introduction

Hidden Markov Models (HMMs) have become the most employed core technique for Automatic Speech Recognition (ASR). After several decades of intense research work in the field, it seems that the HMM ASR systems are very close to reach their limit of performance. Some alternative approaches, most of them based on Artificial Neural Networks (ANNs), were proposed during the late eighties and early nineties. Among them, it is worth to draw out attention to hybrid HMM/ANN systems (see [1] for an overview), since the reported results were comparable or even slightly superior to those achieved by HMMs.

On the other hand, during the last decade, a new tool appeared in the field of machine learning that has proved to be able to cope with hard classification problems in several fields of application: the Support Vector Machines (SVMs) [2]. The SVMs are effective discriminative classifiers with several outstanding characteristics, namely: their solution is that with maximum margin; they are capable to deal with samples of a very high dimensionality; and their convergence to the minimum of the associated cost function is guaranteed.

Nevertheless, it seems clear that the application of these kernel-based machines to the ASR problem is not straightforward. In our opinion, the main difficulties to be overcome are three: 1) SVMs are originally static classifiers and have to be adapted to deal with the variability of duration of speech utterances; 2) the SVMs were originally formulated as a binary classifier while the ASR problem is multiclass; and 3) current SVM training algorithms are not able to manage the huge databases typically used in ASR. In order to cope with these difficulties, some researchers have suggested hybrid SVM/HMM systems [3, 4], that notably resemble the previous hybrid ANN/HMM systems ([5]). In this paper we comparatively describe both types of hybrid systems (SVM/ and ANN/HMM), highlighting both their common fundamentals and their special characteristics with the aim of also conducting an experimental performance comparison for both clean and noisy speech recognition tasks.

2. Hybrid systems for ASR

As a result of the difficulties found in the application of ANN to speech recognition, mostly motivated by the temporal variability of the speech instances corresponding to the same class, a variety of different architectures and novel training algorithms that combined both HMM with ANNs were proposed in the late 80's and 90's. For a comprehensive survey of these techniques see [1]. In this paper, we have focused on those that employ ANNs (and SVMs) to estimate the HMM state posterior probabilities proposed by Bourlard and Morgan ([5, 6]).

The starting point for this approach is the well-know property of using feed-forward networks such as multi-layer perceptrons (MLPs) of estimating a posteriori probabilities given two conditions:

1. There must be enough number of parameters to train a good approximation between the input and output layers and
2. A global error minimum criterion must be used to train the network (for example, mean square error or relative entropy).

The fundamental advantage of this approach is that it introduces a discriminative technique (ANN) into HMM (generative systems) while retaining their ability to handle the temporal variability.

However, this original formulation had to be modified to estimate the true emission (likelihood) probabilities by applying Bayes' rule. Therefore, the a posteriori probabilities output should be normalized by the class priors to obtain what is called *scaled likelihoods*. This fact was further reinforced by posterior theoretical developments in the search of a global ANN optimization procedure (see [7]).

Thus, systems of this type keep being locally discriminant given that the ANN was trained to estimate a posteriori probabilities. However, it can also be shown that, in theory, HMMs can be trained using local posterior probabilities as emission probabilities, resulting in models that are both locally and globally discriminant but the problem is that there are generally mismatches between the prior class probabilities implicit to the training data and the priors that are implicit to the lexical and syntactic models that are used in recognition. In spite of this,

some results imply that for certain cases the division by the priors is not necessary [7].

Among the advantages of using hybrid approaches we can cite the following (from [7]):

- Model accuracy: both MLP and SVM have more flexibility to provide more accurate acoustic models including the possibility of including different combinations of features as well as different sizes of context.
- Local discrimination ability (at a frame level).
- Parsimonious use of parameters: all the classes share the same ANN parameters.
- Complementarity: since the combination of results from standard HMM systems have been proved to provide better results

3. Experimental Setup

3.1. Database

We have used the well-known SpeechDat Spanish database [8] for the fixed telephone network. This database comprises recordings from 4000 Spanish speakers recorded at 8 kHz over the fixed PSTN using an E-1 interface, in a noiseless office environment.

In our experiments we have used a large vocabulary (more than 24000 words) continuous speech recognition database. The training set contains approximately 100 hours of voice from 3496 speakers (71000 utterances). The callers spoke 40 items whose contents are varied, comprising isolated and connected digits, natural numbers, spellings, city and company names, common applications words, phonetically rich sentences, etc. Most items are read, some are spontaneously spoken. The test set, corresponding to a connected digits task, contains approximately 2122 utterances and 19855 digits (3 hours) from 315 different speakers.

3.2. Parameterization

In our preliminary experiments we have used the classical parameterization based on 12 MFCCs (Mel-Frequency Cepstral Coefficients) plus energy, and the first and second derivatives. These MFCCs are computed every 10 ms using a temporal windows of 25 ms. Thus, the resulting feature vectors have 39 components. In this work, we have considered two different kinds of normalization for the features.

The first normalization considered was a per utterance normalization, that is, every parameter is normalized in mean and variance according to the following expression:

$$\hat{x}_i[n] = \frac{x_i[n] - \mu_f}{\sigma_f + \theta}, \quad (1)$$

where $x_i[n]$ represents the i^{th} component of the feature vector corresponding to frame n, μ_f is the estimated mean from the whole utterance, σ_f is the estimated standard deviation, and θ is a constant just to avoid numerical problems (for our experiments, we have chosen $\theta = 10$).

Thus, per utterance normalization will be more appropriate in the case of noisy environments where test and training conditions do not match. Nevertheless, when we work in a noiseless environment, the second normalization we consider provides better performance like we explain in following sections. This normalization consist of a global normalization, that is, we compute the global mean and variance for all the parameterization utterances in the training set, and finally each parameter normalized in mean and variance according to the next expression:

$$\hat{x}_i[n] = \frac{x_i[n] - \mu}{\sigma}, \quad (2)$$

where $x_i[n]$ represents the i^{th} component of the feature vector corresponding to frame n, μ is the estimated mean from all the utterances in the training set and σ is the estimated global standard deviation.

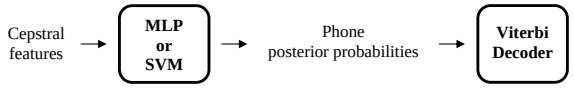


Figure 1: *The whole hybrid recognition system. First, initial phone evidences are estimated using MLPs or SVMs, then these evidences are integrated as local scores for decoding.*

3.3. Baseline experiment with HMMs

Our reference result is the recognition rate achieved by an left-to-right HMM-based recognition system. We use 18 context-dependent phones with 3 states per phone. Emission probabilities for each state were modelled by a mixture of 16 Gaussians, as described in [8].

For this paper, we have partitioned every phone into three segments and obtained a segmentation of the database by performing a forced alignment with this HMM baseline experiment considering each segment delimited by the state transitions of this system (see [4]).

3.4. Experiments with Hybrid Recognition Systems

In this work we consider two different hybrid recognition systems, an ANN/HMM system and a SVM/HMM one. Both of them use a Viterbi decoder using posterior probabilities as local scores as discussed in 2.

The whole hybrid recognition system is composed of two stages shown in Figure 1. The first stage estimates initial evidences for phones in the form of posterior probabilities using an MLP or an SVM. The second stage is a classical Viterbi decoder where we replace the likelihoods estimates provided by the reference HMM-based recognition system by the posteriors estimates obtained in the first stage.

While the reference HMM-based recognition system uses the whole training data set (71000 utterances), the hybrid systems (SVM- and ANN-based recognition systems) only use a small portion of the available training data, due to a practical limitation respect to the number of training samples that the SVM software can consider. Therefore, we have considered useful to evaluate the evolution of the accuracy of each system performing incremental tests using balanced subsets of the available training data (equal number of frames per phone, randomly selected from the whole training set), between 250 and 20000 frames per phone.

3.4.1. Experiments with SVMs

In this case, a multiclass SVM (using the *1-vs-1* approach) is used to estimate posterior probabilities for each frame using

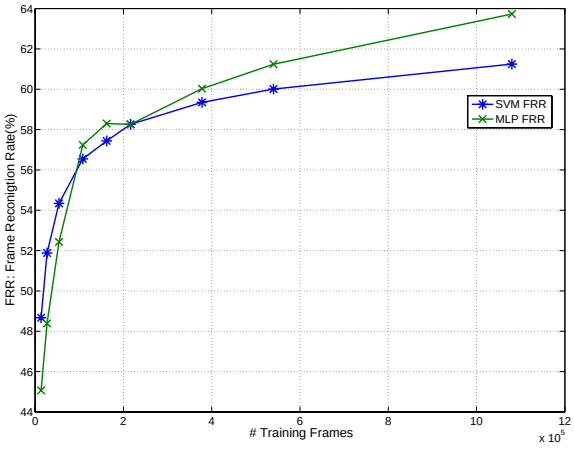


Figure 2: *Frame recognition rate of SVMs and ANNs.*

Platt's approximation ([9]). The SVM uses a RBF (Radial Basis Function) kernel whose parameter, the standard deviation σ, must be tuned by means of a cross-validation process, as well as a parameter C, which establishes a compromise between error minimization and generalization capability in the SVM. The values we have used in our experiments are $C = 256$ and $\sigma = 0.007812$ [4].

3.4.2. Experiments with ANNs

Posterior probabilities used by the Viterbi decoder are now obtained using a MLP trained on a smaller version of the training set, as we mentioned before. The MLP has one hidden layer with 1800 units. MFCC features jointly with energy, delta and acceleration features are used as inputs. There are 54 output units, each of them corresponding to a different part of phone, as we described in section 3.4. The MLP is trained using the relative entropy criterion and the back propagation factor μ was experimentally fixed at 0.14.

4. Preliminary Results and Discussion

This section is devoted to the presentation and discussion of the results obtained by the systems described in the previous section.

Preliminary experiments show a similar behaviour of both SVMs and ANNs at a frame classification level. For the first data normalization method presented in section 3, we observe little differences between SVMs and ANNs. Also, we can see in figure 2 that better results are achieved when more samples are added to the training database, up to a final recognition rate around 61% obtained for the maximum number of input samples our SVM-based system can handle (1080000, 20000 frames per phone).

We have noticed that this first normalization method presents a problem: delta and acceleration coefficients do not have unitary variance. This is due to the constant θ added to the standard deviation in (1). The value used in the experiments (10) is not comparable with the standard deviation of the data and it results in a excessive normalization. This is a problem for the SVM and ANN-based systems. For the first case, the SVM employs a RBF kernel with the same variance for all dimensions, while the training data present different variances for each component (or, at least, for the static, delta and acceleration coefficients). For the latter, this may cause to start in a point far from the solution and, as a consequence, to slow down the convergence of the algorithm. This has led us to apply a second normalization stage to the database, in order to get a unitary variance for all the components of the training data. Some experiments show an important improvement of the previous results (around 4.5%).

Preliminary experiments at word and sentence levels show results are comparable with respect to those of the standard HMM-based speech recognition system used as a baseline. These results are specially promising due to the fact that SVM and ANN-based systems are trained using a maximum of only 3.04% of the available data samples, whereas HMMs are trained using the entire database. This limit is imposed by the SVM software used in the experiments [10], which requieres to mantain the kernel matrix in memory.

In addition, as we have stated in section 2, both SVMs and ANNs provide posteriors to the Viterbi decoder, whereas what we really need and HMMs compute are likelihoods. We think that the hybrid methods might benefit from the use of likelihoods instead of posteriors [5], just by dividing them by the *a priori* probabilities.

Finally, one of the major drawbacks of current HMM-based automatic speech recognition systems is its poor robustness against noisy conditions. During the last years, several techniques aimed at increasing the performance of these systems have been presented, most of them consisting in some preprocessing of the voice signal or modifications of the parameterization stage. From previous experiments ([11]) we suspect that SVM-based systems could provide inherent robust models. Besides, as discussed in section 2, hybrid systems are more amenable for its use with different types of parameterizations that do not comply with the restrictions of independence imposed by HMM. This could result advantageous in the search of robustness.

5. Conclusions

In this paper we have performed a comparison of the accuracy of MLPs and SVMs at a frame level showing a similar performance. However, we still think there is room for improvement of the latter, specially in noisy environment conditions. The maximum margin principle used for its training can make an important difference under those conditions. There are also several issues that should be address as the possibility to incorporate more training samples, the addition of a wider context in the feature vectors, the selection of appropriate feature sets and the computation of further results at a word level.

6. References

[1] E. Trentin and M. Gori, "A survey of hybrid ANN/HMM models for automatic speech recognition," *Neurocomputing*, vol. 37, pp. 91–126, 2001.

[2] B. E. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Computational Learning Theory*, 1992, pp. 144–152. [Online]. Available: citeseer.ist.psu.edu/boser92training.html

[3] A. Ganapathiraju, J. Hamaker, and J. Picone, "Hybrid SVM/HMM architectures for speech recognition," in *Proceedings of the 2000 Speech Transcription Workshop*, vol. 4, Maryland (USA), May 2000, pp. 504–507.

[4] J. Padrell-Sendra, D. Martín-Iglesias, and F. Díaz-de-

María, "Support vector machines for continuous speech recognition," in *Proc. of the 14th European Signal Processing Conference*, 2006.

[5] H. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Norwell, MA (USA): Boston: Kluwer Academic, 1994.

[6] N. Morgan and H. Bourlard, "Continuous speech recognition: An introduction to the hybrid hmm/connectionist approach," *IEEE Signal Processing Magazine*, pp. pp. 25–42, 1995.

[7] H. Bourlard, N. Morgan, C. L. Giles, and M. Gori, *Adaptive Processing of Sequences and Data Structures. International Summer School on Neural Networks 'E.R. Caianiello'. Tutorial Lectures*. Germany; Berlin: Springer-Verlag, 1998, ch. Hybrid HMM/ANN systems for speech recognition: overview and new research directions, pp. 389–417.

[8] A. Moreno, "SpeechDat Spanish database for fixed telephone network," Technical University of Catalonia, Tech. Rep., 1997.

[9] J. C. Platt, *Advances in kernel methods: support vector learning*. Cambridge, MA (USA): MIT Press, 1999, ch. Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pp. 185–208.

[10] C. C. Chang and C. J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/cjlin/∼libsvm.

[11] R. Solera-Ureña, D. Martín-Iglesias, A. Gallardo-Antolín, C. Peláez-Moreno, and F. Díaz-de-María, "Robust asr using support vector machines," *Speech Communication (In press)*, 2007.

Towards phonetically-driven hidden Markov models: Can we incoporate phonetic landmarks in HMM-based ASR?

Guillaume Gravier, Daniel Moraru

Équipe Metiss
Irisa, Rennes, France.
http://www.irisa.fr/metiss

Abstract

Automatic speech recognition mainly relies on hidden Markov models (HMM) which make little use of phonetic knowledge. As an alternative, landmark based recognizers rely mainly on precise phonetic knowledge and exploit distinctive features. We propose a theoretical framework to combine both approaches by introducing phonetic knowledge in a non stationary HMM decoder. To demonstrate the potential of the method, we investigate how broad phonetic landmarks could be used to improve a HMM decoder by focusing the best path search. We show that, assuming error free landmark detection, every broad phonetic class brings a small improvement. The use of all the classes reduces the error rate from 22% to 14% on a broadcast news transcription task. We also experimentally validate that landmarks boundaries does not need to be detected precisely and that the algorithm is robust to non detection errors.

1. Introduction

In hidden Markov models (HMM) based speech recognition systems, the decoding process consists in compiling a graph which includes all the available sources of knowledge (language model, pronunciations, acoustic models) before finding out the best path in the graph in order to obtain the best word sequence

$$\hat{w} = \arg\max_w p(y|w)p(w) \ . \quad (1)$$

At the acoustic level, this approach relies on data-driven methods that learn from examples. Therefore, integrating explicit phonetic knowledge in such systems is difficult.

Alternately, various studies aimed at explicitly relying on phonetic knowledge to represent the speech signal for automatic speech recognition [1, 2, 3]. These approaches are most of the time based on the extraction of a set phonetic features, a.k.a. landmarks, on top of which a model, either rule based or statistical based, is build for the purpose of recognition. Phonetically-driven ASR relies on fine grain phonetic features such as onset and offset times [2] and distinctive features [1, 3]. However, in practice, automatically detecting such features might be difficult and error prone, in particular in the case of noisy signals or spontaneous speech.

This work is a preliminary study which aims at bridging these two paradigms in order to make use of explicit phonetic knowledge in the framework of HMMs. While landmark-based systems use phonetic landmarks as a feature describing the signal, the idea of our approach is to use landmarks in order to guide the search for the best path during Viterbi decoding in an HMM-based system. Hence, prior knowledge on the nature of the signal is used as *anchor points* during decoding. We will use indistinctly the two terms landmark and anchor to designate constraints on the search.

The aim of this study is twofold. The first aim is to define a theoretical framework to incorporate phonetic knowledge in HMM based systems using anchor points and to experimentally validate this approach. This framework allows for uncertainty in the landmark detection step, though this is not validated in the study as of now. The second aim is to study which landmarks effectively complements the data-driven knowledge embedded in HMM systems. We believe that detecting fine grain phonetic features is a particularly challenging problem – in spite of recent promising results on the detection of distinctive features, see *e.g.* [4, 5, 3] – while detecting broad phonetic features can be achieved with reasonably good performance [6, 7, 8]. Hence, to avoid problems related to the detection of fine grain features, we investigate if, and to what extent, broad phonetic landmarks can help.

in this paper, we first extend the Viterbi algorithm in order to incorporate prior knowledge carried out by landmarks. We then study the impact of broad phonetic landmarks in an ideal setting where landmarks are manually detected, with an emphasis on the temporal precision of the landmarks. Finally, we discuss some upcoming experiments whose results are expected to be presented at the workshop.

2. Landmark-driven Viterbi decoding

Most HMM-based systems rely on the Viterbi algorithm in order to solve Eq. (1), along with pruning techniques to keep the search tractable for large vocabularies. We briefly recall the basics of the Viterbi algorithm before extending this algorithm for the integration of phonetic anchors.

2.1. Beam-search Viterbi decoding

The Viterbi algorithm aims at finding out the best alignment path in a graph using dynamic programming (DP) on a trellis. The DP algorithm proceeds incrementally by searching for the best hypothesis reaching the state (j, t) of the trellis according to

$$S(j,t) = \max_i S(i, t-1) + \ln(a_{ij}) + \ln(p(y_t|j)) \ , \quad (2)$$

where j is the state in the decoding graph and t the frame index in the observation sequence. In Eq. (2), $\ln(a_{ij})$ denotes the weight for the transition from state i to j in the graph[1] while

[1]Note that this weight actually combines the language model and the acoustic model probabilities for cross-word transitions.

$p(y_t|j)$ denotes the likelihood of the feature vector y_t conditional to state j. Hence, $S(i,t)$ represents the score of the best partial path ending in state i at time t.

In practice, not all paths are explored in order to keep the algorithm tractable on large decoding graphs. Unlikely partial hypotheses are pruned according to the score of the best path ending at time t.

2.2. Introducing anchors

Anchors can be considered as hints on what the best path is. For example, if a landmark indicates that a portion of an utterance corresponds to a vowel, then we can constrain the best path to be consistent with this piece of information since nodes in the decoding graph are linked to phonemes. One easy way to do this is to penalize, or even prune, all the paths of the trellis which are inconsistent with the knowledge brought by the vowel landmark. Assuming confidence measures are associated with the landmarks, the penalty should be proportional to the confidence.

Formally, the above principle can be expressed using non-stationary graphs, *i.e.* graphs whose transition probabilities are dependent on time. The idea is that if a transition leading to state (i,t) of the trellis is inconsistent with the landmark knowledge, then the transition cost increases. In order to do this, we replace in (2) the transition weights $ln(a_{ij})$ by

$$\ln(a_{ij}(t)) = \ln(a_{ij}) - \lambda(t) I_j(t) \ . \quad (3)$$

$I_j(t)$ is an indicator function whose value is 0 if node j is compatible with the available anchor information and 1 otherwise. The penalization term $\lambda(t) > 0$ reflects the confidence in the anchor available at time t, if any. Hence, if no anchor is available or if a node is consistent with the anchor, then no penalization is applied. In the opposite case, we apply a penalty where the higher the confidence in the landmark, the higher the penalty. In the extreme case where landmark detection is perfect, setting $\lambda(t) = \infty$, enables to actually prune paths inconsistent with the landmarks.

In (3), one can notice that the penalty term only depends on the target state j and hence the proposed scheme is equivalent to modifying the state-conditional probability $p(y_t|j)$ to include a penalty. However, introducing the penalty at the transition level might be useful in the future to introduce phonological constraints or word-level constraints.

A by product of the proposed method is that decoding should be much faster with landmarks as adding a penalty will most likely result in inconsistent paths being pruned.

In this preliminary study, we use manually detected landmarks in order to investigate whether or not broad phonetic landmarks can help and to what extent in an ideal case. We will therefore set $\lambda(t) = \infty, \forall t$ in all the experiments described in section 4.

3. Baseline system

Before describing the experiments, we briefly present the data and baseline system used.

3.1. Corpus

Experiments are carried out on a radio broadcast news corpus in the French language. The data used is a 4 hour subset of the development data for the ESTER broadcast news rich transcription evaluation campaign [9]. The corpus mostly contains high-fidelity planned speech from professional radio speakers. Interviews, however, contain more spontaneous speech from non professional speakers, sometimes in degraded acoustic conditions.

The entire data set was labeled phonetically based on the reference orthographic transcription, using our ASR system to select pronunciation variants.

3.2. ASR system

Two reference systems were used in this study. Both systems are two-pass systems where a first pass aims at generating a word graph which is then rescored in a second pass with more sophisticated acoustic models. The two systems differ in the complexity of the acoustic models used for the word graph generation, the first system using context-independent models while word-internal context-dependent ones are used for the second system. Clearly, using landmarks to guide the decoding is more interesting when generating the word graph as it should enable better and smaller word graphs which already take into account the landmark knowledge. Therefore, the reason for comparing two systems for word graph generation is to determine to what extent phone models capture broad phonetic information.

Both transcription passes are carried out with a trigram language model. Monophone acoustic models have 114 states with 128 Gaussians per state while the word-internal triphone models have 4,019 distinct states with 32 Gaussians each. Cross-word triphones models are used for word graph rescoring, with about 6,000 distinct states and 32 Gaussians per state.

4. Broad phonetic landmarks

The experiments described in this section are performed using manually detected broad phonetic landmarks, the goal being to measure the best expected gain from the use of such landmarks. The main motivation for using this type of landmarks, as opposed to distinctive features, is that we believe that reliable and robust automatic broad phonetic landmark detectors can be build. For example, in [6, 7, 8] (to cite a few), good results are reported on the detection of nasals and vowels. Fricatives also seems relatively easy to detect using energy and zero crossing rate information. Moreover, we observed that the heap of active hypotheses in our ASR system most of time contains hypotheses corresponding to different broad phonetic classes. Though this is normal since hypotheses correspond to complete partial paths rather than to local decisions, this observation indicates that a better selection of the active hypotheses based on (locally detected) landmarks is bound to improve the results.

4.1. Landmark generation

Five broad phonetic classes are considered in this study, namely vowels, fricatives, plosives, nasal consonants and glides. Landmarks are generated from the available phonetic alignments obtained from the orthographic transcription. For each phone, a landmark corresponding to the broad phonetic class to which the phone belongs is generated, centered on the phone segment. The landmark duration is proportional to the phone segment length. In the first set of experiments, the landmark length is set to 50% of the phone segment length. We study in section 4.3 the impact of the landmark duration.

4.2. Which landmarks?

The first question to answer is what is the optimal improvement that can be obtained using each broad phonetic class separately. Results are given in table 1 for the monophone and triphone systems after the first and second pass, with each landmark type taken separately. Results using all the landmarks or a combination of vowel, plosive and fricative landmarks are also reported.

Results show a small improvement for each type of landmarks, thus clearly indicating that the transcription system is not misled by phones from a particular broad phonetic class. The best improvement is obtained with landmarks for glides, that correspond to highly transitory phones which are difficult to model, in particular because of co-articulation effects. More surprisingly, vowel landmarks yield a small but significant improvement, in spite of the fact that the phone models used in the ASR system do little confusions between vowels and other phones. This result is due to the fact that the DP maximization not only depends on the local state-conditional probabilities but also on the score of the entire path resulting in an hypothesis. In other words, even if the local probabilities $p(y_t|i)$ are much better for states corresponding to a vowel than for states corresponding to some other class, some paths, incompatible with the knowledge of a vowel landmark, might get a good cumulated score and are therefore kept in the heap of active hypotheses. Using the landmark-driven version of the Viterbi algorithm actually remove such paths from the search space, thus explaining the gain obtained with vowel landmarks.

Clearly, using all the available landmarks strongly improves the WER for both systems, the improvement being unsurprisingly better for the monophone-based system. One interesting point to note is that, when using all the landmarks, the two systems exhibit comparable levels of performance, with a slight advantage for the monophone system. This advantage is due to the fact that the word graph generated with the monophone system contains more sentence hypotheses than the one generated with the triphone system, though both graphs have roughly the same density. A last point worth noting is the rather good performance obtained after the first pass using the monophone system. This result suggest that combining landmark-driven decoding with fairly simple acoustic models can provide good transcriptions with a limited amount of computation. Indeed, the average number of active hypotheses, and hence the decoding time, is divided by a factor of four when using landmarks.

In a practical setting, the reliable detection and segmentation of a signal into broad phonetic classes is somewhat unrealistic, the detection of nasals and glides being a rather difficult problem. However, detecting vowels, plosives and fricatives seems feasible with a great accuracy. We therefore report results using only landmarks from those three classes (VPF results in table 1). Using such landmarks, a nice performance gain can still be expected, in particular with a monophone-based word graph generation.

These results show the optimal gain that can be obtained using broad phonetic landmarks as anchors in a Viterbi decoding, thus justifying further work on landmark detection.

4.3. Landmark precision

Two questions arise regarding the precision of the landmark detection step. The first question is to determine whether a precise detection of the landmark boundaries is necessary or not. The second question concerns the robustness to detection errors of the proposed algorithm.

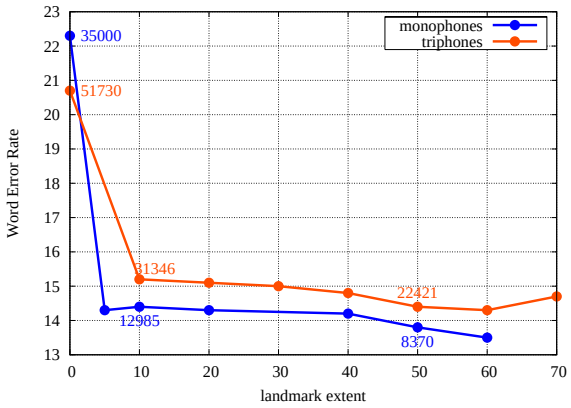


Figure 1: *WER (in %) as a function of the landmarks length, using all landmarks. The landmark length is defined as a fraction of the length of the phone which generated the landmark. Figures reported on the graph correspond to the average size of the active hypotheses heap during word graph generation.*

4.3.1. *Temporal precision*

Figure 1 shows the word error rate for the two systems as a function of the landmark extent, where the extent is defined as the relative duration with respect to the phone used to generate the landmark. An extent of 10 therefore means that the duration of a landmark is 0.1 times that of the corresponding phone. All the landmarks are considered in these experiments. Unsurprisingly, the longer the landmarks, the better the transcription. It was also observed that longer landmarks reduce the search space and yield smaller, yet better, word graphs. In spite of this, most of the improvement comes from the fact that landmarks are introduced, no matter their extent. Indeed, with a landmark extent of only 5%, the word error rate decreases from 22.3% to 14.3% with the monophone system. When increasing the landmark extent to 50%, the gain is marginal, with a word error rate of 13.9%. Note that with an extent of 5%, the total duration of landmarks corresponds to 4.4% of the total duration of the signal, and therefore landmark-based pruning of the hypotheses heap happens only for 4.4% of the frames. Similar conclusions were obtained using only the vowel landmarks. This is a particularly interesting result as it demonstrates that landmark boundaries do not need to be detected precisely. Reliably detecting landmarks on some very short portion of the signal (one or two frames) is sufficient to drive a Viterbi decoder with those landmarks.

4.3.2. *Robustness to non-detection errors*

In the absence of confidence measures, landmark-driven Viterbi is highly sensitive to detection errors. Clearly, false alarms, *i.e.* insertion and confusion errors, have detrimental effects on the system. However, miss detection errors are less disastrous. Therefore, automatic broad phonetic landmark detection systems should be designed to have as low as possible a false alarm rate. However, lower false alarm rates unfortunately imply higher miss detection rates. We tested the robustness of our landmark-driven decoder by simulating miss detection errors at various rates, assuming a uniform distribution of the errors across the five broad phonetic classes. Results show that the word error rate is a linear function of the miss detection rate.

Table 1: *Word error rate (in %) after each pass for the monophone and word-internal triphone systems, as a function of the landmarks used. The landmark ratio indicates the amount of signal (in %) for which a landmark is available.*

landmarks		none	all	VPF	vow.	plo.	fri.	nas.	gli.
landmark ratio			43.6	34.6	18.3	9.0	7.3	2.8	6.2
monophones	passe 1	29.2	15.3	21.7	26.6	26.5	27.5	27.8	25.1
	passe 2	22.3	13.9	17.6	21.2	20.7	21.0	21.5	20.1
triphones	passe 1	27.3	19.6	23.9	27.0	26.3	26.0	26.4	24.9
	passe 2	21.3	15.0	18.2	20.7	20.4	20.3	20.7	19.6

For example, with the monophone system, the word error rate is 17.9% (resp. 15.8%) for a miss detection error rate of 50% (resp. 25%).

5. Discussion

The preliminary experiments reported in this summary are encouraging and prove that integrating broad phonetic landmarks in a HMM-based system can drastically improve the performance, assuming landmarks can be detected reliably. These results also validate the proposed paradigm for the integration of various sources of knowledge: phonetic knowledge via landmarks and data-driven knowledge acquired by the HMMs. However, results are reported in an ideal laboratory setting where landmark detection is perfect. The first step is therefore to work on robust detectors of broad phonetic landmarks, at least for vowels, plosives and fricatives, in order to validate the proposed paradigm in practical conditions.

A naive method for broad phonetic landmark detection was tested, based on broad phonetic class HMMs along with a trigram language model. For each of the five broad phonetic class, a context-independent left-right model with 3 states and 32 Gaussians per states was estimated on the training data of the ESTER corpus. These models were then used for broad phonetic segmentation with a trigram language model, resulting in an accuracy of 76.6[2]. Assuming landmarks extracted from this broad phonetic segmentation with a landmark extent of 20% of the segment length, the amount of landmark time that is not correctly labeled is 10.8%, which did not prove sufficiently low to help the ASR system. Indeed, we used the landmarks associated with sufficiently long segments as anchors to prune inconsistent hypotheses in our system, assuming long segments are more reliable than short ones. On a 1 hour show, the baseline word error rate of 22.3% increased to 23.2% with vowel, plosive and fricative landmarks and 24.3% considering all the landmarks. However, the segmentation system is naive in the sense that it relies on the same features and techniques than the ASR system and therefore does not bring any new information. It seems clear that a better broad phonetic segmentation system, based on different features, can be devised. Moreover, *segmentation* may not be the best strategy for landmark detection and techniques that differs from the HMM framework (*e.g.* MLP, SVM) should be used for the *detection* of broad phonetic landmarks. Results with more robust landmark detectors and using confidence measures will be presented at the workshop.

Finally, let us conclude this discussion with two remarks. First, we believe that mixing the landmark paradigm with data-driven methods offers a great potential to tackle the problem of robustness. In this sense, broad phonetic landmarks seems a reasonable choice to achieve robustness. In particular, we think that human perception might actually follow a similar scheme as the one presented here, where landmarks are used to disambiguate sounds and words. Second, we would like to stress that the framework defined in this paper for the integration of phonetic knowledge in a HMM based system is not limited to speech recognition with landmarks. The framework offers a way to integrate knowledge in a DP algorithm in a general way and has many application fields such as multimodal fusion or audiovisual speech recognition.

[2]Most of the errors are due to the glides while vowels are well detected. Surprisingly, fricatives are not very well detected.

6. Acknowledgments

This work was partially financed in the framework of the Action Concertée Incitative Masses de Données Demi-Ton – Multimodal Description for the Automatic Structuring of TV Streams.

7. References

[1] S. A. Liu, "Landmark detection for distinctive feature-based speech recognition," Ph.D. dissertation, Massachusetts Institute of Technology, 1995.

[2] A. Juneja, "Speech recognition based on phonetic features and acoustic landmarks," Ph.D. dissertation, University of Maryland, 2004.

[3] *Landmark-based speech recognition: report of the 2004 John Hopkins Summer Workshop*. John Hopkins University, Center for Language and Speech Processing, 2005.

[4] E. McDermott and T. Hazen, "Minimum classification error training of landmark models for real-time continuous speech recognition," in *Proc. IEEE Intl. Conf. Acoust., Speech, Signal Processing*, vol. 1, 2004, pp. 937 – 940.

[5] J. G. K. Schutte, "Robust detection of sonorant landmarks," in *Eurospean Conf. on Speech Communication and Technology – Interspeech*, 2005, pp. 1005 – 1008.

[6] M. Chen, "Nasal landmark detection," in *Intl. Conf. Speech and Language Processing*, 2000, pp. 636–639.

[7] A. Howitt, "Vowel landmark detection," in *Intl. Conf. Speech and Language Processing*, 2000.

[8] J. Li and C.-H. Lee, "On designing and evaluating speech event detectors," in *European Conference on Speech Communication and Technology – Interspeech*, 2006, pp. 3365–3368.

[9] S. Galliano, E. Geoffrois, J.-F. Bonastre, G. Gravier, D. Mostefa, and K. Choukri, "Corpus description of the ESTER evaluation campaign for the rich transcription of french broadcast news," in *Language Resources and Evaluation Conference*, 2006.

A HYBRID GENETIC-NEURAL FRONT-END EXTENSION FOR ROBUST SPEECH RECOGNITION OVER TELEPHONE LINES

Sid-Ahmed Selouani[*], *Habib Hamam*[**], *Douglas O'Shaughnessy*[***]

[*]Université de Moncton, Campus de Shippagan, Canada
[**]Université de Moncton, Campus de Moncton, Canada
[***]INRS-Énergie-Matériaux-Télécommunications, Canada
selouani@umcs.ca, hamamh@umoncton.ca, dougo@emt.inrs.ca

Abstract

This paper presents a hybrid technique combining the Karhonen-Loeve Transform (KLT), the Multilayer Perceptron (MLP) and Genetic Algorithms (GAs) to obtain less-variant Mel-frequency parameters. The advantages of such an approach are that the robustness can be reached without modifying the recognition system, and that neither assumption nor estimation of the noise are required. To evaluate the effectiveness of the proposed approach, an extensive set of continuous speech recognition experiments are carried out by using the NTIMIT telephone speech database. The results show that the proposed approach outperforms the baseline and conventional systems.

1. Introduction

Adaptation to the environment changes and artifacts remains one of the most challenging problems for the Continuous Speech Recognition (CSR) systems. The principle of CSR methods consists of building speech sound models based on large speech corpora that attempt to include common sources of variability that may occur in practice. Nevertheless, not all situations and contexts can be exhaustively covered. As speech and language technologies are being transferred to real applications, the need for greater robustness in recognition technology becomes more apparent when speech is transmitted over telephone lines, when the signal-to-noise ratio (SNR) is extremely low, and more generally, when adverse conditions and/or unseen situations are encountered. To cope with these adverse conditions and to achieve noise robustness, different approaches have been studied. Two major approaches have emerged. The first approach consists of preprocessing the corrupted speech input signal prior to the pattern matching in an attempt to enhance the SNR. The second approach attempts to establish a compensation method that modifies the pattern matching itself to account for the effects of noise. Methods in this approach include noise masking, the use of robust distance measures, and HMM decomposition. For more details see [5].

As an alternative approach, we propose a new enhancement scheme based on the combination of subspace filtering, the Multilayer Perceptron (MLP) and Genetic Algorithms (GAs) to obtain less-variant Mel-frequency parameters. The enhanced parameters are expected to be insensitive to the degradation of speech signals due to telephone-channel degradation. The main advantages of such an approach over the compensation method are that the robustness can be reached without modifying the recognition system, and without requiring assumption or estimation of the noise.

This paper is organized as follows. In section 2, we describe the basis of the signal subspace approach, namely the Karhonen-Loeve Transform (KLT) and the extension we proposed to enable the use of the technique in the Mel-frequency space. In section 3, we briefly describe the principle of MLP-based enhacement method. Then, we proceed in section 4 with the description of the evolutionary-based paradigm that we introduced to perform noise reduction. In section 5, we evaluate the hybrid MLP-KLT-GA-based front-end technique in the context of telephone speech. Finally, in section 6, we conclude and discuss our results.

2. Signal and Mel-frequency subspace filtering

The principle of the signal subspace techniques is based on the construction of an orthonormal set of axes. These axes point in the directions of maximum variance, thus forming a representational basis that projects on the direction of maximum variability. Applied in the context of noise reduction, these axes enable decomposing the space of the noisy signal into a signal-plus-noise subspace and a noise subspace. The enhancement is performed by removing the noise subspace and estimating the clean signal from the remaining signal space. The decomposition of the space into two subspaces can be performed by using KLT (eigendecomposition). Let $\mathbf{x} = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}]^\mathbf{T}$ be an N-dimensional noisy observation vector which can be written as the sum of an additive noise distortion vector $\mathbf{w}$ and the vector of clean speech samples $\mathbf{s}$. The noise is assumed to be uncorrelated with the clean speech. Further, let $\mathbf{R_x}$, $\mathbf{R_s}$, and $\mathbf{R_w}$ be the covariance matrices from $\mathbf{x}$, $\mathbf{s}$, and $\mathbf{w}$ respectively. The eigendecomposition of $\mathbf{R_s}$ is given by $\mathbf{R_s} = \mathbf{Q}\mathbf{\Lambda_s}\mathbf{Q^T}$ where $\mathbf{\Lambda_s}$= diag$(\lambda_{s1}, \lambda_{s2}, ..., \lambda_{sN})$ is the diagonal matrix of eigenvalues given in a decreasing order. The eigenvector matrix $\mathbf{Q}$ of the clean speech covariance matrix is identical to that of the noise. Major signal subspace techniques assume the noise to be white with $\mathbf{R_w} = \sigma_\mathbf{w}^\mathbf{2}\mathbf{I}$ where $\sigma_\mathbf{w}^\mathbf{2}$ is the noise variance and $\mathbf{I}$ the identity matrix. Thus, the eigendecomposition of $\mathbf{R_x}$ is given by: $\mathbf{R_s} = \mathbf{Q}(\mathbf{\Lambda_s} + \sigma_\mathbf{w}^\mathbf{2}\mathbf{I})\mathbf{Q^T}$. The enhancement is performed by assuming that the clean speech is concentrated in an $r < N$ dimensional subspace, the so-called signal subspace, whereas the noise occupies the $N - r$ dimensional observation space. Then the noise reduction is obtained by considering only the signal subspace in the reconstruction of the enhanced signal. Mathematically it consists of finding a linear estimate of $\mathbf{s}$ given by $\hat{\mathbf{s}} = \mathbf{F}\mathbf{x} = \mathbf{F}\mathbf{s} + \mathbf{F}\mathbf{w}$ where $\mathbf{F}$ is the enhancement filter. This filter matrix $\mathbf{F}$ can be written as follows:

$\mathbf{F} = \mathbf{Q_r G_r Q_r^T}$ in which the diagonal matrix $\mathbf{G_r}$ contains the weighting factors g_i with $i = 1, ..., r$, for the eigenvalues of the noisy speech. Perceptually meaningful weighting functions exist to generate g_i. These functions are empirically guided in order to constitute an alternative choice for g_i, which results in a more or less aggressive noise suppression, depending on the SNR. In [1], the linear estimation of the clean vector is performed using two perceptually meaningful weighting functions. The first function is given by :

$$g_i = \Big[\frac{\lambda_{xi}}{\lambda_{xi} + \sigma_w^2}\Big]^{\gamma}, \quad i = 1, ..., r, \tag{1}$$

where $\gamma \geq 1$.

The second function constitutes an alternative choice for g_i which results in a more aggressive noise suppression:

$$g_i = \exp\Big\{\frac{-\nu\sigma_w^2}{\lambda_{xi}}\Big\}. \quad i = 1, ..., r, \tag{2}$$

The value of the parameter ν is to be fixed experimentally.

Instead of dealing with the speech signal, we chose to use the noisy Mel-Frequency Cepstral Coefficients (MFCC) vector $\mathbf{C}^{'}$ as well. The reason is that these parameters are suited to speech recognition due to the advantage that one can derive from them a set of parameters which are invariant to any fixed frequency-response distortion introduced by either the adverse environments or the transmission channels [6]. The main advantage of the approach proposed here is that we do not need to define weighting functions. In this approach, the filter matrix $\mathbf{F}$ can be written as follows: $\mathbf{F_{gen}} = \mathbf{Q G_{gen} Q^T}$ in which the diagonal matrix $\mathbf{G_{gen}}$ contains now weighting factors optimized using genetic operators. Optimization is reached when the Euclidian distance between the noisy and clean MFCCs is minimized. To improve the enhancement of noisy MFFCs, we introduce a preprocessing level which uses the MLP. As depicted in Figure 1, the noisy (MFCC) vectors $\mathbf{C}^{'}$ are first enhanced by MLP. Then, a KLT is performed on the output of MLP, denoted by $\hat{\mathbf{C}}$. Finally, the space of feature representation is reconstructed by using the eigenvectors weighted by the optimal factors of the $\mathbf{G_{gen}}$ matrix.

3. MLP-based enhancement preprocessing of the KLT

Numerous approaches were proposed in the literature to incorporate acoustic features estimated by the MLP under noisy conditions [6] [12]. The connectionist approaches offer inherent nonlinear capabilities as well as easy training from pairs of corresponding noisy and noise-free signal frames. Because the front end is very modular, the MLP estimator can be introduced at different stages in the feature processing stream. For instance, the MLP can estimate robust filterbank log-energies that will then be processed with the traditional Distrete Cosine Transform to get the unnormalized cepstral coefficients. Alternatively, we can estimate the cepstral features directly with an MLP. Yet another possibility is to estimate filterbank log-energies but to measure the feature distortion at the cepstrum level and optimize the filterbank log-energy estimator accordingly [12]. The fact that the noise and the speech signal are combined in a nonlinear way in the cepstral domain led us to

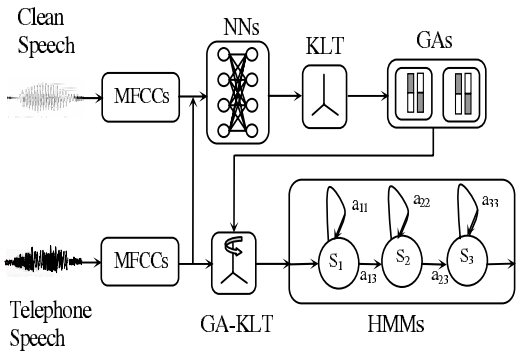


Figure 1: *The proposed MLP-KLT-GA-based CSR system.*

choose the second alternative described above. MLP can approximate the required nonlinear function to some extent [6]. Hence, the input of the MLP is the noisy MFCC vector $\mathbf{C}^{'}$, while the actual response of the network $\hat{\mathbf{C}}$ is computed during a training phase by using a convergence algorithm to update the weight vector in a manner to minimize the error between the output $\hat{\mathbf{C}}$ and the desired clean cepstrum value $\mathbf{C}$. The weights of this network are calculated during a training phase with a back-propagation training algorithm using a mean square error criterion.

The noisy 13-dimensional vector (12 MFCCs + energy) is fed to an MLP network in order to reduce the noise effects on this vector. This first-level pre-processing does not require any knowledge about the nature of the corrupting noisy signal, which permits dealing with any kind of noise. Moreover, using this enhancement technique we avoid the noise estimation process that requires a speech/non-speech pre-classification, which may be not accurate enough for low SNRs. It is worth noting that this technique is less complex than many other enhancement techniques that requires either modeling or compensating for the noise.

Once the enhanced vector is obtained, it is fed to the KLT-GA module, which represents the second enhancement level. This module refines the enhanced vector by projecting its components in the subspace generated by a genetically weighted version of the eigenvectors of the clean signal. The motivation behind the use of a second level of enhancement after using the MLP network is to compensate for the limited power of the MLP network for enhancement outside the training space [6].

4. Hybrid MLP-KLT-GA speech front-end

The KLT processing on the MLP-enhanced noisy vectors $\hat{\mathbf{C}}$ gives the diagonal matrix $\mathbf{G_r}$ containing the weighting factors g_i with $i = 1, ..., r$. In the classical subspace filtering approaches, a key issue is to determine the rank r from which the high order components (those who are supposed to contain the noise are removed). In the evolutionary-based method we propose, all components are used in the optimization process. Only the performance criterion will determine the final components that are retained to perform the reconstruction of the space of enhanced features.

The evolution process starts with the creation of a population of the weight factors, g_i with $i = 1, ..., N$, which represent the individuals. The individuals evolve through many generations in a pool where genetic operators are applied [4].

Some of these individuals are selected to reproduce according to their performance. The individuals' evaluation is performed through the use of an objective function. When the fittest individual (best set of weights) is obtained, it is then use in the test phase to project the noisy data. Genetically modified MFCCs, their first and second derivatives, are finally used as enhanced features for the recognition process. As mentioned earlier, the problem of determining optimal *r* is not needed, since the GA considers the complete space dimension *N*.

4.1. Initialization, termination and solution representation

A solution representation is needed to describe each individual in the population. For our application, the useful representation of an individual for function optimization involves genes or variables from an alphabet of floating point numbers with values within the variables' upper and lower bounds, noted (a_i, b_i) respectively. Concerning the initialization of the pool, the ideal zero-knowledge assumption is to start with a population of completely random values of weights. These values follow an uniform distribution within the upper and lower boundaries. The evolution process is terminated when a certain number of maximum generations is reached. This number corresponds to a convergence of the objective function.

4.2. Selection function

A common selection method assigns a probability of selection, P_j, to each individual, *j*, based on its objective function value. Various methods exist to assign probabilities to individuals. In our application, the normalized geometric ranking is used [7]. This method defines P_j for each individual by:

$$P_j = \frac{q(1-q)^{s-1}}{1-(1-q)^P}, \tag{3}$$

where *q* is the probability of selecting the best individual, *s* is the rank of the individual (1 is the rank of the best), and *P* is the population size.

4.3. Crossover

In order to avoid the extension of the exploration domain of the best solution, a simple crossover operator can be used [7]. It generates a random number l from a uniform distribution and does an exchange of the genes of the parents (*X* and *Y*) on the offspring genes (*X'* and *Y'*). It can be expressed by the following equations:

$$\begin{cases} X' = lX + (1-l)Y \\ Y' = (1-l)X + lY. \end{cases} \tag{4}$$

4.4. Mutation

The principle of the non-uniform mutation consists of randomly selecting one component, x_k, of an individual X, and setting it equal to a non-uniform random number, x'_k:

$$x'_k = \begin{cases} x_k + (b_k - x_k)f(Gen) & \text{if } u_1 < 0.5 \\ x_k - (a_k + x_k)f(Gen) & \text{if } u_1 \geq 0.5, \end{cases} \tag{5}$$

where the function *f(Gen)* is given by:

$$f(Gen) = (u_2(1 - \frac{Gen}{Gen_{max}}))^t, \tag{6}$$

where u_1,u_2 are uniform random numbers in the range (0,1), *t* is a shape parameter, *Gen* is the current generation and Gen_{max} is the maximum number of generations. The multi-non-uniform mutation generalizes the application of the non-uniform mutation operator to all the components of the parent *X*.

4.5. Objective function

The GA must search all the axes generated by the KLT of the MEL-frequency space to find the closest to the clean MFCCs. Thus, evolution is driven by a fitness function defined in terms of a distance measure between noisy MFCCs pre-processed by MLP and projected on a given individual (axis), and the clean MFCCs. The fittest individual is the axis which corresponds to the minimum of that distance. The distance function applied to cepstral (or other voice representations) refers to *spectral distortion measures* and represents the cost in a classification system of speech frames. For two vectors $\mathbf{C}$ and $\hat{\mathbf{C}}$ representing two frames, each with *N* components, the geometric distance is defined as:

$$d(\mathbf{C}, \hat{\mathbf{C}}) = (\sum_{k=1}^{N} (\mathbf{C_k} - \hat{\mathbf{C}}_\mathbf{k})^l)^{1/l}. \tag{7}$$

For simplicity, the Euclidian distance is considered ($l = 2$), which turned out to be a valuable measure for both clean and noisy speech. Note that $-d(\mathbf{C}, \hat{\mathbf{C}})$ is used as a distance measure because the evaluation function must be maximized.

5. Experiments and Results

Extensive experimental studies were carried out to characterize the impairment induced by telephone networks [3]. When speech is recorded through telephone lines, a reduction in the analysis bandwidth yields a higher recognition error, particularly when the system is trained with high-quality speech and tested using simulated telephone speech [9].

In our experiments, the training set composed of the $dr1$ and $dr2$ subdirectories of the TIMIT database, described in [2], was used to train a set of clean speech models. The speech recognition system used the $dr1$ subdirectory of NTIMIT as test set [2]. HTK the HMM-based speech recognition system described in [11] has been used throughout all experiments. We compared three systems: the KLT-based system as detailed in [10], the new MLP-KLT-GA-based CSR system and the baseline HMM-based system which uses a MFCC+first and second derivatives front-end denoted: MFCC_D_A. The architecture of the MLP network consists of three layers. The input layer consists of 13 neurons, while the hidden layer and the output layer consists of 26 and 13 neurons, respectively. The input to the network is the noisy 12-dimensional MFCC vector in addition to the energy. The weights of this network are calculated during a training phase with a back-propagation algorithm with a learning rate equal to 0.25 and a momentum coefficient equal to 0.09. The obtained weight values are then used during the recognition process to reduce the noise in the enhanced obtained vector that is incorporated into the KLT-GA module. To control the run behaviour of a genetic algorithm, a number of parameter values must be defined. The initial population is composed of 250 individuals and was created by duplicating the elements of the weighting matrix. The genetic algorithm was halted after 500 generations. The percentages of crossover rate and mutation rate are fixed respectively at 28% and 4%. The number of

	$\%\epsilon_{Sub}$	$\%\epsilon_{Del}$	$\%\epsilon_{Ins}$	$\%C_{Wrd}$
MFCC_D_A (1)	82.71	4.27	33.44	13.02
KLT-(1)	77.05	5.11	30.04	17.84
MLP-KLT-GA-(1)	52.15	5.07	21.36	**43.22**

[a] $\%C_{Wrd}$ using 1-mixture triphone models.

	$\%\epsilon_{Sub}$	$\%\epsilon_{Del}$	$\%\epsilon_{Ins}$	$\%C_{Wrd}$
MFCC_D_A (1)	81.25	3.44	38.44	15.31
KLT-(1)	78.11	3.81	48.89	18.08
MLP-KLT-GA-(1)	49.78	3.68	49.40	**46.48**

[b] $\%C_{Wrd}$ using 2-mixture triphone models.

	$\%\epsilon_{Sub}$	$\%\epsilon_{Del}$	$\%\epsilon_{Ins}$	$\%C_{Wrd}$
MFCC_D_A (1)	78.85	3.75	38.23	17.40
KLT-(1)	76.27	4.88	39.54	18.85
MLP-KLT-GA-(1)	50.95	3.58	22.98	**49.10**

[c] $\%C_{Wrd}$ using 4-mixture triphone models.

	$\%\epsilon_{Sub}$	$\%\epsilon_{Del}$	$\%\epsilon_{Ins}$	$\%C_{Wrd}$
MFCC_D_A (1)	78.02	3.96	40.83	18.02
KLT-(1)	77.36	5.37	34.62	17.32
MLP-KLT-GA-(1)	47.85	5.86	25.39	**50.48**

[d] $\%C_{Wrd}$ using 8-mixture triphone models.

Table 1: *Percentages of word recognition rate ($\%C_{Wrd}$), insertion rate ($\%\epsilon_{Ins}$), deletion rate ($\%\epsilon_{Del}$), and substitution rate ($\%\epsilon_{Sub}$) of the MFCC_D_A_ (denoted (1)), KLT-MFCC_D_A, and MLP-KLT-GA-MFCC_D_A CSR systems using (a) 1-mixture, (b) 2-mixture, (c) 4-mixture and (d) 8-mixture* **tri-phone** *models. (Best rates are highlighted in boldface.)*

total runs was fixed at 70. After the GA processing, the MFCCs static vectors are then expanded to produce a 39-dimensional (static+dynamic) vector upon which the hidden Markov models (HMMs), that model the speech subword units, were trained.

We found through experiments that using the MLP-KLT-GA as a pre-processing approach to enhance the MFCCs that were used for recognition with N-mixture Gaussian HMMs for N=1, 2, 4 and 8, using tri-phone models, leads to an important improvement in the accuracy of the word recognition rate. A correct rate of 50.48% is reached by the MLP-KLT-GA-MFCC_D_A-based CSR system when the baseline and the KLT-baseline systems achieve 18.02% and 17.32% respectively. This represents an improvement of more than 32% comparatively to the baseline system when the 8-mixture tri-phone model is used. Expanding to more than 8 mixtures did not improve the performance. The results in Table 1 show also that substitution and insertion errors are considerably reduced when the hybrid neural-evolutionary approach is included, leading to more effectiveness to the CSR system.

6. Conclusion

In this paper, a hybrid genetic-neural front-end was proposed to improve speech recognition over telephone lines. It is based on an MLP-KLT-GA hybrid enhancement scheme which aims to obtain less-variant MFCC parameters under telephone-channel degradation. Experiments show that the use of the proposed robust front-end processing increases the recognition rate by 32% when *dr1* and *dr2* TIMIT directories are used for the training and *dr1* directory of NTIMIT for the test. This indicates that both subspace filtering and GA-based optimization gained from the use of MLP as pre-processing. It is worthy of noting that the neural-evolutionary-based technique is less complex than many other enhancement techniques, which need to either model or compensate for the noise. For further work, many other directions remain open. Present goals include the improvement of the objective function in order to perform the online adaptation of the HMM-based CSR system when it faces new and unseen contexts and environments.

7. References

[1] Ephraim Y., and Van Trees H. L., "A signal subspace approach for Speech Enhancement", IEEE Transactions on Speech and Audio Processing, 3(4), pp. 251–266, 1995.

[2] Fisher W. M., Dodington G. R., and Goudie-Marshall K. M., "The DARPA Speech Recognition Research Database: Specification and Status", Proc. DARPA Workshop on Speech Recognition, pp. 93–99, 1986.

[3] Gaylor W. D., "Telephone voice transmission. standards and measurements", Prentice Hall, Englewood Clifs, N.J. 1989.

[4] Goldberg D. E., "Genetic algorithms in search, optimization and machine learning", Addison-Wesley Publishing, 1989.

[5] Gong Y., "Speech Recognition in Noisy Environments: A survey", Speech Communication, 16, pp. 261-291, 1995.

[6] Haverinen H., Salmela P., Hakkinen J., Lehtokangas M., and Saarinen J., "MLP Network for Enhancement of Noisy MFCC Vectors", Proc. Eurospeech, pp. 2371-2374, 1999.

[7] Houk C. R., Joines J. A. and Kay M. G., "A Genetic Algorithm for function optimization: a matlab implementation", North Carolina University-NCSU-IE, TR 95-09, 1995.

[8] Jankowski C., Kalyanswamy A., Basson S. and Spitz J., "NTIMIT: A phonetically balanced continuous speech, telephone bandwidth speech database", Proc. IEEE-ICASSP, Vol.1, pp.109–112, 1990.

[9] Moreno P. J., and Stern R., "Sources of degradation of speech recognition in the telephone network", Proc. IEEE-ICASSP, Vol.1, pp. 109-112, 1994.

[10] Selouani S.-A., and O'Shaughnessy D., "Robustness of speech recognition using genetic algorithms and a Mel-cepstral subspace approach", Proc. IEEE-ICASSP, Vol.I, pp. 201–204, 2004.

[11] Speech Group, Cambridge University , "The HTK Book (Version 3.4)", Cambridge University Group, 2006.

[12] Weintraub M., and Beaufays F., "Increased Robustness of Noisy Speech Features Using Neural Networks", Workshop on Robust Methods for Speech Recognition in Adverse Conditions, Tampere, Finland, May 1999.

Estimating the Stability and Dispersion of the Biometric Glottal Fingerprint in Continuous Speech

P. Gómez, A. Álvarez, L. M. Mazaira, R. Fernández, V. Rodellar

Grupo de Informática Aplicada al Procesado de Señal e Imagen
Universidad Politécnica de Madrid, Campus de Montegancedo, s/n, Boadilla del Monte
E-28660 Madrid, Spain
pedro@pino.datsi.fi.upm.es

Abstract

The speaker's biometric voice fingerprint may be derived from voice as a whole, or from the vocal tract and glottal signals, after separation by inverse filtering. This last approach has been used by the authors in early work, where it has been shown that the biometric fingerprint obtained from the glottal source or related speech residuals gives a good description of the speaker's identity and meta-information, as gender or age. In the present work a new technique is proposed based on the accurate estimation of the glottal residual by adaptive removal of the vocal tract, and the detection of the glottal spectral singularities in continuous speech. Results on a reduced database of speakers demonstrate that the biometric fingerprint estimation is robust, and shows low intra-speaker variability, which makes it a useful tool for speaker identification as well as for pathology detection, and other fields related with speech characterization.

1. Introduction

In previous work it has been shown that the biometric fingerprint obtained from the glottal source or related speech residuals after careful removal of the vocal tract function by inverse filtering [1][2] gives a good description of the speaker's identity and meta-information, as gender. The main inconvenience in using this technique was the requirement of using phonation cycles for the estimation of the fingerprint parameters, and its frame-based stationary nature. Besides, the variability of the estimates was strongly conditioned by the glottal gesture (tension, pitch extent and radiation – chest, mouth or head). To solve these problems a new technique is proposed, based on the accurate estimation of the glottal residual by adaptive removal of the vocal tract, and the detection of the glottal spectral singularities using lateral inhibition. The paper is organized as follows: an overview of the adaptive estimation of the glottal source and the vocal tract is briefly summarized. The glottal residual (glottal pulse, or first derivative of the glottal source) and the glottal source power spectral densities obtained by FFT in succesive sliding windows are scanned to detect their spectral singularities (maxima and minima) as these may be shown to be strongly related to vocal fold biomechanics [5]. Estimates from male and female voice show that these singularities are gender-specific. Based on the estimations from a wide data base of 100 subjects, it may be shown that specific parameters present in the glottal fingerprint may be used for gender classification [7]. The extension of the present results for age and gender characterization following well established research based on formant estimates [11] is foreseen.

2. Glottal Source adaptive estimation

The key for the accurate estimation of the glottal source is to obtain a good estimation of the vocal tract transfer function, and vice-versa. Traditionally it has been considered that the glottal source has a power spectral density of *1/f*. This assumption, being acceptable as far as its spectral envelope is concerned, hiddens the fact that the glottal signals have spectral signatures of their own, on which the fingerprint of vocal fold biomechanics can be found and used for specific applications, as is the biometrical description of the speaker or pathology detection [4].

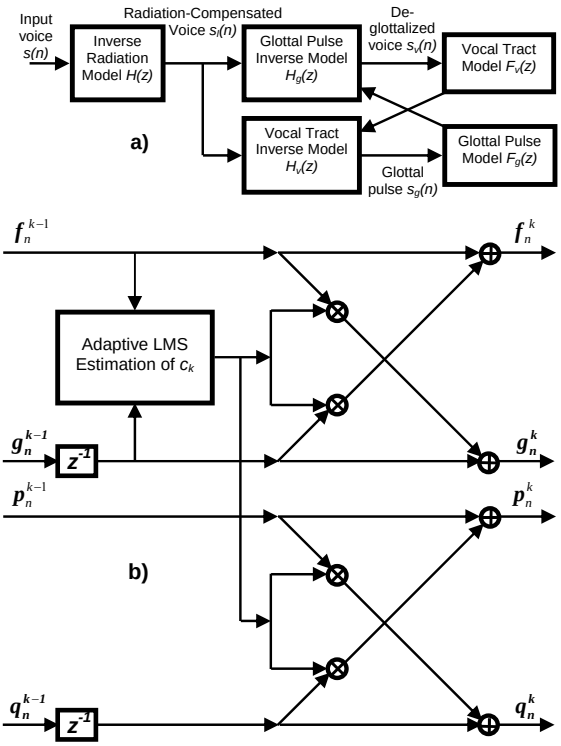


Figure 1. *a) Iterative estimation of the vocal tract transfer function $F_v(z)$ and the glottal pulse residual $s_g(n)$. b) Paired adaptive-fixed lattice section to implement parallel function estimation and removal.*

The iteration is based on the successive application of the following loop as shown in Figure 1.a:

- Estimate the inverse glottal pulse model $F_g(z)$ from input voice using an *order-2* adaptive lattice (upper lattice in Figure 1.b).
- Remove the glottal pulse from input voice using a paired fixed lattice (lower lattice in Figure 1.b) fed with the reflection coefficients obtained by the adaptive lattice for the glottal function. The resulting trace $s_v(n)$ will keep the vocal tract information but the glottal information will be diminished.
- Estimate the vocal tract transfer function $F_v(z)$ from this last trace using another adaptive lattice (typically of order *20-30*), similar to Figure 1.b.
- Remove the vocal tract transfer function from input voice using a fixed lattice (lower lattice in Figure 1.b) fed with the reflection coefficients obtained by the adaptive lattice for the vocal tract function. The resulting trace $s_g(n)$ will keep the glottal information but the vocal tract influence will be small.

This iteration is repeated several times till successive estimations of the vocal tract transfer function is almost free from glottal source information, and vice-versa. The results produced for the present paper were obtained after an initialization lap and two more iterations. The details for the adaptive lattice implemented may be found in [8]. An example of the traces obtained from the utterance of vowel /a/ by a typical male speaker is shown in Figure 2.

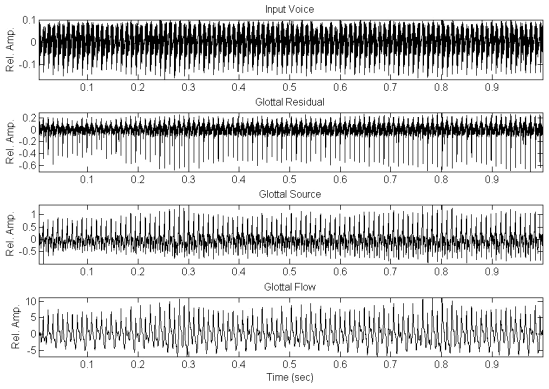


Figure 2. *A typical male speaker utterance of vowel /a/ and derived glottal traces. From top to bottom: input voice, glottal residual after adaptive inverse removal of the vocal tract, glottal source, and glottal flow*

3. Estimating the biometric fingerprint

The fingerprint is estimated from the FFT power spectral density of the glottal residual, after normalization to obtain the positions of envelope singularities as follows:

- The glottal residual and glottal source are windowed in *512* sample frames sliding *2 msec* and the power spectral density of each window is estimated by FFT in logarithmic (dB) scale as shown in Figure 3 (full line) for two examples of typical male and female speakers.
- The envelopes of the power spectral densities of these short-time power spectra are estimated (dot line).
- The maxima (*) and minima (◊) found on the respective envelopes are detected and their amplitudes and frequencies collected as two lists of ordered pairs: $\{T_{Mk}, f_{Mk}\}$ and $\{T_{mk}, f_{mk}\}$, with k the ordering index.
- The largest of all maxima (T_{Mm}, f_{Mm}) is used as a normalization reference both in amplitude and in frequency as given by:

$$\left.\begin{aligned}\tau_{Mk} &= T_{Mk} - T_{Mm}\\ \tau_{mk} &= T_{mk} - T_{Mm}\end{aligned}\right\};\quad 1 \le k \le K \tag{1}$$

$$\left.\begin{aligned}\varphi_{Mk} &= \frac{f_{Mk}}{f_{Mm}}\\ \varphi_{mk} &= \frac{f_{mk}}{f_{Mm}}\end{aligned}\right\};\quad 1 \le k \le K \tag{2}$$

An important parameter derived from the ordered minima and maxima pairs is the *slenderness* factor, defined on each "V" trough formed by each minimum and the two neighbor maxima, which may be defined as:

$$\sigma_{mk} = \frac{f_{Mm}\left(2T_{mk} - T_{Mk+1} - T_{Mk}\right)}{2\left(f_{Mk+1} - f_{Mk}\right)};\quad 1 \le k \le K \tag{3}$$

The results of estimating the normalized singularity parameters for the cases under study may be seen in Figure 4.

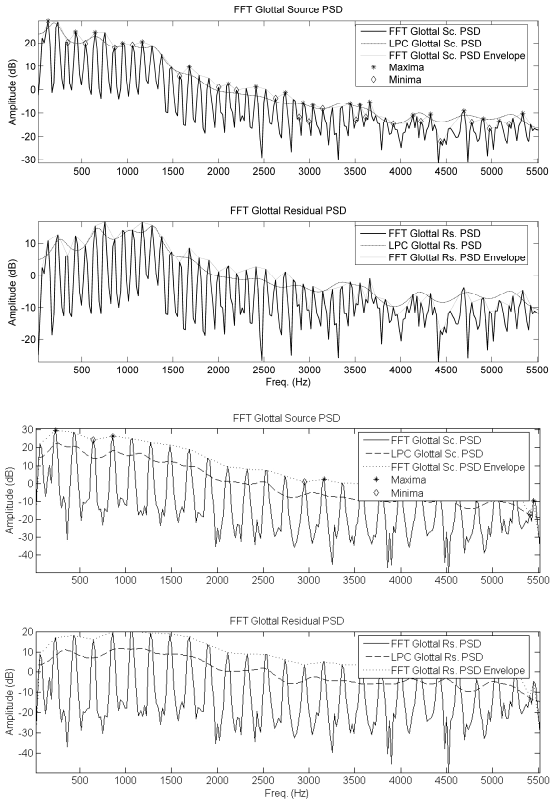


Figure 3. *Short-term Power Spectral Density of Glottal Signals. Two top plots: Glottal Source(showing superimposed the singularities) and Glottal Residual for a typical male speaker (vowel /a/). Two bottom plots: idem for a typical female speaker (vowel /a/).*

4. Materials and methods

To establish the validity of the proposed biometric fingerpring data recorded on a population set of 100 normal speakers equally distributed by gender were used. Subject ages ranged from 19 to 39, with an average of 26.77 years and a standard deviation of 5.75 years. The normal phonation condition of speakers was determined by electroglottographic, video-endoscopic and GRBAS [9] evaluations. The recordings at a sampling rate of 44,100 Hz consisted in three utterances of the vowel */a/* produced in different sessions of about *3 sec* per record, a *0.2 sec* segment derived from the central part being used in the experiments. For presentation purposes the traces were re-sampled at 11,025 Hz. This database was fully parameterized to obtain the singularity biometric fingerprint described in section 3. The normalized biometric fingerprints for the speakers presented are given in Figure 4.

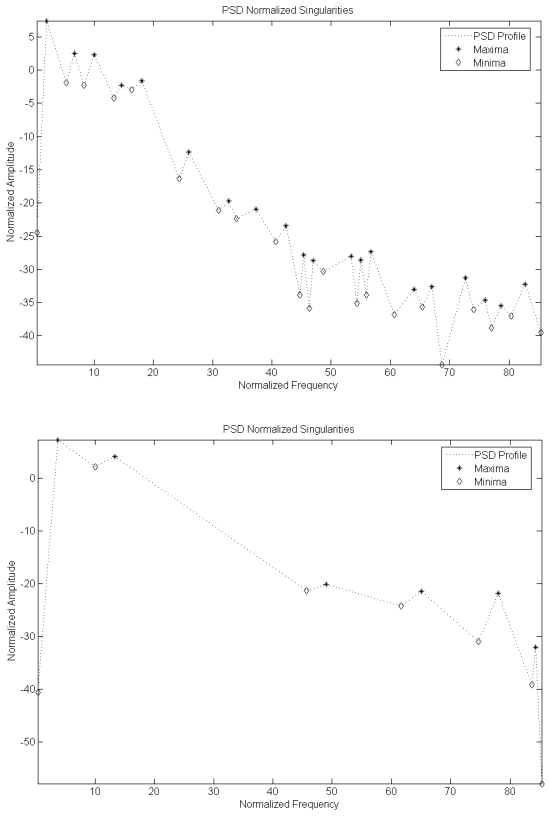


Figure 4. *Normalized singularity profiles for the male (top and female (bottom) typical utterances of vowel /a/*

A first inspection of the fingerprints shows that the one from the male speaker exhibits deeper "V" troughs at lower relative frequencies than the female case. This is consistent with the biomechanical explanation of the nature of peaks and troughs, as these are based on the mechanical resonances and anti-resonances of the systems of masses and springs describing vocal fold vibration. In general, female vocal folds show more stiff links among body and cover masses, and this would explain the lower amount of less sharp anti-resonances (see [3] and [5] for a wider explanation).

Important considerations are robustness and intra-speaker variability of estimates. The processing of the *0.2 sec* segments in *512 sample* windows sliding in *2 msec* steps produce around *76* estimates per segment. The positions of the 3 minima and maxima for the male and female speakers (plus the origin value) vs time are given in Figure 5.

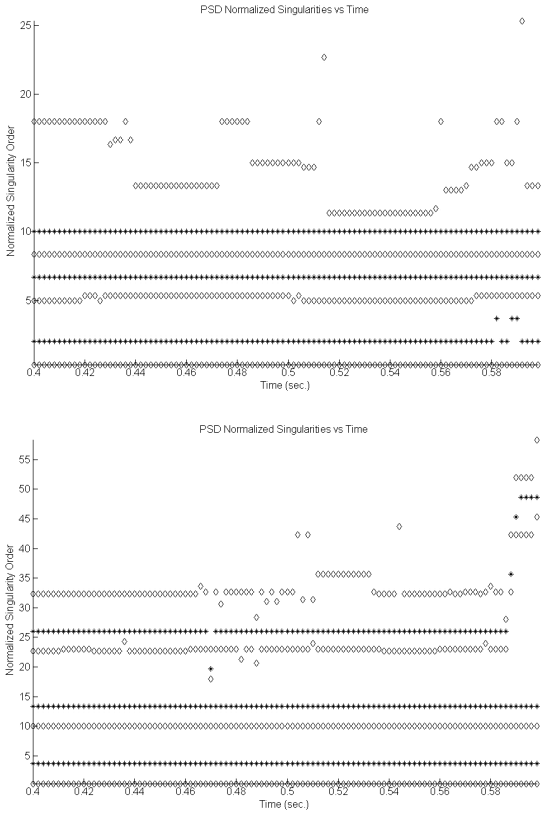


Figure 5. *Relative positions of the first 7 singularity points and the origin for a 0.2 sec segment of the reference male and female traces*

It may be observed that lower order are more stable than higher order estimates. This finding is consistent with the biomechanical explanation of the estimates. Lower frequency troughs and peaks are due to larger vocal fold masses, which for a given articulation and load do not change substantially during phonation, whereas higher order singularities are due to irregular small mass distributions on the cord, which may suffer important alterations during phonation and are more sensitive to vocal tract coupling effects. This is especially so when analyzing results from complete sentences including different voiced sounds, not shown here for the lack of extent.

5. Results and discussion

The issue of intra-speaker variability may be better illustrated giving the statistical dispersion of the estimates for sustained vowels as the cases studied in normalized amplitude and frequency as given in Figure 5.

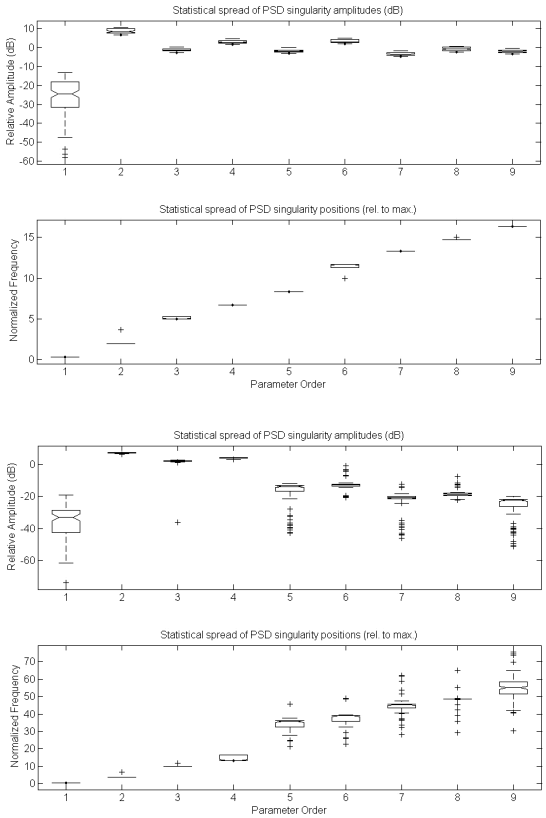


Figure 6. *Statistical distribution of the first 8 singularity points and the origin for a 0.2 sec segment of the traces shown in Figure 5. From top to bottom: Amplitudes (male), Normalized Frequencies (male), Amplitues (female), and Normalized Frequencies (female)*

The results shown confirm that the singularities in the male case are found at lower frequencies and that low frequency estimates are more stable than high frequency ones. This also explains why the spread of the estimates seems to be a little larger in the female case, as they appear at higher frequencies due to the more tense nature of female voice. As a general conclusion, it may be said that the intra-speaker variability of the low order singularities is small in sustained vowel-like utterances for normophonic speakers. This study has to be carried out to pathological cases as well and could be used for pathology detection. The estimations from the typical female voice shown are a little bit less stable. In previous work [6] it was shown that estimates from the glottal source may be used in the determination of the biomechanical parameters of the fold body, whereas the biomechanics of the fold cover could be obtained from the power spectral density of the mucosal wave correlate. An important pending study is the use of the glottal source or the glottal residual in determining the specific speaker's glottal profile.

6. Conclusions

The work shown is a generalization of prior studies using non-adaptive estimations of the vocal tract on short segments of vowels. The use of adaptive estimations allow a better accuracy in the estimates of the vocal tract, and consequently on the glottal signals. The extension of the glottal spectra singularities to time-varying conditions allow a better description of the non-stationary processes appearing in vocal fold vibration even in the production of sustained sounds. This improvement in the estimates may help in conducting more careful studies about inter-speaker and intra-speaker variability to extend the use of the glottal source spectral fingerprint to speaker identification and characterization applications [10][11].

7. Acknowledgments

This work is funded by grants TIC2003-08756, TEC2006-12887-C02-00 from Plan Nacional de I+D+i, Ministry of Education and Science, and project HESPERIA from the Program CENIT, CDTI, Ministry of Industry, Spain.

8. References

[1] Alku, P., "An Automatic Method to Estimate the Time-Based Parameters of the Glottal Pulseform", *Proc. of the ICASSP'92*, pp. II/29-32.

[2] Akande, O. O. and Murphy, P. J., "Estimation of the vocal tract transfer function with application to glottal wave analysis", *Speech Communication*, Vol. 46, No. 1, May 2005, pp. 1-13.

[3] Berry, D. A., "Mechanisms of modal and non-modal phonation", *J. Phonetics*, Vol. 29, 2001, pp. 431-450.

[4] Godino, J. I., Gomez, P., Automatic detection of voice impairments by means of short-term cepstral parameters and neural network based detectors. *IEEE Trans Biomed. Eng*. Vol. 51, 2004, pp. 380-384.

[5] Gómez, P., Godino, J. I., Díaz, F., Álvarez, A., Martínez, R., Rodellar, V., "Biomechanical Parameter Fingerprint in the Mucosal Wave Power Spectral Density", *Proc. of the ICSLP'04, 2004*, pp. 842-845.

[6] Gómez, P., Martínez, R., Díaz, F., Lázaro, C., Álvarez, A., Rodellar, V., Nieto, V., "Estimation of vocal cord biomechanical parameters by non-linear inverse filtering of voice", *Proc. of the 3rd Int. Conf. on Non-Linear Speech Processing NOLISP'05*, Barcelona, Spain, April 19-22 2005, pp. 174-183.

[7] Gómez, P., Rodellar, V., Álvarez, A., Lázaro, J. C., Murphy, K., Díaz, F., Fernández, R., "Biometrical Speaker Description from Vocal Cord Parameterization", *Proc. of ICASSP'06*, Toulouse, France, 2006, pp. 1036-1039.

[8] Haykin, S., *Adaptive Filter Theory*, (4th Ed.), Prentice-Hall, Upper Saddle River, NJ, 2001.

[9] Hirano, M., Hibi, S., Yoshida, T., Hirade, Y., Kasuya, H., and Kikuchi, Y., "Acoustic analysis of pathological voice. Some results of clinical application," *Acta Otolaryngologica*, vol. 105, no. 5-6, pp. 432-438, 1988.

[10] Nickel, R. M., "Automatic Speech Character Identification", *IEEE Circuits and Systems Magazine*, Vol. 6, No. 4, 2006, pp. 8-29.

[11] Whiteside, S. P., "Sex-specific fundamental and formant frequency patterns in a cross-sectional study," *J. Acoust. Soc. Am.*, vol. 110, no. 1, pp. 464–478, 2001.

Trajectory Mixture Density Network with Multiple Mixtures for Acoustic-articulatory Inversion

Korin Richmond

Centre for Speech Technology Research
Edinburgh University, Edinburgh, United Kingdom
korin@cstr.ed.ac.uk

Abstract

We have previously proposed a trajectory model which is based on a mixture density network trained with target variables augmented with dynamic features together with an algorithm for estimating maximum likelihood trajectories which respects the constraints between those features. In this paper, we have extended that model to allow diagonal covariance matrices and multiple mixture components. We have evaluated the model on an inversion mapping task and found the trajectory model works well, outperforming smoothing of equivalent trajectories using low-pass filters. Increasing the number of mixture components in the TMDN improves results further.

1. Introduction

Mainstream speech technology, such as automatic speech recognition and concatenative speech synthesis, is strongly focuses on the acoustic speech signal. This is natural, considering the acoustic domain is where the speech signal exists in transmission between humans, and we can conveniently measure and manipulate an acoustic representation of speech. However, an articulatory representation of speech has certain properties which are attractive and which may be exploited in modelling. Speech articulators move relatively slowly and smoothly, and their movements are continuous; the mouth cannot "jump" from one position to the next. Using knowledge of the speech production system could improve speech processing methods by providing useful constraints. Accordingly, there is growing interest in exploiting articulatory information and representations in speech processing, with many suggested applications; for example, low bit-rate speech coding [1], speech analysis and synthesis [2], automatic speech recognition [3, 4], animating talking heads and so on.

For an articulatory approach to be practical, we need convenient access to an articulatory representation. Recent work on incorporating articulation into speech technology has used data provided by X-ray microbeam cinematography and electromagnetic articulography (EMA). These methods, particularly the latter, mean we are now able to gather reasonably large quantities of articulatory data. However, they are still invasive techniques and require bulky and expensive experimental setups. Therefore, there is interest in developing a way to recover an articulatory representation from the acoustic speech signal. In other words, for a given acoustic speech signal we aim to estimate the underlying sequence of articulatory configurations which produced it. This is termed acoustic-articulatory inversion, or the inversion mapping.

The inversion mapping problem has been the subject of research for several decades. One approach has been to attempt analysis of acoustic signals based on mathematical models of speech production [5]. Another popular approach has been to use articulatory synthesis models, either as part of an analysis-by-synthesis algorithm [6], or to generate acoustic-articulatory corpora which may be used with a code-book mapping [7] or to train other models [8]. Much of the more recent work reported has applied machine learning models to human measured articulatory data, including artificial neural networks (ANNs) [9], codebook methods [10] and GMMs [11].

The inversion mapping is widely regarded as difficult because it may be an ill-posed problem; multiple evidence exists to suggest the articulatory-to-acoustic mapping is many-to-one, which means that instantaneous inversion of this mapping results in a one-to-many mapping. If which case, an inversion mapping method must take account of the alternative articulatory configurations possible in response to an acoustic vector.

In previous work [12, 9], we have successfully employed the mixture density network (MDN) [13] to address this problem. The MDN provides a probability density function (pdf) of arbitrary complexity over the target articulatory domain which is conditioned on the acoustic input. In [14], we began to extend this work to provide a statistical trajectory model, termed the Trajectory MDN, along similar lines as the HMM-based speech production model of [15] and the GMM-based inversion mapping of [11]. This was achieved by augmenting the static articulatory target data with dynamic delta and deltadelta features and incorporating the maximum likelihood parameter generation (MLPG) algorithm [16]. This allows to calculate the maximum likelihood estimate of articulatory trajectories which respect the constraints between the static and derived dynamic features.

This paper seeks to further the work in [14] with three specific aims: 1) to evaluate an extension to the TMDNs in [14] which allows mixture models with diagonal covariance matrices. 2) to evaluate the new implementation of TMDN on the full set of articulator channels, and in comparison with a low-pass filtering approach previously reported. 3) to evaluate TMDNs with multiple mixture components.

2. The Trajectory Mixture Density Network Model

We give here a very brief introduction to the MDN, and describe how it may be extended with the MLPG algorithm to give a trajectory model. For full details of the MDN and MLPG, the reader is referred to [13] and [16] respectively. We have attempted to retain the original notation as far as possible.

2.1. Mixture density networks

The MDN combines a mixture model with an ANN. Here, we will consider a multilayer perceptron and Gaussian mixture components. The ANN maps from the input vector $\mathbf{x}$ to the control parameters of the mixture model (priors α, means μ and

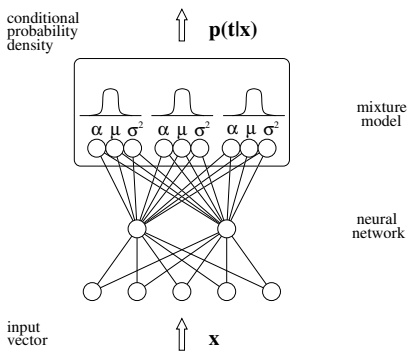


Figure 1: *The mixture density network is the combination of a mixture model and a neural network.*

variances σ^2), which in turn gives a pdf over the target domain, conditioned on the input vector $p(\mathbf{t}|\mathbf{x})$. The toy-example MDN in Figure 1 takes an input vector $\mathbf{x}$ (dimensionality 5) and gives the conditional probability density of a vector $\mathbf{t}$ (dimensionality 1) in the target domain. This pdf takes the form of a GMM with 3 components, so it is given as:

$$p(\mathbf{t}|\mathbf{x}) = \sum_{j=1}^{M} \alpha_j(\mathbf{x})\phi_j(\mathbf{t}|\mathbf{x}) \tag{1}$$

where M is the number of mixture components (in this example, 3), $\phi_j(\mathbf{t}|\mathbf{x})$ is the probability density given by the jth kernel, and $\alpha_j(\mathbf{x})$ is the prior for the jth kernel.

In order to constrain the GMM priors to within the range $0 \leq \alpha_j(\mathbf{x}) \leq 1$ and to sum to unity, the *softmax* function is used

$$\alpha_j = \frac{\exp(z_j^\alpha)}{\sum_{l=1}^{M} \exp(z_l^\alpha)} \tag{2}$$

where z_j^α is the output of the ANN corresponding to the prior for the jth mixture component. The variances are similarly related to the outputs of the ANN as

$$\sigma_j = \exp(z_j^\sigma) \tag{3}$$

where z_j^σ is the output of the ANN corresponding to the variance for the jth mixture component. This avoids the variance becoming ≤ 0. Finally, the means are represented directly:

$$\mu_{jk} = z_{jk}^\mu \tag{4}$$

where z_{jk}^μ is the value of the output unit corresponding to the kth dimension of the mean vector for the jth mixture component.

Training the MDN aims to minimise the negative log likelihood of the observed target data points

$$E = -\sum_n \ln \left\{ \sum_{j=1}^{M} \alpha_j(\mathbf{x}^n)\phi_j(\mathbf{t}^n|\mathbf{x}^n) \right\} \tag{5}$$

given the mixture model parameters. Since the ANN part of the MDN provides the parameters for the mixture model, this error function must be minimised with respect to the network weights. The derivatives of the error at the network output units corresponding separately to the priors, means and variances of the mixture model are calculated (see [13]) and then propagated back through the network to find the derivatives of the error with respect to the network weights. Thus, standard non-linear optimisation algorithms can be applied to MDN training.

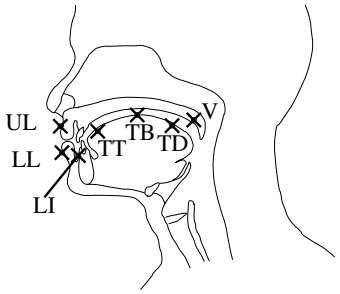


Figure 2: *Placement of EMA receiver coils in the MOCHA database for speaker* `fsew0`. *See Table 1 for the key to abbreviations.*

2.2. Maximum likelihood parameter generation

The first step to an MDN-based trajectory model is to train an MDN with target feature vectors augmented with dynamic features, derived from linear combinations of a window of static features. For the sake of simplicity and economy of space[1], we will consider MDNs with a single Gaussian distribution and a single target static feature c_t at each time step. Next, given the output of this MDN in response to a sequence of input vectors, in order to generate the maximum likelihood trajectory, we aim to maximize $P(\mathbf{O}|\mathbf{Q})$ with respect to $\mathbf{O}$, where $\mathbf{O} = [\mathbf{o}_1^T, \mathbf{o}_2^T, ..., \mathbf{o}_T^T]^T$, $\mathbf{o}_t = [c_t, \Delta c_t, \Delta\Delta c_t]$ and $\mathbf{Q}$ is the sequence of Gaussians output by our MDN. The relationship between the static features and those augmented with derived dynamic features can be arranged in matrix form,

$$\mathbf{O} = \mathbf{WC} \tag{6}$$

where $\mathbf{C}$ is a sequence of static features and $\mathbf{W}$ is a transformation matrix composed of the coefficients of the delta and deltadelta calculation window and $\mathbf{0}$. Under the condition expressed in Eq. 6, maximising $P(\mathbf{O}|\mathbf{Q})$ is equivalent to maximising $P(\mathbf{WC}|\mathbf{Q})$ with respect to $\mathbf{C}$. By setting

$$\frac{\partial \log P(\mathbf{WC}|\mathbf{Q})}{\partial \mathbf{C}} = 0 \tag{7}$$

a set of linear equations is obtained (see [16])

$$\mathbf{W}^T\mathbf{U}^{-1}\mathbf{WC} = \mathbf{W}^T\mathbf{U}^{-1}\mathbf{M}^T \tag{8}$$

where $\mathbf{M}^T = [\mu_{q_1}, \mu_{q_2}, ..., \mu_{q_T}]$ and $\mathbf{U}^{-1} = diag[\mathbf{U}_{q_1}^{-1}, \mathbf{U}_{q_2}^{-1}, ..., \mathbf{U}_{q_T}^{-1}]$ (μ_{q_T} and $\mathbf{U}_{q_t}^{-1}$ are the 3×1 mean vector and 3×3 (diagonal) covariance matrix respectively). Solving Eq. 8 for $\mathbf{C}$ computes the maximum likelihood trajectory.

3. Inversion mapping experiment

3.1. MOCHA articulatory data

The multichannel articulatory (MOCHA) dataset [17] used for the experiments in this paper gives the acoustic waveform recorded concurrently with electromagnetic articulograph (2D EMA) data. The sensors shown in Figure 2 provide x- and y-coordinates in the midsagittal plane at 500Hz sample rate. Speakers were recorded reading a set of 460 short, phonetically-balanced British-TIMIT sentences. Female speaker `fsew0` was used for the experiments here. This is the same data set as used

[1] the full version of this paper will require a description of MLPG in the case of multiple mixture components

label	articulator	label	articulator
UL	Upper lip	TT	Tongue tip
LL	Lower lip	TB	Tongue body
LI	Lower incisor	TD	Tongue Dorsum
V	Velum		

Table 1: *Key for placement of coils in the MOCHA dataset for speaker. Coil placement abbreviations are suffixed with "_x" and "_y" to designate the x- and y-coordinate for a given coil in the midsagittal plane respectively.*

previously [9, 14], and so enables comparison with those and similar results reported in the literature (e.g. [11]).

3.1.1. Data processing

The acoustic data was converted to frames of 20 melscale filterbank coefficients using a Hamming window of 20ms with a shift of 10ms. These were z-score normalised and scaled to the range [0.0,1.0]. The EMA trajectories were downsampled to match the 10ms shift rate, then z-score normalised and scaled to the range [0.1,0.9] using the normalisation method described in [12]. Frames from silence at the beginning and end of the files were discarded, using the labelling provided with MOCHA.

368 utterances were used for the training set, and the validation and test sets contained 46 utterances each (the same as [9, 14]). A context window of 20 consecutive acoustic frames was used as input, which increased the order of the acoustic vector paired with each articulatory vector to 400.

3.2. Method

We trained TMDNs with 1, 2 and 4 mixture components for each of the 14 EMA channels, making a total of 42 models. In [14], we trained separate MDNs for the static, delta and deltadelta features for each articulatory channel. Here, in contrast, our implementation has been extended and now allows diagonal covariance matrices, and so the three feature streams for each articulator channel were trained in a single network. All networks contained a hidden layer of 80 units. The scaled conjugate gradients non-linear optimisation algorithm was run for a maximum of 4000 epochs, and the separate validation set was used to identify the point at which an optimum appeared to have been reached. To generate output trajectories from the TMDN, we simply ran the input data for an utterance through the TMDNs for each articulatory channel, and then ran the MLPG algorithm on the resulting sequences of pdfs.

To evaluate the Trajectory MDN, we compared the resulting trajectories with those of the output units corresponding to the mean of the static feature. This output is approximately equivalent to that of an MLP (with linear output activation function) trained with a standard least-squares error function[2]. In this way, we can directly observe the effect of using the augmented features without considering the effects of two systems having been trained differently. Finally, we also low-pass filtered the static mean trajectories as a smoothing step which has been shown in the past to improve inversion results [12, 11], and compared those smoothed trajectories with the TMDN output.

4. Results

Table 2 lists the results of 14 TMDNs trained on each articulatory channel separately, using an output pdf containing a single Gaussian. Two error metrics have been used: correlation between the target and output trajectories, and root mean square error (RMSE) expressed in millimetres. The table also lists the results previously reported in [9], which used an MLP with exactly the same dataset, for comparison. It can be seen that the improvement is substantial. By way of further comparison with other studies, [11] reported an average RMS error of 1.45mm for MOCHA speaker `fsew0`.

	Correlation		RMSE(mm)		RMSE(mm)
Channel	MLP	TMDN	MLP	TMDN	reduction %
ul_x	0.58	0.68	0.99	0.90	9.5
ul_y	0.72	0.79	1.16	1.05	9.9
ll_x	0.60	0.69	1.21	1.10	9.2
ll_y	0.75	0.83	2.73	2.27	16.8
li_x	0.56	0.63	0.89	0.82	8.1
li_y	0.80	0.85	1.19	1.03	13.3
tt_x	0.79	0.85	2.43	2.12	12.9
tt_y	0.84	0.90	2.56	2.08	18.7
tb_x	0.81	0.85	2.19	1.96	10.4
tb_y	0.83	0.89	2.14	1.76	17.6
td_x	0.79	0.84	2.04	1.85	9.5
td_y	0.71	0.82	2.31	1.89	18.2
v_x	0.79	0.86	0.42	0.35	15.6
v_y	0.77	0.83	0.41	0.37	10.2

Table 2: *Comparison of results for Trajectory MDNs (TMDN) with a single Gaussian with the MLP described in [9]. Exactly the same training, validation and testing datasets have been used. The Average RMSE(mm) in [9] was* 1.62*mm, compared with an average here of* 1.4*mm.*

In order to investigate the effect of using dynamic features and the MLPG algorithm within the Trajectory MDN, we have compared these results for TMDNs with a single Gaussian with those obtained using low-pass filtering, as described in [12, 11]. Table 4 compares three conditions: *"TMDN"*, *"static only"* and *"static lpfilt"*. For the *"static only"* condition, we have used the TMDN's output corresponding to the mean for the static target feature as the output trajectory. For the *"static lpfilt"* condition, we have further low-pass filtered the static mean above using the cutoff frequencies listed in Table 3. These channel-specific cutoff frequencies were determined empirically in [12], and are very similar to those given in [11]. As expected, it can be seen that low-pass filtering improves results for all channels. However, using the dynamic features and the MLPG algorithm in the Trajectory MDN results in the best performance, with improvements varying between 0.6 and 2.4% over low-pass filtering.

The improvements over using low-pass filtering in Table 4, although consistent, are not huge. However, in contrast to low-pass filtering, the TMDN is able to make use of multiple mixture components, which can potentially increase performance further. Table 5 performs this comparison, by the addition of results for TMDNs with 2 and 4 mixture components. It can be seen that increasing the number of mixture components improves results by up to 8.3% over those obtained using low-pass filtering.[3]

5. Conclusion

The results of this paper show we have successfully extended the Trajectory MDN first described in [14] to allow diagonal covariance matrices. For all 14 articulator channels tested, the TMDN with a single Gaussian output pdf performed better than

[2] although this MLP has been trained with augmented target features, which seems to have had a beneficial effect, possible due to "multitask" learning

[3] at the time of submitting this proposal paper, results for the other channels were still in preparation

Channel	Correlation			RMSE(mm)			RMSE(mm)
	static only	static lpfilt	TMDN	static only	static lpfilt	TMDN	reduction %
ul_x	0.63	0.67	0.68	0.93	0.90	0.90	0.6
ul_y	0.74	0.77	0.79	1.13	1.06	1.05	1.5
ll_x	0.64	0.69	0.69	1.17	1.11	1.10	1.0
ll_y	0.81	0.83	0.83	2.40	2.31	2.27	1.6
li_x	0.57	0.62	0.63	0.88	0.84	0.82	2.4
li_y	0.83	0.84	0.85	1.07	1.05	1.03	1.5
tt_x	0.82	0.84	0.85	2.26	2.14	2.12	1.0
tt_y	0.88	0.89	0.90	2.19	2.12	2.08	1.8
tb_x	0.83	0.85	0.85	2.05	1.99	1.96	1.2
tb_y	0.87	0.89	0.89	1.88	1.80	1.76	1.8
td_x	0.81	0.83	0.84	1.95	1.88	1.85	2.1
td_y	0.78	0.81	0.82	2.04	1.92	1.89	1.7
v_x	0.84	0.85	0.86	0.37	0.36	0.35	1.2
v_y	0.80	0.82	0.83	0.39	0.37	0.37	1.6

Table 4: *Comparison of correlation and RMS error (in millimetres) for Trajectory MDN model ("TMDN") with the static mean MDN output only ("static only") and low-pass filtered static mean ("static lpfilt"). Low-pass filtering the static feature output, using the cutoff frequencies in Table 3, improves results for all channels. However, using the delta and deltadelta features in the Trajectory MDN gives the best performance for all channels. The TMDN here has a single Gaussian as its output distribution. Compare these results with those in table 5.*

	ul	ll	li	tt	tb	td	v
_x	3Hz	3Hz	3Hz	6Hz	6Hz	7Hz	5Hz
_y	5Hz	8Hz	7Hz	9Hz	7Hz	6Hz	5Hz

Table 3: *Channel-specific cutoff frequencies used for low pass filtering (empirically determined in [12]).*

Channel	static lpfilt	TMDN			% best reduction
		1mix	2mix	4mix	
tt_x	2.14	2.12	2.09	2.10	2.1
tt_y	2.12	2.08	1.98	1.94	8.3
td_x	1.88	1.85	1.81	1.83	4.1
td_y	1.92	1.89	1.85	1.88	3.6

Table 5: *Comparison of RMS error (expressed in millimetres) between using the low-pass filtered static feature mean ("static lpfilt") and Trajectory MDNs with 1, 2 or 4 mixture components. Compared with the results in Table 4, using multiple mixture components improves the performance of the TMDN over low-pass filtering even further.*

low-pass filter smoothing. Increasing the number of mixture components improved results further. This is a unique advantage of the TMDN model over smoothing single trajectories with, for example, low-pass filters.

6. References

[1] J. Schroeter and M. M. Sondhi, "Speech coding based on physiological models of speech production," in *Advances in Speech Signal Processing*, S. Furui and M. M. Sondhi, Eds. New York: Marcel Dekker Inc, 1992, ch. 8, pp. 231–268.

[2] T. Toda, A. Black, and K. Tokuda, "Mapping from articulatory movements to vocal tract spectrum with Gaussian mixture model for articulatory speech synthesis," in *Proc. 5th ISCA Workshop on Speech Synthesis*, 2004.

[3] S. King, J. Frankel, K. Livescu, E. McDermott, K. Richmond, and M. Wester, "Speech production knowledge in automatic speech recognition," *Journal of the Acoustical Society of America*, vol. 121, no. 2, p. (pages TBA), February 2007.

[4] A. Wrench and K. Richmond, "Continuous speech recognition using articulatory data," in *Proc. ICSLP 2000*, Beijing, China, 2000.

[5] H. Wakita, "Estimation of vocal-tract shapes from acoustical analysis of the speech wave: The state of the art," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-27, pp. 281–285, 1979.

[6] K. Shirai and T. Kobayashi, "Estimating articulatory motion from speech wave," *Speech Communication*, vol. 5, pp. 159–170, 1986.

[7] B. S. Atal, J. J. Chang, M. V. Mathews, and J. W. Tukey, "Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer sorting technique," *J. Acoust. Soc. Am.*, vol. 63, pp. 1535–1555, 1978.

[8] M. G. Rahim, W. B. Kleijn, J. Schroeter, and C. C. Goodyear, "Acoustic-to-articulatory parameter mapping using an assembly of neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1991, pp. 485–488.

[9] K. Richmond, S. King, and P. Taylor, "Modelling the uncertainty in recovering articulation from acoustics," *Computer Speech and Language*, vol. 17, pp. 153–172, 2003.

[10] J. Hogden, A. Lofqvist, V. Gracco, I. Zlokarnik, P. Rubin, and E. Saltzman, "Accurate recovery of articulator positions from acoustics: New conclusions based on human data," *J. Acoust. Soc. Am.*, vol. 100, no. 3, pp. 1819–1834, September 1996.

[11] T. Toda, A. Black, and K. Tokuda, "Acoustic-to-articulatory inversion mapping with Gaussian mixture model," in *Proc. 8th International Conference on Spoken Language Processing*, Jeju, Korea, 2004.

[12] K. Richmond, "Estimating articulatory parameters from the acoustic speech signal," Ph.D. dissertation, The Centre for Speech Technology Research, Edinburgh University, 2002.

[13] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[14] K. Richmond, "A trajectory mixture density network for the acoustic-articulatory inversion mapping," in *Proc. Interspeech*, Pittsburgh, USA, September 2006.

[15] S. Hiroya and M. Honda, "Estimation of articulatory movements from speech acoustics using an HMM-based speech production model," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 2, pp. 175–185, mar 2004.

[16] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, Istanbul, Turkey, June 2000, pp. 1315–1318.

[17] A. Wrench, "The MOCHA-TIMIT articulatory database," http://www.cstr.ed.ac.uk/artic/mocha.html, 1999.

Application of Feature Subset Selection based on Evolutionary Algorithms for Automatic Emotion Recognition in Speech

Aitor Álvarez, Idoia Cearreta, Juan Miguel López, Andoni Arruti, Elena Lazkano, Basilio Sierra, Nestor Garay

Computer Science Faculty (University of the Basque Country)
Manuel Lardizabal 1, E-20018 Donostia (Gipuzkoa), Spain
aalvarez031@ikasle.ehu.es

Abstract

The study of emotions in human-computer interaction is a growing research area. Focusing on automatic emotion recognition, work is being performed in order to achieve good results particularly in speech and facial gesture recognition. In this paper we present a study performed to analyze different machine learning techniques validity in automatic speech emotion recognition area. Using a bilingual affective database, different speech parameters have been calculated for each audio recording. Then, several machine learning techniques have been applied to evaluate their usefulness in speech emotion recognition. In this particular case, techniques based on evolutive algorithms (EDA) have been used to select speech feature subsets that optimize automatic emotion recognition success rate. Achieved experimental results show a representative increase in the abovementioned success rate.

1. Introduction

Human beings are eminently emotional, as their social interaction is based on the ability to communicate their emotions and perceive the emotional states of others [1]. *Affective computing*, a discipline that develops devices for detecting and responding to users' emotions [2], is a growing research area [3]. The main objective of affective computation is to capture and process affective information with the aim of enhancing the communication between the human and the computer.

Within the scope of affective computing, the development of affective applications is a challenge that involves analyzing different multimodal data sources. In order to develop such applications, a large amount of data is needed in order to include a wide range of emotionally significant material. Affective databases are a good chance for developing affective recognizers or affective synthesizers.

In this paper different speech paralinguistic parameters have been calculated for the analysis of the human emotional voice, using several audio recordings. These recordings are stored in a bilingual and multimodal affective database. Several works have already been done in which the use of Machine Learning paradigms takes a principal role.

2. Related work

As previously mentioned affective databases provide a good opportunity for training affective applications. This type of databases usually record information such as images, sounds, psychophysiological values, etc. There are some references in the literature that present affective databases and their characteristics. In [4], the authors carried out a wide review of affective databases. Other interesting reviews are the ones provided in [5] and [6].

Many studies have been focused on the different features used in human emotional speech analysis [7, 8]. The number of voice features analysed varies among the studies, but basically most of these are based in fundamental frequency, energy and timing parameters, such as speech rate or mean phone duration.

Works where the use of Machine Learning paradigms take a principal role can also be found in the literature. [9] presented a good reference paper. The Neural Networks Journal devoted a special issue to emotion treatment from a Neural Networks perspective [10]. The work by [4] is related with this paper in the sense of using a Feature Selection method in order to apply a Neural Network to emotion recognition in speech, although both, the methods to perform the FSS and the paradigms used, are different. In this line it has to be pointed out the work by [11] which uses a reduced number of emotions and a greedy approach to select the features.

3. Study of automatic emotion recognition relevant parameters using Machine Learning paradigms

3.1. RekEmozio Database

The RekEmozio bilingual database was created with the aim of serving as an information repository for performing research on user emotion. The aim when building the RekEmozio resource was to add descriptive information about the performed recordings, so that processes such as extracting speech parameters and video features could be carried out on them. Members of different work groups involved in research projects related to RekEmozio have performed several processes for extracting speech and video features; this information was subsequently added to the database. The emotions used were chosen based on [12], and the neutral emotion was added. The characteristics of the RekEmozio database are described in [13]. The languages that are considered in RekEmozio database are Spanish and Basque.

3.2. Emotional feature extraction

For emotion recognition in speech, one of the most important questions is which features should be extracted from the voice signal. Previous studies show us that it is difficult to find

specific voice features that could be used as reliable indicators of the emotion present in the speech [14].

In this work, RekEmozio database audio recordings (stereo wave files, sampled at 44100 Hz) have been processed using standard signal processing techniques (windowing, Fast Fourier Transform, auto-correlation …) to extract a wide group of 32 features which are described below. Supposing that each recording in the database corresponds to one single emotion, only one global vector of features has been obtained for each recording by using some statistical operations. Parameters used are global parameters calculated over entire recordings. Selected features are detailed next (in italics):

- **Fundamental Frequency (F0):** It is the most common feature analyzed in several studies [7, 8]. For F0 estimation we used Sun algorithm [15] and statistics are computed: *Maximum*, *Minimum*, *Mean*, *Range*, *Variance*, *Standard deviation* and *Maximum positive slope in F0 contour*.
- **RMS Energy:** The mean energy of speech quantified by calculating root mean square (RMS) value and 6 statistics *Maximum*, *Minimum*, *Mean*, *Range*, *Variance* and *Standard Deviation.*
- **Loudness:** *Absolute loudness* based on Zwicker's model [16].
- **Spectral distribution of energy:** Each emotion requires a different effort in the speech and it is known that the spectral distribution of energy varies with speech effort [7]. We have computed energy in *Low band*, between 0 and 1300 Hz, *Medium band*, between 1300 and 2600 Hz and *High band* from 2600 to 4000 Hz [17].
- **Mean Formants and Bandwidth:** Energy from the sound source (vocal folds) is modified by the resonance characteristics of the vocal tract (formants). Acoustic variations dues to emotion are reflected in formants [18]. The *first three mean Formants*, and their corresponding *mean Bandwidths*.
- **Jitter:** *Perturbation in vibration of vocal chords*. It is estimated based on the model presented by [19].
- **Shimmer:** *Perturbation cycle to cycle of the energy*. Its estimation is based on the previously calculated absolute loudness.
- **Speaking Rate:** Rhythm is known to be an important aspect in recognition of emotion in speech. Progress has been made on a simple aspect of rhythm, the alternation between speech and silence [7]. The speaking rate estimation has been divided in 6 values based on their duration with respect to the whole elocution: *Duration of voice* part, *Silence* part, *Maximum voice* part, *Minimum voice* part, *Maximum silence* part and *Minimum silence* part.

3.3. Machine Learning standard paradigms used

In the supervised learning task, a classification problem has been defined where the main goal is to construct a model or a classifier able to manage the classification itself with acceptable accuracy. With this aim, some variables are to be used in order to identify different elements, the so called predictor variables. For the current problem, each sample is composed by the set of 32 speech related values, while the label value is one of the seven emotions identified. The single paradigms used in our experiments that come from the family of *Machine Learning* (ML) are briefly introduced

3.3.1. Decision trees

A decision tree consists of nodes and branches to partition a set of samples into a set of covering decision rules. In each node, a single test or decision is made to obtain a partition. The starting node is usually referred as the root node. In each node, the goal is to select an attribute that makes the best partition between the classes of the samples in the training set [20], [21]. In our experiments, two well-known decision tree induction algorithms are used, ID3 [22] and C4.5 [23].

3.3.2. Instance-Based Learning

Instance-Based Learning (IBL) has its root in the study of nearest neighbor algorithm [24] in the field of machine learning. The simplest form of nearest neighbor (NN) or k-nearest neighbor (k-NN) algorithms simply stores the training instances and classifies a new instance by predicting the same class its nearest stored instance has or the majority class of its k nearest stored instances have, respectively, according to some distance measure as described in [25]. The core of this non-parametric paradigm is the form of the similarity function that computes the distances from the new instance to the training instances, to find the nearest or k-nearest training instances to the new case. In our experiments the IB paradigm is used, an inducer developed in the MLC++ project [26] and based on the works of [27] and [28].

3.3.3. Naive Bayes classifiers

The Naive-Bayes (NB) rule [29] uses the Bayes theorem to predict the class for each case, assuming that the predictive genes are independent given the category. To classify a new sample characterized by d genes X = (X_1,X_2, ...,X_d), the NB classifier applies the following rule:

$$c_{N-B} = \arg\max_{cj \in C} p(cj) \prod_{i=1}^{d} p(xi \mid cj) \quad (1)$$

where C_{N-B} denotes the class label predicted by the Naive-Bayes classifier and the possible classes of the problem are grouped in $C = \{c_1, \ldots, c_l\}$. A normal distribution is assumed to estimate the class conditional densities for predictive genes. Despite its simplicity, the NB rule obtains better results than more complex algorithms in many domains.

3.3.4. Naive Bayesian Tree learner

The naive Bayesian tree learner, NBTree [30], combines naive Bayesian classification and decision tree learning. It uses a tree structure to split the instance space into sub-spaces defined by the paths of the tree, and generates one naive Bayesian classifier in each sub-space.

3.3.5. Feature Subset Selection by Estimation of Distribution Algorithms

The basic problem of ML is concerned with the induction of a model that classifies a given object into one of several known

classes. In order to induce the classification model, each object is described by a pattern of *d* features. Here, the ML community has formulated the following question: *are all of these d descriptive features useful for learning the 'classification rule'?* On trying to respond to this question, we come up with the Feature Subset Selection (FSS) [31] approach which can be reformulated as follows: *given a set of candidate features, select the 'best' subset in a classification problem*. In our case, the 'best' subset will be the one with the best predictive accuracy.

Most of the supervised learning algorithms perform rather poorly when faced with many irrelevant or redundant (depending on the specific characteristics of the classifier) features. In this way, the FSS proposes additional methods to reduce the number of features so as to improve the performance of the supervised classification algorithm.

FSS can be viewed as a search problem [32], with each state in the search space specifying a subset of the possible features of the task. Exhaustive evaluation of possible feature subsets is usually unfeasible in practice due to the large amount of computational effort required. In this way, any feature selection method must determine the nature of the search process. In the experiments performed, an Estimation of Distribution Algorithm (EDA) has been used which has the model accuracy as fitness function.

To assess the goodness of each proposed gene subset for a specific classifier, a wrapper approach is applied. In the same way as supervised classifiers when no gene selection is applied, this wrapper approach estimates, by using the 10-fold crossvalidation [33] procedure, the goodness of the classifier using only the variable subset found by the search algorithm.

4. Experimental Results

The above mentioned methods have been applied over the crossvalidated data sets using the MLC++ library [26]. Each dataset corresponds to a single actor. Experiments were carried out with and without FSS in order to extract the accuracy improvement introduced by the feature selection process. Tables 1 and 2 show the classification results obtained using the whole set of variables, for Basque and Spanish languages respectively. Each column represents a female (Fi) of male (Mi) actor, and mean values corresponding to each classifier/gender is also included. Last column presents the total average for each classifier.

Table 1: 10-fold crossvalidation accuracy for Basque Language using the whole variable set

	Female				*Male*					*Total*
	F1	F2	F3	**mean**	M1	M2	M3	M4	**mean**	
IB	35.4	48.8	35.2	39.8	44.2	49.3	36.9	40.9	42.8	41.5
ID3	38.7	45.5	44.7	**42.9**	46.7	46.9	43.3	51.1	47.0	45.3
C4.5	41.5	52.2	35.0	42.9	60.4	53.3	45.1	49.5	**52.0**	**48.1**
NB	42.9	45.8	37.7	42.1	52.2	44.1	36.2	41.4	43.5	42.9
NBT	42.3	39.8	35.2	39.1	53.1	46.2	45.2	43.3	46.9	43.6

Table 2: 10-fold crossvalidation accuracy for Spanish Language using the whole variable set

	Female						*Male*						*Total*
	F1	F2	F3	F4	F5	**mean**	M1	M2	M3	M4	M5	**mean**	
IB	34.6	43.6	54.6	54.6	38.2	45.1	25.5	33.6	51.8	47.7	33.6	**38.4**	**41.8**
ID3	36.4	52.7	49.1	47.3	42.7	**45.6**	20.9	30.9	40.9	47.3	40.0	36.0	40.8
C4.5	30.9	50.0	46.4	43.6	42.7	42.7	29.1	31.8	46.4	42.7	35.5	37.1	39.9
NB	38.2	42.7	49.1	40.0	42.7	42.5	24.6	30.9	49.1	45.5	34.6	36.9	39.7
NBT	42.7	43.6	49.1	50.0	39.1	44.9	18.2	27.3	40.9	48.2	42.7	35.5	40.2

Results don't seem very impressive; ID3 best classifies the emotions for female actresses, for both Basque and Spanish languages, while C4.5 outstands for Basque male actors and IB for Spanish male actors.

Results obtained after applying FSS are more appealing, as can be seen in Tables 3 and 4. There, classifier IB appears as the best paradigm for all the categories, female and male, and Basque and Spanish languages. Moreover, the accuracies outperform the previous ones in more than 15%. It must also be highlighted that FSS improves the well classified rate for all the ML paradigms, as it can be seen in Figure 1.

Table 3: 10-fold crossvalidation accuracy for Basque Language using FSS

	Female				*Male*					*Total*
	F1	F2	F3	**mean**	M1	M2	M3	M4	**mean**	
IB	63.0	68.0	59.3	**63.5**	72.7	67.4	61.0	62.8	**65.9**	**64.9**
ID3	62.7	60.5	65.5	62.9	72.7	62.0	56.5	62.7	63.4	63.2
C4.5	60.2	66.0	60.0	62.1	71.8	62.8	60.1	63.6	64.6	63.5
NB	64.5	64.6	48.9	59.3	74.6	62.5	62.7	60.0	64.9	62.5
NBT	58.6	61.1	54.8	58.1	74.4	59.9	62.7	59.4	64.1	61.6

Table 4: 10-fold crossvalidation accuracy for Spanish Language using FSS

	Female						***Male***						*Total*
	F1	F2	F3	F4	F5	**mean**	M1	M2	M3	M4	M5	**mean**	
IB	61.8	66.4	75.5	71.8	68.2	**68.7**	42.7	57.3	69.1	63.6	60.9	**58.7**	**63.7**
ID3	59.1	66.4	66.4	60.0	61.8	62.7	42.7	51.8	66.4	61.8	60.0	56.5	59.6
C4.5	57.3	62.7	64.6	65.5	63.6	62.7	43.6	56.4	65.5	64.6	56.4	57.3	60.0
NB	54.6	59.1	68.2	65.5	60.0	61.5	40.9	48.2	64.6	59.1	51.8	52.9	57.2
NBT	53.6	66.4	63.6	58.2	60.0	60.4	38.2	47.3	60.0	63.6	59.1	53.6	57.0

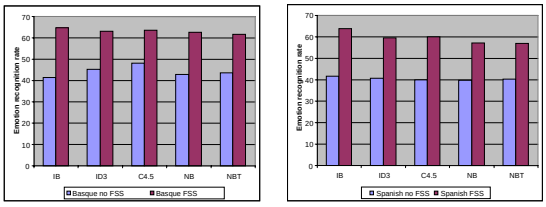


Figure 1: The improvement in Basque and Spanish languages using FSS in all the classifiers.

5. Conclusions and future work

RekEmozio database has been used to training some automatic recognition systems. In this paper we have shown that applying FSS enhances classification rates for the ML paradigms that we have used (IB, ID3, C4.5, NB and NBTree).

An analysis of the selected features by FSS is required. Also, the speech data should be combined with visual information. This combination could be performed by means of a multiclassifier model [34].

6. References

[1] Casacuberta, D., *La mente humana: Diez Enigmas y 100 preguntas (The human mind: Ten Enigmas and 100 questions)*, Océano (Ed), Barcelona, Spain, ISBN: 84-7556-122-5, 2001.

[2] Picard, R. W., *Affective Computing*, MIT Press, Cambridge, MA, 1997.

[3] Tao, J., Tan, T., "Affective computing: A review", *In: J. Tao, T. Tan, R. W. Picard (eds.): Lecture Notes in Computer Science, Vol. 3784 - Proceedings of The First International Conference on Affective Computing & Intelligent Interaction (ACII'05)*, Beijing, China, 981-995, 2005.

[4] Cowie, R., Douglas-Cowie, E., Cox, C., "Beyond emotion archetypes: Databases for emotion modelling using neural networks", *Neural Networks, Vol. 18, 2005, p 371-388.*

[5] Humaine, "Retrieved January 10, 2007", from [http://emotion-research.net/wiki/Databases], (n.d.).

[6] López, J.M., Cearreta, I., Fajardo, I., Garay, N., "Validating a multimodal and multilingual affective database", *To be published in Proceedings of HCI International*, 2007.

[7] Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., Taylor, J., "Emotion recognition in human-computer interaction", *IEEE Signal Processing Magazine, Vol. 18(1), 2001, p 32-80.*

[8] Schröder, M., *Speech and Emotion Research: An overview of research frameworks and a dimensional approach to emotional speech synthesis*, Ph.D. thesis, PHONUS 7, Research Report of the Institute of Phonetics, Saarland University, 2004.

[9] Dellaert, F., Polzin, T., Waibel, A., "Recognizing Emotion in Speech", *In Proc. of ICSLP'96*, 1996.

[10] Taylor, J. G., Scherer, K., Cowie, R., "Neural Networks, special issue on Emotion and Brain", *Vol. 18, Issue 4, 2005, p 313-455.*

[11] Huber, R., Batliner, A., Buckow, J., Noth, E., Warnke, V., Niemann, H., "Recognition of emotion in a realistic dialogue scenario", *In Proc. ICSLP'00*, p 665-668, 2000.

[12] Ekman, P., Friesen, W., *Pictures of facial affect*, Consulting Psychologist Press, Palo Alto, CA, 1976.

[13] López, J.M., Cearreta, I., Garay, N., López de Ipiña, K., Beristain, A., "Creación de una base de datos emocional bilingüe y multimodal", *In Redondo, M.A., Bravo C., Ortega M. (Eds). Proceeding of the 7th Spanish Human Computer Interaction Conference, Interacción-06*, Puertollano, p 55-66, 2006.

[14] Laukka, P., *Vocal Expression of Emotion. Discrete-emotions and Dimensional Accounts*, Acta Universitatis Upsaliensis. Comprehensive Summaries of Uppsala Dissertations from the Faculty of Social Sciences, pp 141, 80, ISBN 91-554-6091-7, Uppsala, 2004.

[15] Sun, X., "Pitch determination and voice quality analysis using subharmonic-to-harmonic ratio", *To appear in the Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Orlando, Florida, 2002

[16] Fernandez, R., *A Computational Model for the Automatic Recognition of Affect in Speech*, Ph.D. thesis, Massachusetts Institute of Technology, 2004.

[17] Kazemzadeh, A., Lee, S., Narayanan, S., "Acoustic correlates of user response to errors in human-computer dialogues", *Proc. IEEE ASRU, (St. Thomas, U.S. Virgin Islands)*, 2003 (December).

[18] Bachorowski, J.A., Owren, M. J., "Vocal expression of emotion: Acoustic properties of speech are associated with emotional intensity and context" *Psychological Science, Vol. 6,1995, p 219-224.*

[19] Rothkrantz, L.J.M., Wiggers, P., van Wees, J.W.A., van Vark, R.J., "Voice stress analysis", *Proceedings of Text, Speech and Dialogues 2004*, 2004.

[20] Martin, J.K., "An exact probability metric for Decision Tree splitting and stopping, Machine Learning", 1997, p 28(2/3).

[21] Mingers, J., "A comparison of methods of pruning induced Rule Trees, Technical Report", Coventry, England: University of Warwick, School of Indutrial and Business Studies, 1988.

[22] Quinlan, J.R., "Induction of Decision Trees", *Machine Learning, Vol 1, 1986, p 81-106.*

[23] Quinlan, J.R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, Inc. Los Altos, California, 1993.

[24] Dasarathy, B.V., "Nearest Neighbor (NN) Norms: NN Pattern Recognition Classification Techniques", *IEEE Computer Society Press*, 1991.

[25] Ting, K.M., *Common issues in Instance-Based and Naive-Bayesian classifiers*, Ph.D. Thesis, Basser Department of Computer Science, The University of Sidney, Australia, 1995.

[26] Kohavi, R., Sommerfield, D., Dougherty, J., "Data mining using MLC++, a Machine Learning Library in C++", *International Journal of Artificial Intelligence Tools, Vol. 6 (4), 1997, p 537-566*, [http://www.sgi.com/Technology/mlc/].

[27] Aha, D., Kibler, D., Albert, M.K., "Instance-Based learning algorithms", *Machine Learning, Vol. 6*, 37-66, 1991.

[28] Wettschereck, D., *A study of distance-based Machine Learning Algorithms*, Ph.D. Thesis, Oregon State University, 1994.

[29] Minsky, M. "Steps towards artificial intelligence", *Proceedings of the IRE*, 49, 8–30, 1961.

[30] Kohavi, R., "Scaling up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid", *In: Simoudis, E., Han, J.-W., Fayyad, U. M. (eds.): Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. Menlo Park, CA: AAAI Press*, p 202-207, 1996.

[31] Liu, H., Motoda, H., *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer Academic Publishers, 1998.

[32] Inza, I., Larrañaga, P., Etxeberria, R., Sierra, B., "Feature subsetselection by Bayesian network-based optimization", *Artificial Intelligence, Vol. 123, 2000, p 157-184.*

[33] Stone, M., "Cross-validation choice and assessment of statistical procedures", *Journal Royal of Statistical Society, Vol. 36, 1974, p 111-147.*

[34] Gunes, V., Menard, M., Loonis, P., Petit-Renaud, S., "Combination, cooperation and selection of classiers", *A state of the art. International Journal of Pattern Recognition, Vol. 17, 2003, p 1303-1324.*

Non-stationary self-consistent acoustic objects as atoms of voiced speech

Friedhelm R. Drepper

Forschungszentrum Jülich GmbH, 52425 Jülich, Germany, f.drepper@fz-juelich.de

Voiced segments of speech are assumed to be composed of non-stationary voiced acoustic objects which are generated as stationary (secondary) response of a non-stationary drive oscillator and which are analysed by introducing a self-consistent part-tone decomposition. The self-consistency implies that the part-tones (of voiced continuants) are suited to reconstruct a topologically equivalent image of the hidden drive (glottal master oscillator). As receiver side image the fundamental drive (FD) is suited to describe the broadband voiced excitation as entrained (synchronized) and/or modulated primary response and to serve as low frequency part of the basic time-scale separation of auditive perception, which separates phone or timbre specific processes from intonation and prosody. The self-consistent time-scale separation avoids the conventional assumption of stationary excitation and represents the basic decoding step of the phase-modulation transmission-protocol of self-consistent (voiced) acoustic objects. The present study is focussed on the adaptation of the contours of the centre frequency of the part-tone filters to the chirp of the glottal master oscillator.

1. INTRODUCTION

Many methods being conventionally used to analyze non-stationary (speech) signals like short time Fourier analysis or wavelet analysis [1, 2] are based on a complete and orthogonal decomposition of the signal into elementary components. The amplitudes of such components can be interpreted in terms of a time-frequency energy distribution. The elementary components are preferentially chosen as near optimal time-frequency atoms, which are each characterized by a reference time t_0 and angular frequency Ω_0. As a characteristic feature of wavelet analysis, the time-frequency atoms are chosen on different time scales. Time-frequency atoms are wave packets which are optimized to describe simultaneously event (particle) and wave type properties of non-stationary wave processes [3]. Their most general form can be written as second order logarithmic expansion of a complex signal around the reference time t_0 resulting in a complex Gaussian of the form

$$S_G(t) \approx \exp(-\frac{(t-t_0)^2}{2\sigma^2} + i\Omega_0(t-t_0)(1+c/2(t-t_0))) \quad (1a)$$

Contrary to the conventional one [3], this parametric set of time-frequency atoms is characterized by a *quadratic* trend phase or a linear trend phase velocity (angular frequency)

$$\omega_{0,t} = \Omega_0(1+c(t-t_0)) \quad (1b)$$

with relative chirp rate *c*. Due to their neglect of the chirp parameter, the time-frequency atoms of short time Fourier analysis and wavelet analysis are preferentially aimed at linear time invariant (LTI) systems [1] (with a time periodic deterministic skeleton). In contrast to the latter approaches, the present one is aimed at non-stationary acoustic objects which represent a superposition of time-frequency atoms with chirped angular frequencies. The general aim, however, is not a complete and orthogonal decomposition of the speech signal, but a (potentially incomplete) decomposition into part-tones which can be interpreted as topologically equivalent images of plausible underlying acoustic modes [4-8]. The part-tones are generated by bandpass filters with impulse responses which represent optimal or near optimal time-frequency atoms. The preference of time-frequency atoms of the form (1a) results from the aim to generate part-tones with a maximal time resolution, which is compatible with a frequency resolution being necessary to isolate a sufficient number of topologically equivalent images of the underlying acoustic modes.

Like in auditory scene analysis, an *a priori* knowledge about the behaviour of the underlying acoustic modes can be used to remove a potential ambiguity of the unknown acoustic object parameters (in particular of the time course of the centre frequencies of the bandpass filters). In case of voiced speech it is "known" *a priori* that the common origin of the acoustic modes (the pulsed airflow through the glottis) and the nonlinearity of the aero-acoustic dynamics in the vocal tract lead to a characteristic phase locking of the acoustic modes [5-8].

In the situation of signal analysis the detection of a strict ($n:n'$) synchronization of the phases of *a priori* independent part-tone pairs (with non-overlapping spectral bands) represents a phenomenon, which has a low probability to happen by chance, in particular, when the higher harmonic order *n* has a large value. For such part-tone pairs it can therefore be assumed that there exists an uninterrupted causal link between those part-tones, including the only plausible case of two uninterrupted causal links to a common drive, which can be identified as a glottal master oscillator [4-8]. Since the ($n:n'$) phase-locking with $n \neq n'$ is generated by the *nonlinear* coupling of the acoustic modes to the glottal oscillator, a stable synchronization of *a priori* independent part-tone phases can be taken as a confirmation of topological equivalence between these part-tones and respective acoustic modes in the vocal tract of the transmitter.

Based on the a priori knowledge about the phase locking of the acoustic modes, the phase velocity contours of the part-tones can be assumed to be centered around harmonic mul-

tiples ($2\pi h$ with integer h) of the frequency contour of the glottal oscillator. A cluster analysis of harmonically normalized part-tone phase velocity contours can thus be used to identify a consistent set of part-tone phases, which is suited to reconstruct a unique phase velocity of the fundamental drive [6-8]. The present study is focussed on the construction of centre frequency contours of the bandpass filters which are consistent with the corresponding part-tone phase velocity contours. Self-consistently reconstructed part-tone phases are proven to be suited for a phase-modulation transmission protocol of voiced speech.

2. VOICE ADAPTED PART-TONES

In case of the characteristic isolated pulse type events of stop consonants, single time-frequency atoms are potentially suited to describe such events. For real time analysis of voiced continuants it is unavoidable to generate part-tones which result from *causal* bandpass filters. The present study uses an all pole approximation of complex Γ-tone bandpass filters with approximately gamma-distribution like amplitudes of the impulse response [10]. For sufficiently high autoregressive order, the Γ-function like amplitude distribution guarantees a near optimal time-frequency atom property of the impulse responses. (That is why an autoregressive order (Γ-order) $\Gamma = 5$ will be used in the example instead of the more common choice $\Gamma = 4$ [11, 10].)

The choice of (roughly) audiological bandwidths for the part-tone decomposition has the effect that we can distinguish a lower range of part-tone indices characterized by guaranteed single harmonic (resolved) part-tones and a range of potentially multiple harmonic (unresolved) part-tones. In the resolved part-tone range $1 \le j \le 6$ the harmonic order h_j is identical to the part-tone index *j*. To avoid a substantial over-completeness (and *a priori* correlation between neighbouring part-tones) in the unresolved range $6 < j \le N$, the set of harmonic part-tones is pruned according to the respective equivalent rectangular bandwidths (ERB). A typical set of part-tones may have the harmonic orders $\{h_j\} = \{1,2,...,6,8,10,12,15,...\}$. (Diplophonic voice types may lead to rational winding numbers $h_j = n_j / m$ with a common subharmonic period number $m > 1$ [5-8].) In particular for speech segments, which correspond to nasals or vowels it is typical that some of the part-tones in the (a priori) unresolved range are also dominated by a single harmonic acoustic mode. The under-completeness of the part-tones in the (a priori) resolved range has a welcome noise suppression effect.

The all pole approximation of the gammatone filters has the advantage of a fast autoregressive algorithmic implementation [10]. For theoretical reasons we prefer its description in terms of a matrix recursion with a lower triangular matrix *L* of dimension Γ which plays the role of the cascade depth of the cascaded first order autoregressive filter,

$$L\,X_t \;=\; \lambda \exp(i\,\omega_t)\,X_{t-1} \;+\; e_1\,S_t \qquad (2a)$$

with input signal S_t being sampled at discrete times t, Γ-dimensional vectors $X_t = \{v_t, w_t, \cdots, z_t\}'$, $e_1 = \{1,0,\cdots,0\}'$, $X_0 = 0$ and matrix

$$L \;=\; \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ -1 & 1 & \ddots & & \vdots \\ 0 & -1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & -1 & 1 & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{pmatrix} \quad . \qquad (2b)$$

The scalar λ represents the damping factor of every first order autoregressive filter and is directly related to the ERB of the Γ-tone filter, $\lambda = \exp(-a_\Gamma\, ERB)$, the Γ-order dependent factor a_Γ being given e.g. in [10]. The complex phase factor $\exp(i\,\omega_t)$ defines the instantaneous centre filter frequency $F_t = \omega_t / 2\pi$ being simply related to the instantaneous angular velocity ω_t. The unusual feature is the time dependence of the angular velocity ω_t which will be specified later. The inverse of matrix *L* is the lower triangular matrix with ones on and below the diagonal. It can be used to obtain X_t as a power series of matrix L^{-1},

$$X_t \;=\; \sum_{t'=0}^{t} \prod_{k=t'+1}^{t} \exp(i\,\omega_k)\; \lambda^{t-t'} L^{-(t+1-t')}\, e_1\, S_{t'} \;. \qquad (3)$$

The filter output is represented by the last component z_t of vector X_t. Therefore we are interested in the matrix element in the lower left corner of any power of matrix L^{-1}. For the (n+1)$^{\text{th}}$ power this element can easily be obtained by complete induction as the ratio of three factorials $(n+\Gamma-1)!/(\Gamma-1)!n!$. Taking into account the additional dependence on the part-tone index *j,* the output of the (non-normalized) bandpass filter of part-tone *j* is thus obtained as

$$z_{j,t} \;=\; \sum_{t'=0}^{t} \exp\Big(i \sum_{k=t'+1}^{t} \omega_{j,k}\Big)\, \lambda_j^{t-t'}\, \frac{(t-t'+\Gamma-1)!}{(\Gamma-1)!\,(t-t')!}\, S_{t'}. \qquad (4)$$

For $j = 1,...,N$ the set of (normalized) part-tones can be interpreted as a highly over-sampled time-frequency decomposition of the speech signal $S_{t'}$ where the over-sampling is restricted to the time axis. The non-normalized part-tones (4) can be used to generate the part-tone phases (carrier phases)

$$\varphi_{j,t} \;=\; \arctan(im(z_{j,t}) / re(z_{j,t})) \qquad (5)$$

as well as the (harmonically) normalized part-tone phases $\varphi_{j,t} / h_j$ in the frequency range of the pitch. If the trajectory (contour) of the centre filter frequency $\omega_{j,k} / 2\pi$ is chosen as identical to the one of the instantaneous frequency $\omega_k' / 2\pi$ of a constant amplitude input signal $S_{t'} = A\exp(i \sum_{k=0}^{t'} \omega_k')$, the application of bandpass filter (4) generates the output

$$z_{j,t} \;=\; A\,\exp(i \sum_{k=0}^{t} \omega_k')\; \sum_{t'=0}^{t} \lambda_j^{t-t'}\, \frac{(t-t'+\Gamma-1)!}{(\Gamma-1)!\,(t-t')!} \quad . \qquad (6)$$

This filter output has the remarkable property that its instantaneous phase velocity is identical to the one of the input signal. For a given filter frequency contour, other input signals experience a damping due to interference of the phase factors. For a given input frequency contour, other filter frequency contours generate a phase distortion of the output. In the limit $t \to \infty$, the sum in equation (6) represents an asymptotic gain

factor $g_{j,\Gamma}$. Being exclusively dependent on the bandwidth parameter λ_j and the Γ order, the gain factors can be used to obtain the normalized part-tone amplitudes $a_{j,t} = |z_{j,t}| / g_{j,\Gamma}$.

For more general voiced input signals the determination of filter frequency contours, which are identical to frequency contours of some underlying acoustic modes, represents a non-trivial problem. Conventionally [14-17] the adjustment of the filter frequency contours of part-tones (or "sinusoidal components") is achieved by introducing a short-time stationary (zero-chirp) subband decomposition which is densely sampled with respect to frequency and by determining for each point in time local maxima of the amplitudes of the subbands with respect to frequency. In a second step the maximizing frequencies of consecutive points in time are tested, whether they are suited to form continuous frequency contours. Suitable maxima are joined to form weakly non-stationary contours and part-tones. It is well known that the non-stationarity of natural voiced speech leads to frequent death and birth events of such contours, even within voiced segments [14-16].

The present approach is aimed at *self-consistent* centre filter frequency contours which are chosen as identical (or as consistent) to the frequency contours of the respective part-tones (outputs). It is based on the assumption that sustained voiced signals are composed of one or several part-tones which can iteratively be disclosed and confirmed to be self-consistent, when starting from appropriate contours of the centre filter frequency. In a first step we restrict the self-consistency to a single part-tone. In this case the self-consistency is defined as the existence of a centre filter-frequency contour of the bandpass filter being used to generate the part-tone which can be obtained as stable invariant set of the iteration of two cascaded mappings, where the first mapping uses a filter-frequency contour (out of a basin of attraction of preliminary frequency contours) to generate a part-tone phase velocity contour and the second mapping relates this part-tone phase velocity contour to an update of the mentioned filter-frequency contour. Whereas the first mapping is given by part-tone filter (4) and phase definition (5), the second mapping is chosen according to the acoustic properties of the assumed underlying physical system.

The acoustic properties include the physical law *natura non facit saltus*. The resulting smoothing of the centre filter frequency is suited to improve the convergence properties of the adaptation. Being inspired by equation (1b) the smoothing step might simply be chosen as a linear approximation of the trend of the filter-frequencies within each analysis window. However, due to the time reversal asymmetry of the Γ-tone filters, a negative chirp rate leads to a singularity of the instantaneous period length of the impulse response at finite times. This singularity can be avoided, if the time dependence of centre filter frequency $\omega_{j,t}/2\pi$ of part-tone j is chosen separately depending on the sign of the (relative) chirp rate c_j. For negative chirp rate it is useful to assume alternatively a linear trend of the inverse of the respective centre filter frequency with a smooth transition at zero chirp rate

$$\omega_{j,k} = \begin{Bmatrix} \omega_{j,0}(1+c_j k) \\ \omega_{j,0}/(1-c_j k) \end{Bmatrix} \quad for \quad \begin{Bmatrix} c_j \geq 0 \\ c_j < 0 \end{Bmatrix}. \qquad (7)$$

As is well known, human pitch perception is not limited to the frequency range of the separable part-tones. In particular it is known that the modulation amplitudes (envelopes) of the higher frequency subbands play an important role in hear physiology and psychoacoustics [11, 12]. It is therefore plausible to extend the analysis of part-tone phases to the non-separable range, i.e. to phases, which can be derived from the envelopes of the part-tones. Being used preferentially for part-tones with unresolved harmonics, the envelope phases are determined e.g. as Hilbert phases of (appropriately scaled and smoothed) modulation amplitudes (envelopes) of part-tones.

To achieve a more uniform time evolution of the envelope phases and in agreement to well known results from hear-physiology and psycho-acoustics [11, 12], the normalized modulation amplitudes $a_{j,k} = |z_{j,k}| / g_{j,\Gamma}$ are submitted to a sublinear transformation (scaling) and smoothing prior to the determination of the Hilbert phases. It is common practice to choose a power law with an exponent in the range $\nu = 0.33$ [11, 12]. In contrast to the carrier phases (which do not need a correction due to their self-consistency as expressed in equation (6)) the envelope phases need a group delay correction of the respective part-tones. The part-tone index specific part of this correction has been derived from the maxima of the amplitude of the impulse responses of equation (4) [10]. The relative importance of the envelope phases is expected to increase, when the voice source changes from a modal (ideal) voice to a breathy one.

3. PART-TONES OF A SIMPLE PULSED EXCITATION

To demonstrate the generation of self-consistent part-tones of a non-stationary voiced acoustic object, a sequence of synthetic glottal pulses with a chirped frequency is chosen as input signal. For simplicity the pulses are chosen as constant amplitude saw teeth with a power spectrum, which is roughly similar to the one of the glottal excitation. The pulse shape is described by an impulse function [21] or wave shaper function [22] of the form

$$G(\psi_t) = \min\left(\mathrm{mod}(\psi_t, 2\pi), s(2\pi - \mathrm{mod}(\psi_t, 2\pi))\right), \quad (8)$$

where ψ_t represents the phase of an artificial glottal master oscillator [5-8]. The parameter s (chosen to be 6) determines the ratio of the modulus of the downhill slope of the glottal pulses to the uphill one. The chirp of the glottal oscillator is described by a time dependent phase velocity $\omega(t) = \dot{\psi}(t)$ which is chosen in analogy to equation (7), however, with potentially different chirp rate c' and initial phase velocity ω_0'. (In the specific example, the glottal chirp parameter c' is chosen to generate a doubling of the frequency (or period length) after about 25 periods.) The fundamental phase ψ' is obtained by integrating the analogue of equation (7) with respect to time t (replacing index k)

$$\psi(t) \;=\; \left\{ \begin{array}{l} \omega_0' \,(t + c'\, t^2/2) \\ -\,\omega_0' / c' \;\ln(1 - c' t) \end{array} \right\} \quad for \quad \left\{ \begin{array}{l} c' \geq 0 \\ c' < 0 \end{array} \right\} . \qquad (9)$$

In the situation of signal analysis, appropriate contours of the centre filter frequency of the part-tone specific bandpass filters have to be obtained iteratively from the observed signal. As part of the time scale separation step of the second mapping of the last section, we assume that these contours can be described (within the current rectangular window of analysis) by a simple smooth function of time chosen as indicated in equation (7). The part-tone adaptation of the filter-frequency contour of the bandpass filter of part-tone *j* can thus be achieved by estimating the parameters of equation (7). To reduce the dependence of the estimate on the size and position of the window of analysis (and/or to avoid the adaptation of the window length to the instantaneous period length), time scale separation ansatz (7) is extended by a 2π periodic function $P_j(\varphi_{j,t} / h_j)$ of the respective normalized part-tone phase

$$\dot{\varphi}_{j,t} / h_j \;=\; \alpha_j\, t \;+\; P_j(\varphi_{j,t} / h_j) \qquad (10a)$$

$$h_j / \dot{\varphi}_{j,t} \;=\; -\alpha_j\, t \;+\; P_j(\varphi_{j,t} / h_j)\,. \qquad (10b)$$

The 2π periodic function $P_j(\varphi)$ accounts for the periodic oscillations of the phase velocity around the long term trend being generated by the characteristic auto phase-locking and is approximated by an appropriate finite order Fourier series. The Fourier coefficients as well as the trend parameter α_j are obtained by multiple linear regression.

Within a voiced segment of speech the adaptation of the parameters is performed sequentially for successive analysis windows. The initial value of the centre filter frequency of the current window is therefore typically given as result of the adaptation of the filter chirp of the preceding analysis window. Thus we treat the latter parameter as given ($\omega_{j,0} \;=\; h_j\, \omega_0'$) and concentrate on the convergence properties of the chirp parameter of a filter-frequency contour. The adaptation of a single parameter can be represented graphically. To explain the approach to self-consistency we use a graph, which shows the trend parameter α_j of equations (10a or 10b) for several part-tone indices *j* as function of the common filter chirp rate *c*. To make figure 1 suited for the graphical analysis it gives the estimates of the relative trend $\alpha_j / (\omega_0' c')$ for the indices j = 2, 4, 6, 9 (corresponding to the sequence of the fixed points from bottom to top) as function of the relative filter chirp rate c / c'.

The iterative adaptation of the chirp parameter of each filter-frequency contour can be read off from figure 1 by an iteration of two geometric steps: Project horizontally from one of the described curves to the diagonal of the first quadrant (which indicates the line where the fixed points of the iteration are situated) and project vertically down (or up) to the curve again. As can be seen from figure 1, the chirp parameters of all four part-tones have a stable fixed point (equilibrium) within a well extended basin of attraction of the chirp parameter which exceeds the shown interval of the abscissa. The fixed points (corresponding to the more general invariant sets of the preceding section) indicate the final error of the filter chirp which depends not only on the part-tone index but also on the size of the analysis window (which was chosen to have a length of about five periods of the glottal process). Due to the simple least squares regression of equations (10a,b), the modulus of the trend α_j is systematically underestimated.

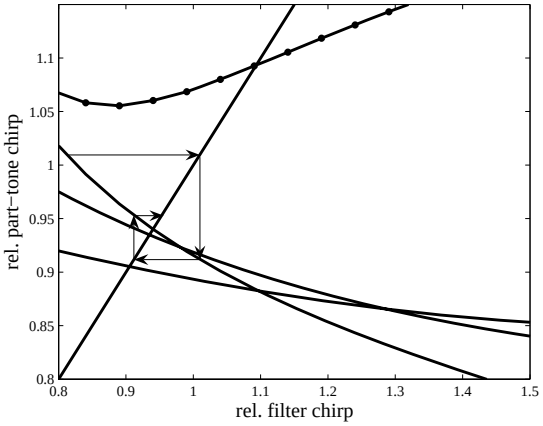


Figure 1: Estimated relative part-tone chirp rates as function of the relative chirp rate of the respective centre filter frequency, given for the envelope phase of part-tone 9 (circles, top) and the three carrier phases of part-tones 2, 4, and 6 (lines crossing the diagonal from bottom to top). All chirp rates are given relative to the chirp rate of the input sawtooth process defined in equations (8-9). The arrows and the diagonal of the first quadrant explain the algorithm, to determine the self-consistent centre filter frequencies.

4. MULTI PART-TONE STABLE ACOUSTIC OBJECTS

It is well known that human pitch perception can be trained to switch between analytic listening to a spectral pitch and synthetic listening to a virtual pitch [12, 16]. It is thus plausible to correlate the described single part-tone stable acoustic objects (with a macroscopic basin of attraction of the filter frequency contour or contour parameters) to outstanding part-tones, which are potentially perceived as spectral pitches by analytic listening [16, 27]. The number of stable invariant sets (fixed points) with a macroscopic basin of attraction depends in particular on the width of the power spectrum of the voiced signal. In the example of the last section a strong asymmetry of the sawteeth ($s >> 1$ in equation 10) favors the stability of higher order fixed points.

From psychoacoustic experiments it is also known that virtual pitch is a more universal and robust percept than spectral pitch [4, 16]. Based on the *a priori* assumption that the signal is generated by a voice production system, which generates several phase locked higher frequency acoustic modes, the observed (carrier or envelope) phase velocity of one part-tone might be used to adjust the centre filter frequency of other part-tones. This opens the possibility to use a more robust multi part-tone adaptation strategy which can be expected to converge even in cases with no single part-tone stability.

In analogy to the single part-tone stability of the last sections we relate multipart-tone stability of an acoustic object to

the existence of a fundamental phase velocity contour which can be obtained as stable invariant set of the iteration of three cascaded mappings, where the first mapping relates a preliminary fundamental phase velocity contour (out of a macroscopic basin of attraction) to a set of filter-frequency contours, the second mapping uses the set of filter-frequency contours to generate a corresponding set of part-tone phase velocity contours and the third mapping relates a subset of the part-tone phase velocity contours to update the fundamental phase velocity contour. The first mapping makes use of the characteristic auto-phase-locking of the voiced excitation ($\omega_{j,k} = h_j\, \omega_{0,k}$). The second mapping is given by filter (4) and phase (5) and the third mapping uses cluster analysis to identify invertible phase relations which are suited to reconstruct the phase velocity of the fundamental drive [4-8].

This way the contradiction between Rameau's concept of a *son fundamentale* or fundamental bass [20] and Seebeck's observation, that pitch perception does not rely on a fundamental acoustic mode as part of the heard signal [21], can be reconciled by replacing Rameau's *son fundamentale* by the described FD. Being an abstract order parameter and in need of a confirmation of its existence, the FD of a *multi-part-tone* voiced acoustic object cannot be reconstructed from a single part-tone alone. This qualifies the instantaneous fundamental phase velocity as acoustic correlate of *virtual* pitch perception. When reconstructed coherently for uninterrupted voiced speech segments, the fundamental phase becomes the central ingredient of a phase modulation decoder of voiced speech.

Contrary to the conventional psycho-acoustic theory [12, 16] (originating from Ohm and Helmholtz) which interprets the amplitudes of part-tones (with psycho-acoustically calibrated bandwidths) as primary acoustic cues, it is expected that the deviations of the phases of self-consistently determined part-tones from the synchronization manifold of the unperturbed ideally pulsed excitation have a comparable or higher relevance for acoustic perception than the corresponding amplitudes [17]. At the present state of analysis this hypothesis is mainly based on deductive arguments, which favor phase modulation features as more differentiated and robust cues for the distinction of the voiced phones of human speech as well as for the distinction of their speakers.

5. Conclusion

A transmission protocol of non-stationary self-consistent (voiced) acoustic objects is outlined, which are generated as stationary response of a non-stationary fundamental drive (FD) and which can self-consistently be decomposed into non-stationary part-tones. Self-consistent part-tones are characterized by phase velocities which are consistent with the centre filter frequencies being used to generate the part-tones. The second property of the self-consistent acoustic objects qualifies them as most elementary symbols of a voice transmission protocol which is centred on a time scale separation with a precise and robust decoding option. It is hypothesized that the self-consistent decomposition of speech segments, which are suited to transmit voiced continuants, leads to a subset of part-tones which shows generalized synchronization of their phases. The iterative identification of multi part-tone stable voiced acoustic objects relies on and enables a high precision reconstruction of a fundamental phase which can be confirmed as phase of a topologically equivalent image of a glottal master oscillator on the transmitter side. As topologically equivalent image on the receiver side, the self-consistent FD represents the long time scale part of the basic time scale separation known from human acoustic perception. The self-consistent reconstruction of the FD avoids the assumption of a frequency gap being necessary to justify the conventional assumption of a stationary or periodic voice source.

Acknowledgements: The author would like to thank M. Kob, B. Kröger, C. Neuschaefer-Rube and R. Schlüter, Aachen, J. Schoentgen, Brussels, A. Lacroix and K. Schnell, Frankfurt, and J. Rouat, Québec for helpful discussions.

6. References

[1] Rabiner L.R. and R.W. Schafer, "Digital Processing of Speech Signals",.Englewood Cliffs, NJ: Prentice-Hall (1978)
[2] Daubechies I., "Ten Lectures on Wavelets", SIAM, Philadelphia (1992)
[3] Gabor D, „Acoustic quanta and the theory of hearing", *Nature* **159**, 591-594 (1947)
[4] Drepper F.R., "Topologically equivalent reconstruction of instationary voiced speech", in C. Manfredi (editor), *MAVEBA 2003*, Firenze University Press (2004)
[5] Drepper F.R., „Selfconsistent time scale separation of non-stationary speech", *Fortschritte der Akustik-DAGA'05* (2005)
[6] Drepper F.R., "A two-level drive-response model of non-stationary speech signals", in M. Faundez-Zanuy et al. (Eds), NOLISP 2005, *LNAI* 3817, 125-138, Springer (2005)
[7] Drepper F.R., "Voiced excitation as entrained primary response of a reconstructed glottal master oscillator", *Interspeech 2005,* Lisboa (2005)
[8] Drepper F.R., „Stimmhafte Sprache als sekundäre Antwort eines selbst-konsistenten Treiberprozesses", *DAGA'06* (2006)
[10] Hohmann V., "Frequency analysis and synthesis using a Gammatone filterbank", *Acta Acustica* **10**, 433-442 (2002)
[11] Patterson R.D., "Auditory images: How complex sounds are represented", *J. Acoust. Soc. Jp*n. (E) **21**, 4 (2000)
[12] Moore B.C.J., "An introduction to the psychology of hearing", Academic Press (1989)
[14] McAulay R. and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation", *IEEE Trans. Acoust., Speech a. Signal Proc.* ASSP-**34**(4), 744-754 (1986)
[15] Heinbach W**.,** "Aurally adequate signal representation: The part-tone-time-pattern", Acustica **67**, 113-121 (1988)
[16] Terhardt E. „*Akustische Kommunikation*", Springer, Berlin (1998)
[17] Paliwal K.K. and B.S. Atal, "Frequency-related representation of speech", *Eurospeech 2003*, Genf (2003)
[20] Jean-Philippe Rameau, "Generation harmonique" (1737) reprinted in E. Jacobi (ed.), Complete Theoretical Writings Vol. 3, American Institute of Musicology (1967)
[21] August Seebeck, "Über die Definition des Tones", *Poggendorf's Annalen der Physik und Chemie* Vol. LXIII, pp 353-368 (1844)
[22] Schoentgen J., "Non-linear signal representation and its application to the modelling of the glottal waveform", *Speech Communication* **9**, pp. 189–201 (1990)

THE HARTLEY PHASE CEPSTRUM AS A TOOL FOR SIGNAL ANALYSIS

I. Paraskevas, E. Chilton,
Centre for Vision Speech and Signal Processing (CVSSP)
School of Electronics and Physical Sciences,
University of Surrey
Guildford, GU2 7XH, Surrey, UK

e-mail: paraskevas@env.aegean.gr,
E.Chilton@ee.surrey.ac.uk

M. Rangoussi
Department of Electronics
Technological Education Institute
of Piraeus
250, Thivon str., Aigaleo-Athens,
GR-12244, GREECE
Phone/Fax: +302105381222,6
e-mail: mariar@teipir.gr

1. Introduction

This paper proposes the use of the Hartley Phase Cepstrum as a tool for signal analysis. The phase of a signal conveys critical information, which is exploited in a variety of applications. The role of phase is particularly important for the case of speech or audio signals. Accurate phase information extraction is a prerequisite for speech applications such as coding, synchronization, synthesis or recognition. However, signal phase extraction is not a straightforward procedure, mainly due to the discontinuities appearing in it ('wrapping' effect). A variety of phase 'unwrapping' algorithms have been proposed to overcome this point, when the extraction of the accurate phase values is required. In order to extract the phase content of a signal for subsequent utilization, it is necessary to choose a function that can encapsulate it. In this paper we propose the use of the Hartley Phase Cepstrum (HPC).

2. The Hatrley Phase Cepstrum

In general, computation of the cepstrum of a signal belongs to a class of methods known as homomorphic deconvolution processes, [1]. A homomorphic process describes the invertible procedure in which a signal is transformed into another domain via an orthogonal transform Ξ, a non-linear process is applied to the transformed signal in the new domain, and the result is transformed back to the original domain, via the inverse transform, Ξ^{-1}:

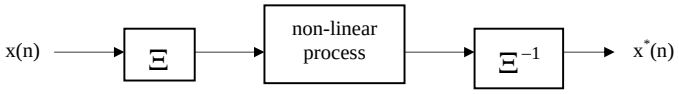


Figure 1: Summary of the homomorphic deconvolution process

In the special case where Ξ and Ξ^{-1} represent the *DTFT* (Discrete-Time Fourier Transform) and the *IDTFT* (Inverse Discrete-Time Fourier Transform), respectively, while the non-linear process is the evaluation of the Fourier phase spectrum,

$$\varphi(\omega) = \arctan\left(\frac{\Im(S(\omega))}{\Re(S(\omega))}\right) \quad (1)$$

where $\Re(S(\omega))$ and $\Im(S(\omega))$ are the real and imaginary components of the Fourier transform $S(\omega)$ of the signal $s(t)$, respectively, we obtain the so-called Fourier Phase Cepstrum, $c_F(\tau)$:

$$c_F(\tau) = IDTFT(\varphi(\omega)) \quad (2)$$

The Fourier phase spectrum (2nd stage of figure 1) experiences two categories of discontinuities. The first category of the discontinuities ('extrinsic') is related to the use of the *arctan* function and is overcome using the 'unwrapping' algorithm, [2]. The second category of discontinuities ('intrinsic') originates from the properties of the signal itself and is overcome with their compensation, [3]. Hence, for the Fourier case, the non-linear process (2nd stage of figure 1) can be divided into the three stages:

Figure 2: Stages of the non-linear part of the homomorphic deconvolution process applied to the Fourier case

For the Hartley Phase Cepstrum case, the first, the second and the third stages of figure 1 are the *DTHT* (Discrete-Time Hartley Transform), the evaluation of the Hartley phase spectrum [4], i.e.

$$Y(\omega) = \cos(\varphi(\omega)) + \sin(\varphi(\omega)) \quad (3)$$

and the *IDTHT* (Inverse Discrete-Time Hartley Transform), respectively. Hence, the Harley Phase Cepstrum is defined as:

$$c_H(\tau) = IDTHT(Y(\omega)) \qquad (4)$$

The Hartley phase spectrum (equation (3)), unlike its Fourier counterpart (equation (1)), does not have 'wrapping' ambiguities. Hence, it experiences only the 'intrinsic' category of discontinuities, which can be detected and compensated, [4].

So, for the Hartley case, the non-linear process (2nd stage of figure 1) can be divided in the following two stages:

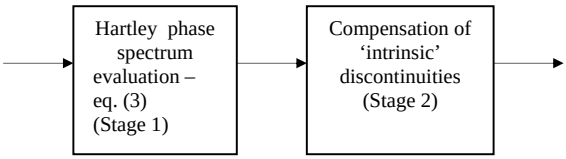


Figure 3: Stages of the non-linear part of the homomorphic deconvolution process applied to the Hartley case.

The proposed HPC is a signal feature that bears certain advantages over its Fourier counterpart, especially useful for practical applications in speech. These advantages are based on the properties of the respective spectra, which carry over to the cepstral domain thanks to the analytic relations that hold between the two domains. Localization ability and robustness to noise are two such advantages. As a simplified example of a signal, let us consider a pulse signal in the time domain. The Fourier Phase Cepstrum can identify only a single pulse, due to the ambiguities introduced by the use of the 'unwrapping' algorithm, whereas the Hartley Phase Cepstrum can indicate the location of a sequence of pulses, even for the case where noise is present [5]. Moreover, the HPC, unlike its Fourier counterpart, is more tolerant to noise – a property justified via the shape of the probability density function of the HPC, in the case where the time domain signal is pure Gaussian noise (section 3). Another property of the HPC, of interest in speech synthesis, is its invertibility: Unlike the Fourier case (figure 2), both stages of the non-linear process of the evaluation of the HPC (figure 3) are invertible, because the 'unwrapping' algorithm is not used.

3. Noise robustness of the HPC

The aim of this section is to show why the Hartley phase spectrum is more immune to noise as compared to the Fourier phase spectrum. To this end are employed the Probability Density Functions (PDFs) of the Hartley and of the Fourier phase spectra, in the special case of a pure Gaussian noise signal. The PDF of the Hartley phase spectrum is given by:

$$p_H(\beta) = \frac{1}{\pi\sqrt{2}\sqrt{1-\left(\frac{\beta}{\sqrt{2}}\right)^2}}\ ,\ -\sqrt{2} < \beta < \sqrt{2}\ , \qquad (5)$$

where β denotes the Hartley phase function values, $Y(\omega)$. (See [5] for the proof of eq. (5) – proof not shown here because of lack of space).

The shape of $p_H(\beta)$ is shown in figure 4 (left). It can be observed from figure 4 (left) that the peaks of this PDF are in its upper and lower range of the horizontal axis (i.e. $\sqrt{2}$ and $-\sqrt{2}$). However, the information content of the signal, in the Hartley phase spectrum, is encapsulated in the zero crossings with respect to the frequency axis rather than in the minimum / maximum values of the cosinusoidal signal (i.e. $\pm\sqrt{2}$), [5]. Consequently, noise mainly affects the higher and the lower domain of the Hartley phase spectrum values and hence, its information content (encapsulated in the zero-crossings, middle part of its domain) is less affected.

For the Fourier phase spectrum case though, assuming again a Gaussian noise signal, and if the *arctan* function is omitted from eq. (1), then the $p_F(\beta)$ is a Cauchy distribution, assuming that the real and the imaginary parts of the Fourier spectrum are independent, [6]. In this case:

$$p_F(\beta) = \frac{1}{\pi\sqrt{1+\beta^2}}\ ,\ -\infty < \beta < \infty \qquad (6)$$

where now β denotes the Fourier phase spectrum, $\varphi(\omega)$. The Cauchy distribution $p_F(\beta)$ (shown for $-5 < \varphi(\omega) < 5$ in figure 4 (right)), similarly to the Gaussian distribution, is symmetrical about $\beta = 0$ with its maximum value at $\beta = 0$. However, the Cauchy distribution falls more rapidly as $|\beta|$ increases and also its tails are heavier, compared to the Gaussian. Audio signals (e.g. speech, mechanical sounds etc.), convey a heavy noise additive component and hence, the PDFs of their phase spectra are similar to the Cauchy distribution for the Fourier case and to the distribution in eq. (5) for the Hartley case, respectively.

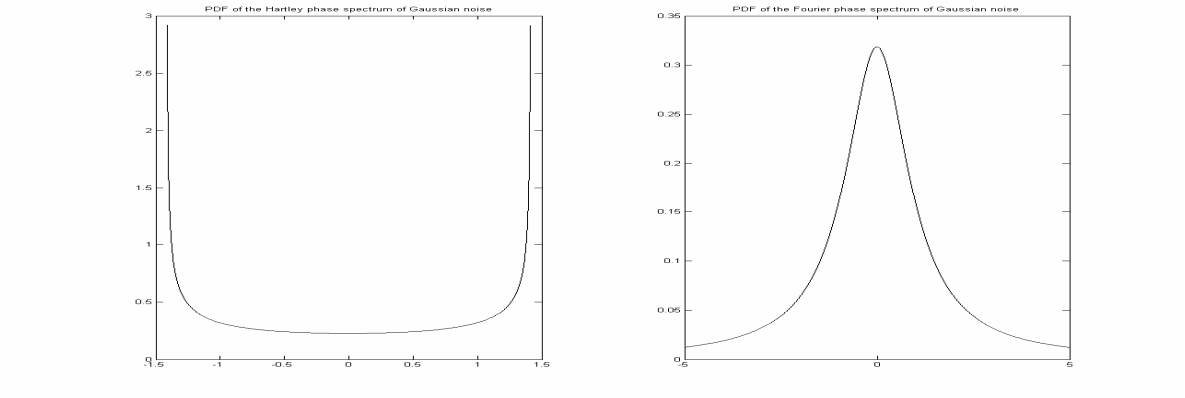


Figure 4: PDFs (a) of the Hartley phase spectrum, $Y(\omega)$, (left) and (b) of the Fourier phase spectrum, $\varphi(\omega)$, (right) for a pure Gaussian noise signal.

It should be noted here that, if the definition of the conventional Fourier phase spectrum is used (i.e. if the *arctan* function is not omitted in eq. (1)) then the PDF of the phase spectrum is no more a Cauchy, (see, eg., [6], chapter V.5); rather, it becomes uniform in $-\pi < \varphi(\omega) < \pi$.

Nevertheless, in either of the above two choices for the definition of the Fourier phase spectrum (i.e., either including the *arctan* function or not), the information content is distributed across the whole range of $\varphi(\omega)$ values and hence there does not exist a specific region of the PDF horizontal axis where the information content is mainly encapsulated. This constitutes a major difference to the case of the Hartley phase spectrum, where, as pointed out earlier, information lies mainly towards the two endpoints of the PDF range. This difference in the shapes of the respective PDFs justifies the relative noise immunity of the proposed HPC.

4. Conclusions

The phase of a signal as a function of frequency conveys meaningful information that is particularly useful for speech or audio signals. Accurate phase extraction is crucial in various speech processing applications, such as localization, synchronization, coding, etc. The major disadvantage of the computation of the phase spectrum via the Fourier transform is the heuristics employed of the compensation of the 'extrinsic' discontinuities ('wrapping' ambiguities). The effect of the 'wrapping' ambiguities is more severe in the case where noise is present. The Hartley phase spectrum, on the other hand, is advantageous as (a) it does not convey 'extrinsic' discontinuities and (b) due to its structure, it is less affected by the presence of noise, as justified through comparison of the shapes of the respective PDFs. As signal localization applications show, the phase content of a signal is encapsulated in a more efficient and easy to identify manner in the Hartley Phase rather than in the Fourier Phase Cepstral function. Hence, the Hartley Phase Cepstrum is proposed here as a promising and viable substitute to its Fourier counterpart.

References

[1] 'Theory and Applications of Digital Signal Processing', Rabiner, L.R., Gold, B., ch. 12, Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

[2] 'A new phase unwrapping algorithm', Tribolet, J., Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on, vol. 25, no. 2, Apr 1977, pp. 170 – 177.

[3] 'Combination of Magnitude and Phase Statistical Features for Audio Classification', Paraskevas, I., Chilton, E., Acoustics Research Letters Online (ARLO), Acoustical Society of America, July 2004.

[4] 'Audio Classification Using Features Derived From The Hartley Transform', Paraskevas, I., Chilton, E., Rangoussi, M., 13th Int. Conference on Systems, Signals and Image Processing (IWSSIP'2006), Budapest, Hungary, September 2006.

[5] 'Phase as a Feature Extraction Tool for Audio Classification and Signal Localisation', Paraskevas I., Ph.D. thesis, University of Surrey, 2005.

[6] 'Introduction to the theory of statistics', Mood, A.M., Graybill F.A. and Boes, D.C., MacGraw-Hill International, 1974.

Acknowledgement

Authors gratefully acknowledge Dr. Alex Kadyrov of the CVSSP, University of Surrey, for his contribution in the derivation of eq. (5).

Quantitative perceptual separation of two kinds of degradation in speech denoising applications

Anis Ben Aicha and Sofia Ben Jebara

Unité de recherche TECHTRA
Ecole Supérieure des Communications de Tunis, 2083 Cité El-Ghazala/Ariana, TUNISIE
anis_ben_aicha@yahoo.fr,sofia.benjebara@supcom.rnu.tn

Abstract

Classical objective criteria evaluate speech quality using one quantity which embed all possible kind of degradation. For speech denoising applications, there is a great need to determine with accuracy the kind of the degradation (residual background noise, speech distortion or both). In this work, we propose two perceptual bounds UBPE and LBPE defining regions where original and denoised signals are perceptually equivalent or different. Next, two quantitative criteria PSANR and PSADR are developed to quantify separately the two kinds of degradation. Some simulation results for speech denoising using different approaches show the usefulness of proposed criteria.

1. Introduction

Evaluation of denoised speech quality can be done using subjective criteria such as MOS (Mean Opinion Score) or DMOS (Degradation MOS) [1]. However, such evaluation is expensive and time consuming so that, there is an increasing interest in the development of robust quantitative speech quality measures that correlate well with subjective tests. Objective criteria can be classified according to the domain in which they operate. We relate for example the Signal to Noise Ratio (SNR) and segmental SNR operating in time domain [2], the Cepstral Distance (CD) and Weighted Slope Spectral distance (WSS) operating in frequency domain [2] and Modified Bark Spectral Distortion (MBSD) operating in perceptual domain [3]. Perceptual measures are shown to have the best chance of predicting subjective quality of speech and other audio signals since they are based on human auditory perception models.

The common point of all objective criteria is their ability of evaluating speech quality using a single parameter which embed all kind of degradations after any processing. Indeed, speech quality measures are basing their evaluation on both original and degraded speeches according to the following application C:

$$C: \begin{array}{l} \mathbb{E}^2 \longrightarrow \mathbb{R} \\ (x,y) \longmapsto c \end{array} \quad (1)$$

where $\mathbb{E}$ denotes the time, frequency or perceptual domain. x (resp. y) denotes original speech (resp. observed speech altered by noise or denoised speech after processing) and c is the score of the objective measure.

Mathematically, C is not a bijection from $\mathbb{E}^2$ to $\mathbb{R}$. It means that it is possible to find a signal y' which is perceptually different from y but has the same score than the one obtained with y ($c(x,y) = c(x,y')$). We relate for example the case of an original signal x which is corrupted by an additive noise to construct the signal y. Then, x is coded and decoded using a CELP coder to obtain the signal y'. It is obviously that the degradation noticed in both y and y' are not the same. Degradation of y is heard as a background noise and the degradation of y' is perceptually heard as distortion of original signal. However, in a previous work, we show that they have the same SNR [4].

In this paper, we aim improving speech quality evaluation by separating two kinds of degradation which are the additive residual noise and the speech distortion. Each degradation will be evaluated using its adequate criterion so that the non bijection C will be avoided and replaced by a bijection one characterized by a couple of outputs instead of a single output. Moreover, thanks to the advantage of perceptual tools in the evaluation of speech quality, the new couple of criteria will be based on auditor properties of human ear.

2. Study context: speech denoising

Before defining novel criteria of speech quality evaluation, let's define the different kinds of degradation altering speech. Without loss of generality, we consider the speech denoising application and we use spectral denoising approaches. They are viewed as a multiplication of noisy speech spectrum $Y(m,k)$ by a real positive coefficient filter $H(m,k)$ (see for example [5]). The estimated spectrum of clean speech is written

$$\hat{S}(m,k) = H(m,k)Y(m,k), \quad (2)$$

where m (resp. k) denotes frame index (resp. frequency index).

The estimation error spectrum $\xi(m,k)$ is given by

$$\xi(m,k) = S(m,k) - \hat{S}(m,k). \quad (3)$$

We assume that speech and noise are uncorrelated. Thus, the estimated error power spectrum is given by

$$\begin{array}{rl} E\{|\xi(m,k)|^2\} = & [H(m,f)-1]^2 E\{|S(m,k)|^2\} \\ & +H(m,k)^2 E\{|N(m,k)|^2\}, \end{array} \quad (4)$$

where $|N(m,k)|^2$ denotes the noise power spectrum.

Since $0 < H(m,k) < 1$, the first term of Eq. 4 expresses the 'attenuation' of clean speech frequency components. Such degradation is perceptually heard as a distortion of clean speech. However, the second term expresses the residual noise which is perceptually heard as a background noise. Since, it is additive, it is possible to formulate it as an 'accentuation' of clean speech frequency components.

3. Proposed perceptual characterization of audible degradation

We aim to perceptually characterize the degradation altering denoised speech. Hence, auditory properties of human ear are

considered. More precisely, the masking concept is used: a masked signal is made inaudible by a masker if the masked signal magnitude is below the perceptual masking threshold MT. In our case, both degradation can be audible or inaudible according to their position regarding the masking threshold. We propose to find decision rules to decide on the audibility of residual noise and speech distortion by using the masking threshold concept. If they are audible, the audibility rate will be quantified according to the proposed criterion. There are many techniques to compute masking threshold MT, we use in this paper Johnston model well known for its simplicity and well used in coding context [6].

3.1. Perceptual characterization of audible noise

According to MT definition, it is possible to add to the clean speech power spectrum, the MT curve (considered as a 'certain signal') so that the resulting signal (obtained by inverse FFT) has the same audible quality than the clean one. The resulting spectrum is called *Upper Bound of Perceptual Equivalence* "$UBPE$" and is defined as follows

$$UBPE(m,k) = \Gamma_s(m,k) + MT(m,k), \tag{5}$$

where $\Gamma_s(m,k)$ is the clean speech power spectrum.

When some frequency components of the denoised speech are above $UBPE$, the resulting additive noise is heard.

3.2. Perceptual characterization of audible distortion

By duality, some attenuations of frequency components can be heard as speech distortion. Thus, by analogy to $UBPE$, we propose to calculate a second curve which expresses the lower bound under which any attenuation of frequency components is heard as a distortion. We call it *Lower Bound of Perceptual Equivalence* "$LBPE$". To compute $LBPE$, we used the audible spectrum introduced by Tsoukalas *and al* for audio signal enhancement [7]. In such case, audible spectrum is calculated by considering the maximum between clean speech spectrum and masking threshold.

When speech components are under MT, they are not heard and we can replace them by a chosen threshold $\sigma(m,k)$.
The proposed $LBPE$ is defined as follows

$$LBPE(m,k) = \begin{cases} \Gamma_s(m,k) & \text{if } \Gamma_s(m,k) \geq MT(m,k) \\ \sigma(m,k) & \text{otherwise .} \end{cases} \tag{6}$$

The choice of $\sigma(m,k)$ obeys only one condition $\sigma(m,k) < MT(m,k)$. During this work, we choose it equal 0 dB.

3.3. Usefulness of $UBPE$ and $LBPE$

Using $UBPE$ and $LBPE$, we can define three regions characterizing the perceptual quantity of denoised speech: frequency components between $UBPE$ and $LBPE$ are perceptually equivalent to the original speech components, frequency components above $UBPE$ contain a background noise and frequency components under $LBPE$ are characterized by speech distortion. This characterization constitutes our idea to identify and detect audible additive noise and audible distortion. As illustration, we present in Fig. 1 an example of speech frame power spectrum and its related curves $UBPE$ (upper curve in bold line) and $LBPE$ (bottom curve in dash line). The clean speech power spectrum is, for all frequencies index, between the two curves $UBPE$ and $LBPE$. We remark that the two curves are the same for most peaks. It means that for these frequency intervals, any kind of degradation altering speech will be audible. If it quite over $UBPE$, it will be heard as background noise. In the opposite case, it will be heard as speech distortion.

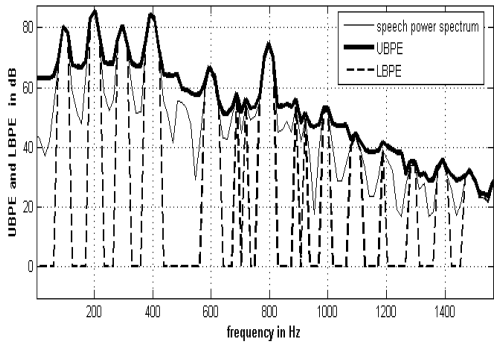


Figure 1: An example of $UBPE$ and $LBPE$ in dB of clean speech frame.

4. Audible degradation estimation

4.1. Audible additive noise PSD estimation

Once $UBPE$ calculated, the superposition of denoised signal power spectrum and $UBPE$ leads to separate two cases. The First one corresponds to the regions of denoised speech power spectrum which are under $UBPE$. In such case, there is no audible residual noise. In the second case, some denoised speech frequency components are above $UBPE$, the amount above $UBPE$ constitutes the audible residual noise. As illustration, we represent in Fig.2 an example of denoised speech power spectrum and its related $UBPE$ curve calculated from clean speech. The used denoising approach is spectral subtraction [5]. From Fig.2, we notice that frequency regions between 1 kHz and 2 kHz are above $UBPE$, they hence contain residual audible noise. In term of listening tests, such residual noise is annoying and constitutes in some cases the musical noise. Such musical noise is well popular and constitutes the main drawback of spectral subtraction.

Once the $UBPE$ is calculated, it is possible to estimate the audible power spectrum density of residual noise using a simple subtraction when it exists. Hence, the residual noise power spectrum density PSD is written

$$\Gamma_n^p(m,k) = \begin{cases} \Gamma_{\hat{s}}(m,k) - UBPE(m,k) \\ \text{if } \Gamma_{\hat{s}}(m,k) > UBPE(m,k) \\ 0 \qquad \text{otherwise} \end{cases} \tag{7}$$

where $\Gamma_{\hat{s}}(m,k)$ denotes the PSD of denoised speech and the suffix p designs the perceptually sense of the PSD.

4.2. Audible speech distortion PSD estimation

We use the same methodology as the one used for residual background noise. We represent in Fig.3 an example of denoised speech power spectrum and its related curve $LBPE$ calculated from the clean speech. We notice that some regions are under $LBPE$ (for example regions between 1.5 kHz and 2 kHz), they

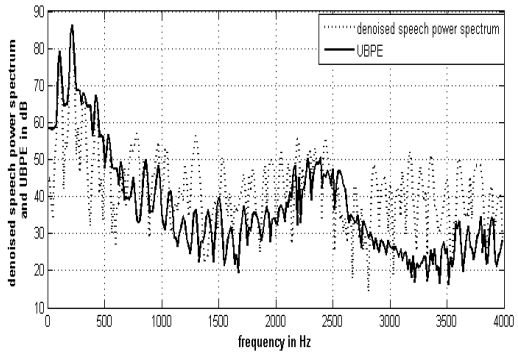


Figure 2: Superposition of a denoised speech power spectrum and its related clean speech $UBPE$.

hence constitute the audible distortion of the clean speech. In term of listening tests, they are completely different from residual background noise. They are heard as a loss of speech tonality.
It is possible to estimate the audible distortion PSD Γ_d^p as follows

$$\Gamma_d^p(m,k) = \begin{cases} LBPE(m,k) - \Gamma_{\hat{s}}(m,k) \\ \text{if } \Gamma_{\hat{s}}(m,k) < LBPE(m,k) \\ 0 \qquad \text{otherwise .} \end{cases} \tag{8}$$

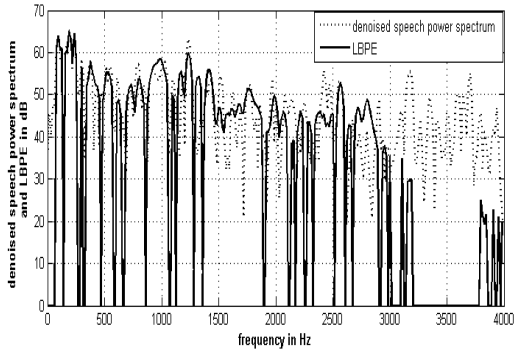


Figure 3: Superposition of a denoised speech frame and its related clean speech $LBPE$.

5. Audible degradation evaluation

In this section, we detail the proposed approach to quantify separately the two kinds of degradation. The assessment of the denoised speech quality by means of two parameters permits to overcome the problem of non bijection of classic objective evaluation and to better characterize each kind of speech degradation. Hence, instead of the application defined in Eq. 1, we develop a novel application from perceptual domain to $\mathbb{R}^2$

$$\begin{aligned} C: \quad & \mathbb{E}^2 \longrightarrow \mathbb{R}^2 \\ & (x,y) \longmapsto (PSANR, PSADR) \end{aligned} \tag{9}$$

where $PSANR$ and $PSADR$ are two parameters related respectively to the residual noise and the distortion.

The definition of $PSANR$ and $PSADR$ is inspired from the SNR definition which is the ratio between signal energy and noise energy. Thanks to Parseval theorem it can be calculated in frequency domain. Moreover, since the $UBPE$ and $LBPE$ are perceptually equivalent to the original signal, the proposed definition uses the energy of $UBPE$ and $LBPE$ instead of the energy of the clean speech. The time domain signal related to $UBPE$ is called "upper effective signal" whereas the time domain signal related to $LBPE$ is called "lower effective signal". In the following subsection, we define the proposed criteria.

5.1. Perceptual noise criterion PSANR

The perceptual residual noise criterion is defined as the ratio between the upper effective signal which is the $UBPE$ and the audible residual noise. The Perceptual Signal to Audible Noise Ratio $PSANR(m)$ of frame m is calculated in frequency domain (due to the Parseval theorem) and it is formulated as follows

$$PSANR(m) = \frac{\sum_{k=1}^{N} UBPE(m,k)}{\sum_{k=1}^{N} \Gamma_n^p(m,k)}. \tag{10}$$

5.2. Perceptual distortion criterion PSADR

By the same manner, we define the Perceptual Signal to Audible Distortion Ratio $PSADR(m)$ of frame m as a ratio between the lower effective signal which is $LBPE$ and the audible distortion. The $PSADR(m)$ is given by:

$$PSADR(m) = \frac{\sum_{k=1}^{N} LBPE(m,k)}{\sum_{k=1}^{N} \Gamma_d^p(m,k)}. \tag{11}$$

5.3. PSANDR criteria

to compute the global $PSANR$ and $PSADR$ of the total speech sequence, we are referred to the segmental SNR SNR_{seg} thanks to its better correlation with subjective tests when compared to the traditional SNR. The principle of segmental SNR consists on determining the SNR for each frame $SNR(m)$ and then calculating their geometric mean over the total number of frames $SNR_{seg} = \sqrt[N]{\prod_m^N SNR(m)}$ [2]. Moreover, since the SNR and SNR_{seg} are usually expressed in dB. The geometric mean is equivalent to the arithmetic mean in log domain.
Using this approach, we compute the global $PSANR$ and $PSADR$ for a given sequence of speech. Next, the couple ($PSANR$,$PSADR$) defines the new criterion to evaluate both kinds of degradation. We call it *Perceptual Signal to Audible Noise and Distortion Ratio* "$PSANDR$".

6. experimental results

6.1. Test signals

To show the ability of $PSANDR$ to take into account the perceptual effect of an additive noise, we add artificial noise, constructed from the masking threshold by multiplying it with a factor $\alpha \geq 0$ ($y(n) = s(n) + \alpha MT(n)$). In Fig.4, we represent the evolution of SNR_{seg}, $PSANR$ and $PSADR$ versus α. For the range of α between 0 and 1, SNR_{seg} decreases which means that there is a degradation of speech. This fact is true in term of signal to noise ratio but not true in term of perceptual sense, because the power of added artificial noise don't overtake MT. With $PSANR$, the amount of audible noise is

null (see Eq. 7) and the $PSANR$ is infinity which is truncated to 35 dB in our simulations. For $\alpha > 1$, the background noise becomes audible and the $PSANR$ decreases as α increases but remains above SNR_{seg}. This is explained by the ability of the clean speech to mask a certain portion of the added noise.

We notice that for any value of α, the second term $PSADR$ is still constant and is equal to 35 dB. In fact, there is no distortion of the clean speech and the only audible degradation is the background noise.

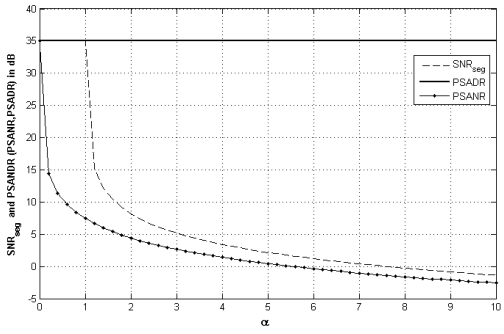


Figure 4: Evolution of SNR_{seg}, $PSANR$ and $PSADR$ versus α in case of additive noise.

6.2. real signals

Let's now compare some denoising techniques by means of the new objective criteria. We propose to denoise a corrupted signal, by gaussian noise with $SNR = 0$ dB, using the following techniques:

- Classical wiener filtering [5].
- Perceptual filtering proposed by Gustafsson *and al* in [8] which consists in masking the residual noise and allowing a variable speech distortion.
- Modified wiener technique [9]. In this technique, the shape of the tones is used as a selective parameter to detect and eliminate musical tones.

Evaluation of denoising quality is done using classic objective criteria (segmental SNR, WSS, MBSD) and the proposed PSANDR. Results are resumed in Tab.1. In term of SNR_{seg}, the used techniques are comparable even if there is a little improvement noticed with perceptual technique. But, subjective tests show that the denoised signals are completely different. Using WSS criterion, the best score is obtained with perceptual technique and it is nearly equal to the noisy speech score. Although, subjective tests show that the two signals are perceptually different. Indeed, the denoised speech using perceptual technique is heard as distorted version of clean speech and not as clean speech with background noise. In term of MBSD, the perceptual technique is also the best. However, this technique is characterized by a loss of the speech tonality comparing to wiener technique. Thus, we can see that classic evaluation tools don't give any idea of the kind and nature of the degradation of the signals. $PSANR$, giving idea about residual noise, shows that perceptual technique is the best one regarding noise attenuation. $PSADR$, determining the distortion of the denoised signals, shows that the important distortion is obtained using perceptual technique. These observations are confirmed by subjective tests.

Table 1: Evaluation of denoised signals.

	SNR_{seg} dB	WSS	MBSD	PSANR dB	PSADR dB
noisy speech	-4.30	46.07	2.32	-3.90	17.27
wiener technique	1.05	74.25	0.28	5.04	**7.53**
modified wiener	1.13	69.63	0.19	5.54	7.01
perceptual technique	**1.62**	**45.41**	**0.15**	**12.71**	6.93

7. Conclusion

The spectral and perceptual analysis of the degradation, in the case of denoised speech, imposes to separate between residual noise and signal distortion. We first propose two curves $UBPE$ and $LBPE$ to calculate the audible residual noise and audible distortion. Next, two parameters $PSANR$ and $PSADR$ characterizing the two kinds of degradation are developed. Simulation results comparing different denoising approaches and classical objective measures, show a better characterization of degradation nature of denoised signal. The calculation of the degree of correlation of the proposed criteria with MOS criterion constitutes the perspectives of our work.

8. References

[1] Recommandation UIT-T P.800. Mthodes d'valuation subjective de la qualité de transmission, 1996.

[2] J.H.L. Hansen and B.L. Pellom "An effective quality evaluation protocol for speech enhancement algorithms " Int. Conf. on Spoken Language Processing ICSLP, Austria 1998.

[3] W. Yang, M. Benbouchta and R. Yantorno, "Performance of the modified bark spectral distortion as an objective speech measure," Proc. Int. Conf. on Acoustics, Speech and Signal Processing ICASSP, vol 1, pp. 541-544, 1988.

[4] A. Ben aicha et S. Ben Jebara, "Caractérisation perceptuelle de la dégradation apportée par les techniques de débruitage de la parole," submited in Traitement et Analyse de l'Information Méthodes et Applications TAIMA, Tunisia 2007.

[5] J.S Lim and A.V Oppenheim, Enhancement and Bandwidth Compression of Noisy Speech, in Proc. IEEE, vol. 67, pp. 1586-1604, 1979.

[6] J. D. Johnston, "Transform coding of audio signal using perceptual noise criteria," IEEE J. Select. Areas Commun, vol 6, pp. 314-323, 1988.

[7] D. E. Tsoukalas, J. Mourjopoulos and G. Kokkinakis, "Speech enhancement based on audible noise suppression," IEEE Trans. Speech and Audio Processing, vol. 5, no. 6, pp. 497- 514, November 1997.

[8] S. Gustafsson, P. Jax and P. Vary, "A novel psychoacoustically motivated audio enhancement algorithm preserving backround noise characteristics," in Proc. Int. Conf. Acoustics, Speech, and Signal Processing, Seattle, WA, pp. 397-400, May 1998

[9] A. Ben Aicha, S. Ben Jebara and D. Pastor "Speech denoising improvement by musical tones shape modification," International Symposium on Communication, Control and Signal Processing ISCCSP, Morocco 2006.

Threshold Reduction for Improving Sparse Coding Shrinkage Performance in Speech Enhancement

Neda Faraji, S. M. Ahadi*, S. Saloomeh Shariati***

Department of Electrical Engineering
* Amirkabir University of Technology, **Iran University of Science and Technology,
Tehran, Iran
nfaraji@cic.aut.ac.ir , sma@aut.ac.ir, ssaloomeh_shariati@ee.iust.ac.ir

Abstract

In this paper, we modify the Sparse Coding Shrinkage (SCS) method with an appropriate optimal linear filter (Wiener filter) in order to improve its efficiency as a speech enhancement algorithm.
SCS transform is only applicable for sparse data and speech features do not have this property in either time or frequency domains. Therefore we have used Linear Independent Component Analysis (LICA) to transfer the corrupted speech frames to the sparse code space in which noise and speech components are separated by means of a shrinkage function. Before employing SCS, Wiener filtering was applied on the ICA components to reduce noise energy and consequently the SCS shrinkage threshold. Experimental results have been obtained using connected digit database TIDIGIT contaminated with NATO RSG-10 noise data.

1. Introduction

The primary purpose of noise compensation methods applied in the context of speech processing is to reduce the effect of any signal which is alien to and disruptive of the message and to extract original speech as pure as possible. Depending on the application, speech enhancement methods aim at speech quality improvement and or speech or speaker recognition. Some common noise compensation methods are a) Spectral Subtraction, b) least mean square (LMS), adaptive filtering, c) filter-based parametric approaches , d) Hidden Markov Model (HMM)-based speech enhancement techniques. Wavelet transform has also been employed in speech enhancement systems during recent years [1].
In this paper we focus on modifying a relatively new method that is Sparse Coding Shrinkage. It has been used in [2, 3] for image denoising and in [4] for speech enhancement. The advantages of this method with respect to other popular methods can be summarized as: 1) Most algorithms apply Fourier transform, discrete Fourier transform or Karhunen-Loeve transform which facilitate the estimation of the clean speech model parameters. However, SCS is based on a data-driven transformation that is highly conformed to the structure of clean speech data. 2) Most methods just amend the distorted amplitude of the speech signal leaving the phase unprocessed. On the other hand, experimental results have shown that phase parameter plays a relatively important role in speech quality [5]. In the SCS method, the speech frame is uniformly processed without need to separating the amplitude and phase data. Independent Component Analysis is a basic method for blind source separation. In Linear Independent Component Analysis (LICA), the goal is to transfer the observed data to the independent source space. Assuming that all independent sources are supergaussian, ICA technique will be equivalent to sparse coding method [2].
Denoising process includes an offline training stage in which clean speech frames are employed for estimating the ICA transform matrix and required parameters of the shrinkage function. These estimated functions can then be used to extract the clean components from the noisy ones.
The organization of the paper is as follows: First we explain in detail each of the methods used for speech enhancement in section 2. Section 3 demonstrates the noise reduction capability of the proposed algorithm through the computer simulations. Finally, conclusions are given in the last section.

2. Algorithms used in the proposed enhancement method

Sparse Coding Shrinkage is carried out in 2 stages:
1) Transferring data to the sparse coding space that is performed using ICA in this paper.
2) Computing Shrinkage function for each sparse code that can be achieved through Maximum Likelihood Estimation.

In this section we have provided details about the employed algorithms.

2.1. Independent Component Analysis

Linear Independent component analysis mixing model can be formulated as [6]:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{1}$$

We can rewrite the Eq. (1) as:

$$\mathbf{x} = \sum_i \mathbf{a}_i s_i \tag{2}$$

where $\mathbf{a}_i, \mathbf{x}$ and s_i are basis functions , observed vector and independent components respectively. This ICA model is a generative model, i.e. it describes how the observed data are generated by a process of mixing the components s_i. After estimating the matrix $\mathbf{A}$, we can compute its inverse, say $\mathbf{W}$, and obtain the independent components simply by:

$$\mathbf{s} = \mathbf{W}\mathbf{x} \quad , \quad \mathbf{W} = \mathbf{A}^{-1} \tag{3}$$

In mixing model, $\mathbf{A}$ and $\mathbf{s}$ are both unknown and should be solely estimated using observed data $\mathbf{x}$. One estimation method is to use probabilistic characteristics of components that are assumed independent in ICA method. It has been shown that component independency in ICA mixing model is

directly related to maximum nongaussianity of the components. One robust criterion for nongaussianity testing is negetropy that can be expressed as:

$$J(y) = H(y_{gauss}) - H(y) \tag{4}$$

In which *H* is entropy function and *J(y)* represents the entropy difference between random value *y* and Gaussian variable y_{gauss}, which has the same covariance matrix as *y*. *J(y)* will be zero if and only if *y* is Gaussian, otherwise it has positive non-zero value. Using a robust estimation of negentropy and ascent gradient method, we can rewrite **W** matrix rows as following:

$$\mathbf{w}^{+} = E\{\mathbf{x}g(\mathbf{w}^{T}\mathbf{x})\} - E\{g'(\mathbf{w}^{T}\mathbf{x})\}\ \mathbf{w}^{-} \tag{5}$$

where **x** is the observed vector and $\mathbf{w}^{-}$ and $\mathbf{w}^{+}$ represent the **W** matrix row before update and after update respectively. Here *g(u)=tanh(au)* and *a* is selected so that *1<a<2*. In order to increase the convergence speed, data is whitened in the first step. Hence, the searching space is limited to the space perpendicular to the previous vectors (rows).

2.1.1. Parametric estimation of the probabilistic density function of the independent components

Probability density of independent components is defined as a generalized exponential function as follows [7]:

$$p(x|\mu,\sigma,\beta) = \frac{\omega(\beta)}{\sigma} exp\left[-c(\beta)\left|\frac{x-\mu}{\sigma}\right|^{2/1+\beta}\right]$$

$$c(\beta) = \left[\frac{\Gamma\left[\frac{3}{2}(1+\beta)\right]}{\Gamma\left[\frac{1}{2}(1+\beta)\right]}\right]^{1/1+\beta}, \tag{6}$$

$$\omega(\beta) = \frac{\Gamma\left[\frac{3}{2}(1+\beta)\right]^{1/2}}{(1+\beta)\Gamma\left[\frac{1}{2}(1+\beta)\right]^{3/2}} \quad \sigma>0$$

where μ is the mean parameter, σ represents standard deviation and Γ is Gamma function. β_i is the parameter that controls the deviation of probability density function from normal distribution. The higher β gets ($\beta >> 1$), the more the super-Gaussian of the independent components PDF is. Assuming s_i components with zero mean $(\mu = 0)$ and unit variance $(\sigma^2 = 1)$, we can estimate β by Maximum A Posterior (MAP) method as follows:

$$\beta_i = max_{\beta_i}\ p(\beta_i|s_i) \propto max_{\beta_i}\ p(s_i|\beta_i)p(\beta_i) \tag{7}$$

Assuming $p(\beta_i)$ is uniform for $\beta_i > 1$ and making the derivative with respect to β_i equal to zero, we can estimate β_i by means of iteration methods.

2.2. Shrinkage function extraction

SCS uses Maximum A Posterior for estimating the non-Gaussian variable contaminated with Gaussian noise.
Suppose *s* is the random non-Gaussian variable and *v* is the Gaussian noise with zero mean and unit variance and *y* is the observed variable, so that:

$$y = s + v \tag{8}$$

We want to extract *s* from the observed vector *y*. If the probability density function of *s* is shown by *p*, then we can estimate *s* by:

$$\hat{s} = argmin_u\left[\frac{1}{2\sigma^2}(y-u)^2 + f(u)\right] \tag{9}$$

where $f = -logp$ and σ^2 is the noise variance. Minimizing the left side of (9) and making its derivative with respect to *u* equal to zero is equivalent to the following equation:

$$\frac{1}{2\sigma^2}(\hat{s} - y) + f'(\hat{s}) = 0 \tag{10}$$

For the Laplace function in the form of $p(\hat{s}) = \frac{\lambda}{2} exp(-\lambda|\hat{s}|)$, we have $f'(\hat{s}) = sign(\hat{s})$. After substitution of $f'(s)$ in (10), we find $\hat{s} = |y| - \lambda\sigma^2$ for $|y| > \lambda\sigma^2$ and the equation has no answer for $|y| < \lambda\sigma^2$. Assuming that the clean signal components are sparse, we can conclude that, for $|y| < \lambda\sigma^2$ the observed component y is only resulted from noise and therefore $\hat{s} = 0$. In general, nonlinear relationship between $\hat{s}$ and *y* components can be stated:

$$\hat{s} = g(y) = sign(y) \times max(0, |y| - \lambda\sigma^2) \tag{11}$$

where *g* represents the shrinkage function. In ICA space, the probability density of the clean speech, *p(s)*, can be obtained as explained in section 2-1-1. Assuming generalized exponential distribution for *s*, it is unlikely to find $\hat{s}$ with respect to *y* from (10) in the closed form. The shrinkage function of a high sparse variable ($\beta >> 1$) can be approximated as [1]:

$$\hat{s} = sign(y) \times max\left(0, \frac{|y| - bd}{2} + \sqrt{\left(\frac{|y| + bd}{2}\right)^2 - 4\sigma^2(a+3)}\right) \tag{12}$$

where *d* is the standard deviation of the independent components and σ^2 is the noise components variance in ICA space and $a = 1/d/(E^2\{|s|\} - 1)$, $b = \sqrt{a(a+1)/2}$. Two examples of shrinkage function are illustrated in Fig.1. Since independent components extracted from speech have high sparsity, we can use the shrinkage function stated in (12).

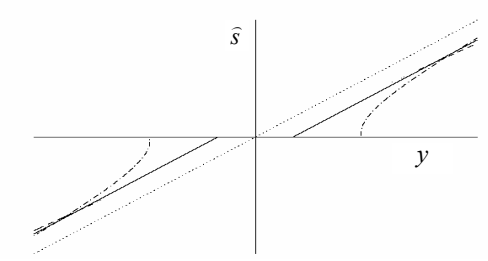


Figure 1: Two example of shrinkage function. Solid line: the shrinkage function of Laplace distribution (11), Dotted line: the shrinkage function of supergaussian distribution (12).

2.3. The Proposed Method

In previous works, SCS is used alone for speech enhancement. It seems that shrinkage threshold reduction can reduce the approximations used in solving (10) and therefore may result in improvement in denoising process. Shrinkage threshold is directly related to the noise energy. This relationship is apparent in the shrinkage function of Laplace distribution, i.e. (11). Shrinkage threshold reduction can be carried out by applying an appropriate preprocessing. One preprocessing method that can keep the Guassianity of the noise is a linear transform and optimal linear transform that causes maximum noise reduction can be a Wiener filter.
Wiener filter applied in ICA space can be easily found [8]:

$$\mathbf{F} = \mathbf{I}(\mathbf{I} - \mathbf{W}\mathbf{R}_n\mathbf{W}^T)^{-1} \tag{13}$$

where $\boldsymbol{I}$ is the unit matrix and is equivalent to the covariance matrix of independent components. Note that independent components attained from ICA model have zero mean and unit variance. Besides, $\mathbf{W}$ is the unmixing matrix of ICA model and $\mathbf{R}_n$ is the noise covariance matrix in the time domain. Additive noise $\mathbf{v}$ can be found by applying $\mathbf{F}$ function on ICA space components.

$$\begin{aligned} &\mathbf{F}(\hat{\mathbf{s}} + \mathbf{v}) - \hat{\mathbf{s}} = \hat{\mathbf{v}} \;\rightarrow\; (\mathbf{F} - \mathbf{I})\hat{\mathbf{s}} + \mathbf{F}\mathbf{v} = \hat{\mathbf{v}} \\ &\mathbf{R}_{\hat{v}} = (\mathbf{F} - \mathbf{I})(\mathbf{F} - \mathbf{I})^T + \mathbf{F}\mathbf{R}_v\mathbf{F}^T \end{aligned} \tag{14}$$

$\mathbf{v}$ has Gaussian distribution and according to the central limit theorem, and employing only linear function, $\hat{\mathbf{v}}$ can also be assumed Gaussian. The σ in (12) can be substituted with diagonal components of the covariance matrix $\mathbf{R}_{\hat{\mathbf{v}}}$ (noise variance) and the resultant shrinkage function can be applied to each noisy component.

3. Experimental Results

In the experimental setup, we first train the ICA model with 22 files extracted from TIDIGIT database. For this purpose, we divide the speech files to the frames of 40 samples with one sample frame shift (sampling frequency is 8 kHz).
After applying hamming window on the frames and eliminating the mean value, ICA training algorithm using negentropy criteria was performed. $\mathbf{W}$ matrix has been estimated in the training process and independent components were calculated by $\mathbf{s} = \mathbf{W}\mathbf{x}$. β corresponds to the probability density function of independent components and a, d and b pertaining to the shrinkage function have been also estimated. The estimated β was in the range of 7-12 that verified the supergaussianity of the independent components. Finally, we found 40 shrinkage functions for 40 independent components.
In order to test the algorithm, 30 files have been selected randomly and specific noises were added manually from NATO RSG-10 noise data. We used two parameters for evaluating the enhancement capability of our approach which are global SNR and segmental SNR. Table 1 and Table 2 include the obtained values of global and segmental SNRs respectively after applying Wiener filter, Shrinkage function and jointly using them aimed to improving the speech signal quality. Segmental SNR results represent the difference between segmental SNR of the input noisy speech and enhanced speech signal. The results show that our proposed method (with the presumption of having only Gaussian noise) not only improved the SNR of the speech signal in the presence of white noise but also could be efficient for the speech signal contaminated with the colored noise. Except for the Volvo noise (where the Wiener filter has reduced the noise level considerably) it can be seen that our proposed method has made a significant improvement in all SNRs.

4. Conclusion

In this paper we proposed a new method by jointly using Sparse Coding Shrinkage and Wiener filtering. In general, each speech enhancement algorithm independently reduces the effect of a specific noise. Thus, combinational algorithms can be more efficient in various noise environments. For example, in the presence of Volvo noise, Table 1 shows that the performance of Wiener filter in reducing the noise level is considerable, in comparison with the SCS effect. Therefore, jointly use of these methods may lead to better results in the presence of various types of noises. On the other hand, Wiener filter improves the SCS method performance by reducing the shrinkage threshold of the shrinkage function. The performance of the proposed method should be further studied before its application in speech recognition systems.

5. Acknowledgements

This work was in part supported by Iran Telecommunications Research Center (ITRC).

6. References

[1] I. Potamitis, N. Fakotakis, G. Kokkinakis, "*Speech* enhancement *using the sparse code shrinkage technique*, " in *Proc. ICASSP*, vol.1, pp.621-624, May 2001.

[2] A. Hyvarinen, P. Hoyer, E. Oja, "*Sparse code shrinkage for image denoising*, " in *Proc IJCNN*, vol.2, pp.859-864, May 1998.

[3] A. Hyvarinen, P. *Hoyer*, E. Oja, "*Sparse Code* Shrinkage*: Denoising of Nongaussian Data by Maximum Likelihood Estimation,* " Neural Computation, Vol. 11, No. 7, pp. 1739-1768, Oct 1999.

[4] J. Liu, C. Zhao, X. Zou, Wei Zhang, "*An Approach of Speech Enhancement by Sparse Code Shrinkage*," in *Proc. ICNN & B*, vol.3, pp.1952-1956, Oct 2005.

Table 1: Global SNRs of enhanced signals in different enhancement algorithms.

noise \ SNR(dB)		20	15	10	5	0	-5	Ave.
Babble	SCS	20.92	16.39	11.98	7.74	3.73	-.26	10.08
	Wiener	20.2	15.47	10.9	6.7	3.17	.78	9.54
	SCS+Wiener	**20.9**	**16.4**	**12**	**7.8**	**4.1**	**1.59**	**10.47**
Factory	SCS	21	16.59	12.24	7.95	3.8	-.3	10.21
	Wiener	20.68	16.2	11.8	7.6	4	1.3	10.26
	SCS+Wiener	**21.2**	**17.1**	**13.1**	**8.9**	**4.9**	**2**	**11.20**
F16	SCS	21.22	16.79	12.42	8.11	3.99	-.8	10.29
	Wiener	20.2	16.5	12.2	8.2	4.53	1.7	10.56
	SCS+Wiener	**21.5**	**17.5**	**13.6**	**9.6**	**5.6**	**2.49**	**11.72**
White	SCS	22.37	18	13.55	9	4.65	.2	11.30
	Wiener	20.7	16.14	12	8.25	5	2.38	10.75
	SCS+Wiener	**22.4**	**18**	**13.8**	**9.9**	**6.2**	**3.2**	**12.25**
Volvo	SCS	21.28	17.22	12.95	8.68	4.46	.16	10.79
	Wiener	25.8	24.7	23.4	20.95	17.8	16.4	21.51
	SCS+Wiener	**24**	**22.9**	**21.4**	**18.8**	**15.8**	**13**	**19.42**
Destroyer Engine	SCS	22.51	18.24	13.8	9.3	4.86	.4	11.52
	Wiener	20.32	15.9	11.94	8.36	5.03	2.2	10.63
	SCS+Wiener	**22**	**18**	**13.9**	**10**	**6.3**	**2.2**	**12.07**
Ave.	SCS	21.55	17.21	12.82	8.46	4.25	-0.10	10.70
	Wiener	21.32	17.49	13.71	10.01	6.59	4.13	12.21
	SCS+Wiener	**22.00**	**18.32**	**14.63**	**9.24**	**7.15**	**4.08**	**12.85**

Table 2: Segmental SNR improvemnet for different enhancement algorithms (the results are differential SNR between input signal and enhanced signal).

noise \ SNR(dB)		20	15	10	5	0	-5	Ave.
Babble	SCS	1.19	2.10	3.5	4.8	5.6	5.9	3.85
	Wiener	.17	.46	.99	1.93	3.77	7.1	2.40
	SCS+Wiener	**1.18**	**2.32**	**4.46**	**7.46**	**10.63**	**14.43**	**6.75**
Factory	SCS	1.18	2.14	3.32	4.37	5.1	5.6	3.62
	Wiener	.56	1	1.8	2.8	4.56	7.6	3.05
	SCS+Wiener	**1.42**	**2.9**	**5.2**	**8.1**	**11**	**14.9**	**7.25**
F16	SCS	1.4	2.27	3.33	4.33	5.1	5.6	3.67
	Wiener	.7	1.34	2.19	3.31	5	7.9	3.41
	SCS+Wiener	**1.77**	**3.3**	**5.6**	**8.4**	**11.4**	**15**	**7.58**
White	SCS	2.54	3.33	4.01	4.58	5.06	5.45	4.16
	Wiener	.8	1.4	2.43	3.97	6.12	9	3.95
	SCS+Wiener	**3.12**	**4.47**	**6.25**	**8.65**	**11.79**	**15.8**	**8.35**
Volvo	SCS	.16	1.13	1.82	2.61	3.6	4.56	2.31
	Wiener	5	8.2	11	13	15	17	11.53
	SCS+Wiener	**3.5**	**7**	**10.3**	**13.3**	**16.5**	**19.8**	**11.73**
Destroyer Engine	SCS	2.97	3.9	4.56	5.05	5.43	5.74	4.61
	Wiener	.4	1.13	2.34	4	6	8.8	3.78
	SCS+Wiener	**3.12**	**4.5**	**6.3**	**8.9**	**12**	**15.8**	**8.44**
Ave.	SCS	1.57	2.48	3.42	4.29	4.98	5.48	3.70
	Wiener	1.27	2.26	3.46	4.84	6.74	9.57	4.69
	SCS+Wiener	**2.35**	**4.08**	**6.35**	**9.14**	**12.22**	**15.96**	**8.35**

[5] P. Vary, *"Noise suppression by spectral magnitude estimation," Signal Processing*, Vol. 8, pp. 387-00, 1985.

[6] A. Hyvarinen, J. Karhunen, E. Oja, *Independent Component Analysis*, John Wiley & Sons, New Yourk etc. 2001.

[7] T.-W. Lee, M.-S. Lewicki″ *The Generalized Gaussian Mixture Model Using ICA,"* in *Proc. Int Workshop.* on *Independent Component Analysis*, pp.239-244, 2000.

[8] T.-W. Lee, M.-S. Lewicki″ *The Generalized Gaussian Mixture Model Using ICA,"* in *Proc. Int Workshop.* on *Independent Component Analysis*, pp.239-244, 2000.

Efficient Viterbi algorithms for lexical tree based models

S. España-Boquera, M.J. Castro-Bleda, F. Zamora-Martínez, J. Gorbe-Moya

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
sespana@dsic.upv.es

Abstract

In this paper we propose a family of Viterbi algorithms specialized for lexical tree based FSA and HMM acoustic models. Two algorithms to decode a tree lexicon with left-to-right models with or without skips and other algorithm which takes a directed acyclic graph as input and performs error correcting decoding are presented. They store the set of active states topologically sorted in contiguous memory queues. The number of basic operations needed to update each hypothesis is reduced and also more locality in memory is obtained reducing the expected number of cache misses and achieving a speed-up over other implementations.

1. Introduction

Most of large vocabulary Viterbi based recognizers (for speech, handwritten or other recognition tasks, although speech terminology is used in this work with no loss of generality) make use of a lexicon tree organization which has many advantages over a linear lexicon representation [1, 2]. As it is shown in the literature, more compact representations are possible (using a *lexicon network* [3], which is a minimized Finite State Automaton –FSA–) but the gain in space is accompanied with a more complex Viterbi decoder. Therefore, lexical tree organization is a very good tradeoff between compact space representation and adequacy for decoding.

The search space in a recognizer can be huge and the key to achieve practical performance is to consider only the set of active hypothesis (those with non trivial zero probability) and to apply pruning techniques such as beam search which only maintain active the best hypothesis.

Large vocabulary one-step decoders [4] usually keep a set of lexical tree based Viterbi parsers in parallel. Two common approaches are the *time-start copies* and *language model history copies* [5, 6]. In the time-start approach, all hypothesis competing in a tree parsing share the same word start time. When a trigram language model is used, the language model history copies approach maintains a tree parsing for every bigram history $(w1, w2)$. This second approach has a loss of optimality which is known as *word-pair approximation* [6]. In both cases, it is straightforward to use a specialized Viterbi algorithm for the lexical tree model and, as the core of an automatic speech recognizer lies in the search process, every little improvement in performing specialized decoding of lexical tree models has a great impact in the overall performance. Therefore, it is not strange to find specialized algorithms which take advantage of the properties of tree based HMM models which integrate the tree lexicon and the acoustic HMM models.

This work has been partially supported by the Spanish Government (TIN2006-12767) and by the Generalitat Valenciana (GVA06/302).

When the acoustic models are strict left-to-right without skips, the resulting expanded HMM model is acyclic if loops are ignored, and every node has only zero or one preceding state to take into account in the dynamic programming equation. Left-to-right units with skips are known as Bakis topology and have a widespread use as acoustic models in most recognizers. When those models are used in conjunction with a tree lexicon, the number of predecessors given an active state can be zero, one or two.

Not expanding the acoustic models in the tree lexicon and maintaining a pure tree structure which matches a phone-graph is another possibility. In this case, the input is no longer a sequence of acoustic frames but a phone-graph (a directed acyclic graph –DAG– labelled with phones and acoustic scores). Besides the capability of using a directed acyclic input, the possibility of insertions, deletions and substitutions of phones is needed to tolerate the errors in the phone-graph generation.

In this work, three specialized Viterbi algorithms based on contiguous memory queues (FIFO data structures) are proposed. When performing a Viterbi step, a new result queue is created with the help of one or several auxiliary queues.

The basic algorithm uses left-to-right HMM acoustic models with no skips. This algorithm can be applied whenever acoustic left-to-right models without skips are used: it can be used for isolated word or continuous speech recognition, either with a one-step or a two-step approach, with time-start or language model history copies, and also within-word or across-word context dependent units (triphones, quinphones, etc.). A simple extension is presented to show how to use it with across-word context dependent models [7].

A second version of the algorithm extends the first one to allow the use of skips in the acoustic units with a negligible additional cost.

The last proposed algorithm performs an error correcting Viterbi decoding and is capable of analyzing a DAG instead of a sequence. This algorithm can be used, for instance, to obtain a word-graph from a phone-graph.

2. Left-to-right without skips algorithm

If a lexicon tree is expanded with left-to-right acoustic HMM models without skips, the following observations about the expanded tree models are straightforward:

- Every state has at most two predecessors: itself and possibly his parent.
- If we ignore the loops, the expanded model is acyclic. Therefore, a topological order is possible in general.
- A level traversal of the tree provides a topological order with some additional features:

– The children of a given node occupy contiguous positions. The grandchildren also occupy contiguous positions.

– If a subset of states is stored in topological order and we generate the children of every active state following that order, the resulting list also is ordered with respect to the topological order.

2.1. Model representation

A tree model T of n states is represented with three vectors of size n and one of size $n+1$ as follows:

- *loop_prob* stores the loop transition probabilities.
- *from_prob* stores the parent incoming transition probabilities.
- e_index stores the index of the associated emission probability class associated to the acoustic frame to be observed. The vector of emission probabilities can be obtained with a multilayer perceptron in a hybrid model [8] or with a set of mixture of Gaussian distributions in a conventional continuous density HMM.
- *first_child* stores the index of the first child. The last child is deduced by looking the first child of the next state thanks to the topological sorting. This representation allows specifying an empty set of children. A sentinel in position $n+1$ is needed for the last state.

2.2. Viterbi-Merge algorithm

The Viterbi-M algorithm takes a sequence of acoustic frames as input and updates a set of active states after observing every frame (a Viterbi step). An active state is composed by an index state and a score (i, s). A queue $\alpha(t)$ (a FIFO data structure) is used to store the set of active states at time t. The purpose of a Viterbi step consists of creating another queue $\alpha(t+1)$ given the model T, the current queue $\alpha(t)$ and the vector of emission probabilities $emission$ associated to the observed acoustic frame.

An auxiliary queue aux_child is used to store temporally the scores of states produced by the transitions from parent to child. The algorithm proceeds as follows (see Figure 1):

1. $best\_prob \leftarrow 0$.
2. For every active state (i, s) of the queue $\alpha(t)$ whose score s is above the beam threshold:
   (a) $s\_next \leftarrow s \cdot loop\_prob[i]$.
   (b) While the first active state (i', s') of the queue *aux_child* satisfies $i' < i$, extract it and place $(i', s' \cdot emission[e\_index[i']])$ in $\alpha(t+1)$.
   (c) If the first active state (i', s') of the queue *aux_child* satisfies $i' = i$, drop it and update the score $s\_next \leftarrow \max(s\_next, s')$.
   (d) $s\_next \leftarrow s\_next \cdot emission[e\_index[i]]$, insert (i, s_next) in the queue $\alpha(t+1)$ and update $best\_prob \leftarrow \max(best\_prob, s\_next)$.
   (e) For every state j from *first_child*$[i]$ to *first_child*$[i+1]-1$, add $(j, s \cdot from\_prob[j])$ to the queue *aux_child*.
3. For every active state (i', s') of the queue *aux_child*, extract it and place $(i', s' \cdot emission[e\_index[i']])$ in the queue $\alpha(t+1)$.

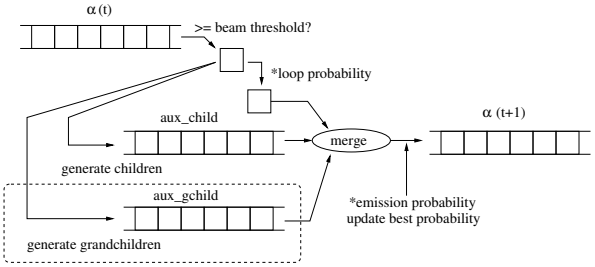


Figure 1: *Viterbi-M and Viterbi-MS algorithms. The queue* aux_gchild *(dotted part) is only used in Viterbi-MS.*

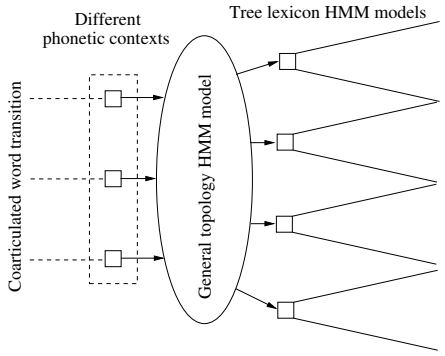


Figure 2: *HMM model for across-word context dependent units. The general topology HMM model represents only the first context dependent phones of the words.*

The active states of $\alpha(t)$ whose score is below the beam threshold are discarded. Whenever an active state is placed in the queue $\alpha(t+1)$, the emission probability associated to it is applied, and the best probability is updated. This value is used to obtain the beam threshold for the following Viterbi step.

This algorithm is linear with the number of active states and the number of children of active states which is an upper bound to the number of active states in the resulting queue. In total, the global cost is linear with the number of active states, which is the same as any reasonable implementation of a conventional Viterbi implementation. The main advantage of this algorithm is the use of contiguous memory FIFO queues to store the active states. Therefore, a better cache performance and an internal loop with less overhead is obtained compared to other algorithms that use linked lists or use hash tables to store and look up the set of active states. Therefore, the asymptotic cost is the same but a practical speed-up is obtained.

2.3. Extension to across-word context dependent units

Since this algorithm is used on lexical tree models with expanded acoustic HMM models, the use of context dependent units is straightforward for within-word context modeling.

Across-word models consider a different context dependent unit at the beginning of a word to take into account the last phones of the preceding word during continuous speech recognition. It would be very inefficient to use a different tree model for every possible context since they only differ in the first context dependent acoustic models. Therefore, a model which resembles a tree lexicon excepting the root is used. This model can be composed of two models: a general HMM connecting a set of tree lexicon models (see Figure 2).

A set of trees can be traversed by levels as if they were just one tree and the resulting model can be used with the same algorithm with no modification. Therefore, a conventional Viterbi algorithm can be used to update the scores of the states of the general topology HMM part of the model, and the rest of the model (a forest) can be computed with the Viterbi-M algorithm.

3. Left-to-right with skips algorithm

This algorithm generalizes the previous one by allowing the use of Bakis HMM acoustic models.

3.1. Model representation

The model representation is similar to the previous section. The only difference is another vector *skip_prob* which stores, for every state, the incoming skip transition probabilities. In order to iterate over the set of grandchildren of a given state i, the algorithm loops from $first\_child[first\_child[i]]$ to $first\_child[first\_child[i+1]]-1$.

3.2. Viterbi-Merge algorithm with skips

The Viterbi-MS algorithm is the same of previous section but another auxiliary queue *aux_gchild* is used to store the active states with scores computed by means of the skip transitions. Every time an active state is extracted from $\alpha(t)$, the set of grandchildren is used to add items to the queue *aux_gchild* just as the set of children is used to add items to the other auxiliary queue. Now, the resulting queue $\alpha(t+1)$ is obtained by merging the loop transition score of the processed active state with the states from the two auxiliary queues (see Figure 1).

This algorithm is linear with the number of active states and the number of children and grandchildren of active states. The resulting cost is thus linear with the number of active states.

3.3. Extension to across-word context dependent units

The same observations of previous algorithm are also applicable here.

4. Error-Correcting Viterbi for DAGs

The last proposed algorithm performs an error correcting Viterbi decoding and is capable of analyzing a DAG instead of a sequence.

4.1. Model representation

A tree model T of n states where symbols are placed at the transitions is represented with two vectors of size n and other of size $n+1$ as follows:

- $symbol$ stores the incoming transition label.
- $from\_prob$ stores the incoming transition probability.
- $first\_child$ stores the index of the first child as in the previous algorithms.

A table with the costs of insertions, deletions and substitution of every symbol is also required.

4.2. Error-Correcting Viterbi-Merge algorithm for DAGs

The Viterbi-MEC-DAG algorithm takes a DAG as input. Consider the phone-graph of Figure 3. A set of active states is associated to every vertex of the input DAG. The algorithm applies two different procedures associated to the input DAG,

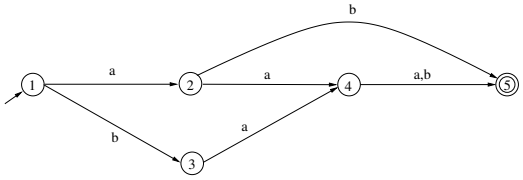


Figure 3: *Phone-graph example.*

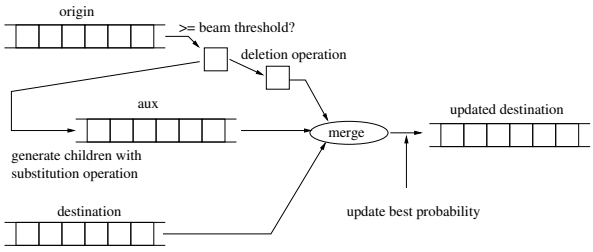


Figure 4: *Viterbi-MEC-DAG edge-step procedure.*

which must be applied following the DAG topological order: the vertex-step procedure must be applied to every vertex before using this set of active states as the origin of an edge edge-step procedure.

4.2.1. Edge-step

For every edge, a Viterbi step takes the set of active states of the origin vertex and use them to update the active states of the destination vertex. This procedure only considers the cost of deletions and substitutions. As can be observed in Figure 4, this algorithm is similar to the Viterbi-M algorithm where loop probability updating is replaced by the deletion operation, the generation of children states corresponds to the substitution operation (including a symbol by itself or a correct transition). Another difference, which can be also used in Viterbi-M and Viterbi-MS to process a DAG as input data, is the presence of second input queue which stores the active states already updated at the destination vertex by means of other edges of the DAG. These values are simply merged and this queue is not needed when the input data is a sequence. The cost of this procedure is linear with the number of active states in both input queues because the number of generated successor states grows linearly with the number of active states.

4.2.2. Vertex-step

Once all edges arriving at a given vertex have been processed, the insertion operation is considered. This operation updates a set of active states without consuming any symbol. As can be observed in Figure 5, the output of the auxiliary queue is used to insert more active states in the same queue to take into account the possibility of several insertion operations. The cost is not linear with the number of active states: a sole active state at the root could, in principle, activate all the states of the model, but most of them are expected to be pruned by the beam search depending on the cost of insertions and the beam width. The cost of this operation is linear with the number of active states before applying the procedure plus the number of active states after the procedure, which is bounded by the number of states in the model.

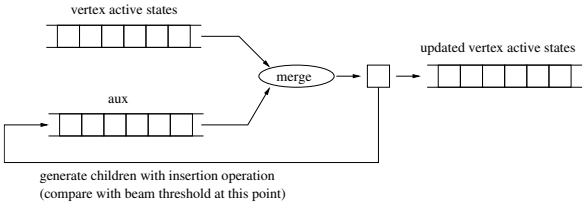


Figure 5: *Viterbi-MEC-DAG vertex-step procedure.*

Num. states	Hash swap	A. Envelope	Viterbi-M
9 571	3.001	16.082	29.350
76 189	2.761	12.924	28.036
310 888	1.922	6.442	24.534

Table 1: *Experimental results. The results are shown in millions of active states or hypothesis updated per second.*

5. Experimental results

A previous work related to lexical tree Viterbi decoding we were aware of after our algorithms were developed is the *active envelope* algorithm [9]. This algorithm also uses a total order which subsumes the partial order of the states of the model and places siblings contiguously. This algorithm is specified for left-to-right models without skips, so it is only comparable to our first algorithm Viterbi-M. The active envelope algorithm uses a single linked list to perform a Viterbi step, which is an advantage in memory usage. Since this algorithm modifies the original set of active states, it is restricted to sequential input data and cannot be used when the input data is a DAG. In order to use only a list, the active hypothesis in active envelope algorithm are traversed in *reverse* topological order. The "price to pay" for this advantage is the need of linked lists instead of contiguous memory arrays. Since the use of linked lists cannot assure memory locality and the cost of traversing them is greater than traversing memory arrays, it is expected to perform worse than Viterbi-M algorithm. The memory occupied by an active hypothesis is an index state and a score; if linked lists are used, a pointer is also needed: so a linked list needs approximately 50% or 100% more memory per active state depending on the computer architecture. On the other hand, the use of memory arrays needs an estimation of the number of active states.

In order to compare the performance of our Viterbi-M algorithm, two more algorithms have been implemented: a conventional Viterbi algorithm based on hash tables with chaining to store and to look up the active states and the active envelope algorithm. All algorithms have been implemented in C++ and use the same data structures to represent the tree based HMM models as described in section 2.1.

The experiments were done on a Pentium D machine at 3GHz with 2 Gbytes of RAM using a Linux with kernel `2.6.18` and the `gcc` compiler version `4.1.2` with `-O3` optimization. The lexical trees used in the experiments were obtained by expanding 3-state left-to-right without skips hybrid neural/HMM acoustic models in the tree lexicon. The size of these trees varies from 9 571 to 310 888 states. Only the Viterbi decoding time has been measured (the emission scores calculation and other preprocessing steps were not taken into account). The result is shown in Table 1. The speed is measured in millions of active states updated per second.

6. Conclusions

In this paper, three Viterbi algorithms specialized for lexical tree based FSA and HMM acoustic models have been described. Two of these algorithms are useful to decode a set of words given a sequence of acoustic frames and the third one is useful to parse a phone-graph with error-correcting edition operations. These algorithms are based on contiguous memory queues which contain the set of active states topologically sorted.

Although the asymptotic cost of these algorithms is the same as any reasonable implementation of the Viterbi algorithm, the experimental comparison between the Viterbi-M algorithm, a conventional Hash-table swapping algorithm and the active envelope algorithm, shows that our algorithm is approximately 10 times faster than the hash-table swapping implementation and from 2 to 4 times faster than the active envelope algorithm. A decrease in speed with the size of the models is observed in the three algorithms, which is possibly related with the main memory and the cache relative speeds. For this reason, more experimentation is needed in order to better understand this behaviour and also to study the effect of other parameters such as the beam width of the pruning during the search.

7. References

[1] J. Klovstad and L. Mondshein, "The CASPERS linguistic analysis system," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 23, no. 1, pp. 118–123, Feb. 1975.

[2] D. Klatt, *Trends in Speech Recognition*. Prentice-Hall, 1980, ch. Scriber and Lafs: Two New Approaches to Speech Analysis, pp. 529–525.

[3] K. Demuynck, J. Duchateau, and D. V. Compernolle, "A Static Lexicon Network Representation for Cross-word Context Dependent Phones," in *Proc. European Conference on Speech Communication and Technology*, vol. I, Rhodes, Greece, September 1997, pp. 143–146.

[4] X. L. Aubert, "An overview of decoding techniques for large vocabulary continuous speech recognition," in *Computer Speech and Language*, vol. 16, 2002, pp. 89–114.

[5] S. Ortmanns, H. Ney, F. Seide, and I. Lindam, "A comparison of time conditioned and word conditioned search techniques for large vocabulary speech recognition," in *Proc. ICSLP '96*, vol. 4, Philadelphia, PA, 1996, pp. 2091–2094.

[6] H. Ney and S. Ortmanns, "Dynamic programming search for continuous speech recognition," *IEEE Signal Processing Magazine*, vol. 16, no. 5, pp. 64–83, 1999.

[7] S. Kanthak, A. Sixtus, S. Molau, and H. Ney, "Within-word vs. across-word decoding for online speech recognition," 2000.

[8] Y. Konig, H. Bourlard, and N. Morgan, "REMAP: Recursive estimation and maximization of A posteriori probabilities — application to transition-based connectionist speech recognition," in *Advances in Neural Information Processing Systems*, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds., vol. 8. The MIT Press, 1996, pp. 388–394.

[9] P. Nguyen, L. Rigazio, and J.-C. Junqua, "EWAVES: An efficient decoding algorithm for lexical tree based speech recognition," in *ICSLP-2000*, vol. 4, 2000, pp. 286–289.

Acoustic Units Selection in Chinese-English Bilingual Speech Recognition

Lin Yang. Jianping Zhang. Yonghong Yan

ThinkIT Speech Lab
Chinese Academy of Sciences, Beijing, China
yanglin@h

Abstract

We present an effective method to merge the acoustic units between Chinese and English to develop a language-independent speech recognition system. Chinese as a tonal language has large differences from English. An optimal Chinese phoneme inventory is set up in order to keep consistent with the representation of English acoustic units. Two different approaches for Chinese-English bilingual phoneme modeling are illustrated and compared. One is to combine the Chinese and English phonemes together based on International Phonetic Association (IPA). The other is a data-driven method on the basis of the confusion matrix. Experimental results show that all these methods are feasible and the data-driven method reduced the WER by 0.73% in Chinese and 3.76% in English relatively compared to the IPA-based method. As a by-product, the idea of data sharing across languages can obtain relative 8.7% error reduction under noise condition.

1. Introduction

With the increasing internationalization, research in multilingual speech recognition (MSR) has gained more and more interest in the last few years. But Chinese as one of the most important languages in the world was not considered as much as western languages in some present MSR systems [1][2][3] because of its own peculiarity. So a thorough research on the Chinese-English bilingual recognition is potentially needed, especially the groundwork of a MSR system: the selection of language-independent acoustic units.

So far, there are two main frameworks when solving the MSR problem. One is using a language identification (LID) module to identify the speech as a specific language. Then the specific monolingual utterance can be recognized by language-dependant speech recognizers. The other is using a universal framework including multi-lingual acoustic model, language model and decoder. In the first framework an apparent weak is that the upper bound of performance is limited by the accuracy of LID module. Thus we use the second framework to study how to construct a consistent Chinese-English phoneme inventory and acoustic model.

Linguistically speaking, there are many differences between Chinese and English since they come from two different language families [4]. Firstly, Chinese is a kind of tonal language, including 5 tones; Secondly it is monosyllabic mainly in CV structure where C is consonant and V is vowel, whereas English is a kind of atonal language whose structure is more complex than Chinese. As a result in most cases the basic acoustic units are different largely between the two languages. For example, the Chinese mono phoneme is often represented by initials and tonal finals rather than more subtle units as English. So it is essential to make the acoustic units of the two languages uniform. This paper gives some experiments about how to split the phonemes of Chinese to make it keep consistent with English representations.

Some phonemes across the two languages may be similar enough to be equated. Those resemblances must be merged together for decreasing the number of parameters in the acoustic model. At present there are two main methods of phoneme cluster. One is based on phonetic knowledge; the other is data-driven. Both of the methods are studied to find the most suitable universal phoneme inventory and build a language-independent acoustic model which keeps balance between the number of parameters and system performance in a real MSR framework.

The remainder of this paper is organized as follows. In section 2, an approach of Chinese phone splitting to subtle units and two methods of bilingual phoneme clustering are illustrated in detail. Some experimental results on language independent speech recognizer with different cluster technologies are compared in section 3. Conclusion is given in section 4.

2. Building Language-Independent Acoustic Model and MSR System

In order to define a universal phoneme sets for the two languages, we firstly split original Chinese initials and tonal finals into subtle units consistent with the representation of English. If the phonemes of the two languages are put together directly, the final number of acoustical model parameters would be much large and be a burden for decoder. The cluster of phonemes can be performed either manually or automatically. A kind of most common used method of data-driven cluster is based on the direct distance of mono phoneme models. However this method does not consider the context of the phonemes. In order to utilize the information of triphones a new data-driven method based on the concept of confusion matrix is given.

2.1. Setup of Chinese phoneme inventory

In our original system the acoustic units are initials and tonal finals, a total number of 213, which is a characteristic of Chinese. In order to be compatible with English atonal phonemes, we discard the tones at first, as a result with the total number of 69. There still exist some compound vowels in the atonal Chinese phoneme inventory although they have resembled to an extent with English units. So splitting according to IPA [5] at various levels is attempted with a number of comparative experiments. Detailed results are shown in table 1. Finally the best Chinese inventory is given with a total number of 49 (including silence and short pause),

balanced in size with English phoneme set with the number of 42 (including silence and short pause).

Table 1. Results of recognition in various phoneme sets of Chinese

	213 tonal set	69 atonal set	57 atonal set	49 atonal set	43 atonal set
Accuracy (%)	94.9	94.1	94.2	94.4	94.0

In this group of comparative experiments, the training set includes 70 hours data and the test set is 863test set including 9042 read utterances. From the results we could conclude that the set of 49 phonemes provides the best performance comparative with other atonal sets, although there still exists a disparity from the original tonal phone set.

2.2. Experience-based phoneme cluster

After the subtle split of Chinese phonemes, we get two sets of similar phoneme inventories representing the two languages. Then the language-dependent phonemes should be combined into one set in order to realize the language-independent phoneme modeling. A manual and direct way of building bilingual inventory is according to the phonetic knowledge. Some language-dependent phonemes which are represented by the same IPA symbols can be merged into one unit. Table 2 is a list of IPA-based Chinese-English universal phoneme set consisting of 67 units (excluding silence, short pause and garbage model). In this table 19 pairs of phonemes sharing the same IPA symbols are merged, thus the total number of parameters is reduced by more than 21% comparative to the total number of the two sets, which is 89.

Table 2. Phoneme cluster based on IPA

Lang.	Phonemes	number
Chinese	p_ch t_ch nn_ch k_ch z_ch c_ch sh_ch r_ch zh_ch ch_ch j_ch q_ch x_ch h_ch a_ch au_ch at_ch e_ch err_ch ix_ch iy_ch v_ch iaa_ch ioo_ch iee_ch iii_ch iuu_ch ivv_ch	28
English	b_en ch_en d_en dh_en g_en hh_en jh_en r_en sh_en th_en v_en w_en y_en z_en zh_en ah_en ao_en aw_en ay_en oy_en	20
Merged	b_ch/p_en f_ch/f_en n_ch/n_en g_ch/k_en ng_ch/ng_en d_ch/t_en m_ch/m_en l_ch/l_en s_ch/s_en aa_ch/aa_en ee_ch/eh_en ak_ch/ae_en o_ch/ow_en uu_ch/uh_en u_ch/uw_en ea_ch/ey_en ii_ch/ih_en er_ch/er_en i_ch/iy_en	19

2.3. Data-driven phoneme cluster

The data-driven method of cluster is based on the statistical similarity or distance measurement rather than phonetic knowledge. So far the clustering algorithm is mostly applied Bhattacharyya distance [6] as a theoretical similar measure between two Gaussian distributions. However this approach considers only the mono phoneme's distance without any information of context. A novel method of distance measurement with the information of triphones, which is similar to [7][8], is used in this study. The phonemes' distances are calculated according to confusion matrix between English and Chinese phonemes. The confusion is measured by finding the best path for test data of one language on the other's triphone model. The optimization is realized by simple Viterbi decoding and a large number of training utterances are required.

Detailed steps are as follows:

(1) Based on the English triphone HMM model, find the most possible phoneme series for every Chinese utterance using Viterbi decoding. That is

$$\begin{aligned}\hat{\Lambda}_{en} &= \arg\max_{\Lambda_{en}} P(\Lambda_{en} \mid O_{ch}, M_{trien}) \\ &= \arg\max_{\Lambda_{en}} P(O_{ch} \mid \Lambda_{en}, M_{trien})\end{aligned} \quad (1)$$

where Λ_{en} means the best possible English phoneme series, M_{engtri} means the English triphone model, and O_{ch} represents the Chinese training utterance. Then based on the time information to align Λ_{en} and the real phoneme array Λ_{ch} by DTW.

(2) A confusion matrix is build by a large number of training data. The degree of confusion is calculated by

$$C_{chi \to enj} = N(\lambda_{enj} \mid \lambda_{chi}) / N(\lambda_{chi}) \quad (2)$$

where $C_{chi \to enj}$ means the distance of Chinese phoneme i from English phoneme j, and $N(\lambda_{chi})$ and $N(\lambda_{enj})$ represent the number of Chinese phoneme i and the number of English phoneme j respectively. Thus a matrix denotes the distance of Chinese phonemes from English phonemes is achieved.

(3) Vice verse. Change the places of English and Chinese and repeat the step 1 and 2 to calculate the distance of English phonemes from Chinese phonemes.

(4) Since the measure is asymmetrical, the average distance is given as follows:

$$C = (C_{chi \to enj} + C_{enj \to chi}) / 2 \quad (3)$$

In order to compare with IPA-based method, we limited the size of data-driven cluster to 67 mono phonemes. The English training data come from TIMIT corpora including 6300 utterance, and transcriptions with the time information are available. The Chinese data come from 863test sets, including 9042 sentences with the time information transcriptions. Experimental results show that when the training data exceeds 2000 utterances the distance of phonemes in the confusion matrix is stable. Thus the size of our training data can gain the reliable statistical measurement. Table 3 shows the result of bilingual phoneme inventory using data-driven method.

Table 3. Phoneme cluster based on Confusion Matrix Data-Driven

Lang.	Phonemes	number
Chinese	ng_ch l_ch t_ch nn_ch z_ch c_ch r_ch j_ch q_ch x_ch h_ch aa_ch ee_ch ak_ch o_ch ii_ch er_ch at_ch e_ch err_ch ix_ch iy_ch v_ch iaa_ch ioo_ch iee_ch iuu_ch ivv_ch	28
English	ng_en t_en l_en dh_en hh_en r_en th_en v_en z_en zh_en eh_en ae_en uh_en uw_en ih_en er_en ah_en ao_en aw_en oy_en	20
Merged	a_ch/aa_en au_ch/ay_en b_ch/b_en ch_ch/ch_en d_ch/d_en ea_ch/ey_en f_ch/f_en g_ch/g_en i_ch/iy_en iii_ch/y_en k_ch/k_en m_ch/m_en n_ch/n_en p_ch/p_en s_ch/s_en sh_ch/sh_en u_ch/u_en uu_ch/ow_en zh_ch/jh_en	19

3. Experiments and Discussion

The goal of the experiment is to evaluate the performance of our language-independent LVCSR system and compare the cluster method based on IPA with our proposed approach based on confusion matrix data-driven.

3.1. Corpora and experiment setup

The training data has about 340 hours Chinese speech data (including various dialects) and 160 hours English data (including TIMIT, WSJ). These data are recorded under relatively clean acoustic conditions. The recognizer makes use of continuous density HMM with Gaussian mixture for acoustic model. Each triphone model is a 3-state left-to-right with Gaussian mixture observation densities (typically 32 components). The acoustic feature of speech is MFCC with 39 dimensions.

The size of vocabulary is 43K for Chinese and 50K for English. The bilingual dictionary is composed of the pooled monolingual dictionaries and consists of 93K entries. The training corpora of bilingual language model (BILM) are the combination of Chinese and English data. In order to test the performance of language-independent acoustic model we also build monolingual LM, labeled as MONOLM.

We selected two sets of Chinese testing data in order to evaluate the performance under various situations. One is recorded under clean acoustic condition, including 341 Chinese sentences labeled as TestCH1, while the other is recorded under noise condition consisting of 200 sentences labeled as TestCH2. The English testing data is standard WSJ0 testing set labeled as TestEN, including 330 English utterances.

3.2. Experiments and discussion

For the purpose of comparing the language-independent system with the language-dependent system, we conducted the experiments on the monolingual LVCSR system. Table 4 shows the word error rates of the baseline on different testing sets.

Table 4. The monolingual results on different testing sets

	TestCH1	TestCH2	TestEN
WER (%)	25.8	36.8	9.7

In order to compare the two methods of cluster we trained the acoustic model on the two bilingual phoneme sets, labeled as IPA and CMDD. They are combined with different language models MONOLM and BILM. The comparative results are shown in Table 5.

Table 5. The comparative results of language-independent models

	TestCH1 (%)	TestCH2 (%)	TestEN (%)
IPA-MONOLM	27.1	36.0	10.8
CMDD-MONOLM	26.8	36.0	11.1
IPA-BILM	27.4	36.4	13.3
CMDD-BILM	27.2	36.2	12.8

From the table 4 and table 5, we can see that the bilingual system can achieve comparable performance to the monolingual system whether in English or Chinese testing sets. In the worst cases the WER increased 1.6% in Chinese and 3.6% in English respectively. This may be due to the dramatic increase of the size of bilingual dictionary, with nearly 10K entries.

By comparing the IPA cluster method with CMDD method in table 5, we can see that the CMDD method accepts moderate improvements in various testing conditions except combined with the monolingual language model in TestEN set. This loss may result both from the unmatchable acoustic and language model and from the asymmetric size of training data between English and Chinese. But on English testing set, using bilingual LM a significant improvement can be observed, with WER relative reduction by 3.76% than IPA-based method. Although the improvement is not dramatic as a whole, the advantage of CMDD is evident whether for adjusting the size of phoneme inventory or from the theoretical foundations. As a result, the method of confusion matrix data-driven gives us a promising belief that a great improvement can be achieved by conditioning the size of universal phoneme inventory and bilingual dictionary.

It is interesting that in the universal system the performance is improved greatly under noise conditions TestCH2, which WER is from 36.8% to 36%. For confirming the contribution of bilingual acoustic models which share data across languages we put them into monolingual systems, avoiding the influences of merged dictionary and language model. The results are listed in Table 6. These results demonstrated that the shared data across language can deduce the WER by %8.7 relatively, making the recognizer more robust under noise condition.

Table 6: Acoustic model performance comparison under noise conditions

	Monolingual baseline	IPA bilingual model	CMDD bilingual model
WER(%)	36.8	33.6	34.2

4. Summary and Future Work

In this paper, we presented the work of setting up Chinese phoneme inventory and two methods of building language-independent MSR system. By comparing the two cluster sets on different testing data and language models, the CMDD cluster method outperforms the IPA-based approach as a whole. As a by-product, the sharing data across languages provides us a new idea to improve the performance of recognizer under noise condition. In future experiments, we will try various methods to build bilingual language model and select the optimal size of universal phoneme inventory. It is also a challenging task about how to combine the experience-based method and data-driven method. This approach proposed in this paper could be generalized to other languages.

5. Acknowledgements

This work is supported by Chinese 973 program (2004CB318106), National Natural Science Foundation of China (10574140, 60535030)

6. References

[1] Zhirong Wang, Umut Tokpara, Tanja Schultz, Alex Waibel: Towards Universal Speech Recognition, Proceedings of the Fourth International Conference on Multimodal Interface (ICMI), IEEE Computer Society, (2002), 247-252

[2] F. Weng, H. Bratt, L. Neumeyer, A Stolcke: A Study of Multilingual speech recognition. In Proc. Eurospeech'97, Rhodos, Greece (1997)

[3] Ulla Uebler: Multilingual speech recognition in seven languages. Speech Communication. 35(2001), 53-69

[4] D. Lyu, R.Lyu, Y.Chiang, C.Hsu: Speech Recognition on Code-switching Among Chinese Dialects, ICASSP, I(1105-1108), (2006)

[5] IPA: The International Phonetic Association (revised to 1993) IPA Chart. Journal of the International Phonetic Association 23, (1993).

[6] M. Brian, B.Etienne: Phone Clustering Using the Bhattacharyya Distance, In Proc. ICSLP, 2005-2008, (1996)

[7] R.Bayeh, S. Lin, G. Chollet, C. Mokbel: Towards Multilingual Speech Recognition Using Data Driven Source/Target Acoustical Units Association, ICASSP'04, 521-524, (2004)

[8] E. Wong, T. Martin, T. Svendsen, S. Sridharan: Multilingual Phone Clustering for Recognition of Spontaneous Indonesian Speech Utilising Pronunciation Modeling Techniques, EUROSPEECH'03, Geneva, 3133-3136, (2003)

TONE RECOGNITION IN MANDARIN SPONTANEOUS SPEECH

Zhaojie Liu[1,2] *Pengyuan Zhang*[1] *Jian Shao*[1]
Qingwei Zhao[1] *Yonghong Yan*[1] *Ji Feng*[2]
1 ThinkIT Speech Lab, Institute of Acoustics, Chinese Academy of Sciences
2 Institute of Physics, Chinese Academy of Sciences
zliu@hccl.ioa.ac.cn

Abstract

This paper reports our study on tone recognition in Mandarin spontaneous speech, which is characterized by complicated tone behaviors. Real-Context is proposed as a new concept used in the tone modeling. First, the "error" data, which may bring negative influences to the tone model, are removed from the training data by an iterative method. Then we cluster the reduced training data into a few subsets to generate a more refined tone model. Gaussian Mixture Model (GMM) is used for the tone modeling. All experiments are based on the spontaneous speech database, Train04. Experimental results demonstrate the effectiveness of the methods.

1. Introduction

Mandarin is a kind of tonal languages. Its words are composed of one or multiple mono-syllable units called characters, and each Chinese character corresponds to a syllable associated with a lexical tone. Syllables or words with the same sequence of consonants and vowels have different tones. Usually, Mandarin contains five tones, characterized by syllable-level pitch or fundamental frequency (F0) contour pattern: high-level (tone 1), high-rising (tone 2), low-dipping (tone 3), high-falling (tone 4) and neutral (tone 5). The neutral tone often occurs in word-end or sentence-end contexts in continuous speech and does not have a stable F0 contour, which is not considered in this paper.

Accurate tone recognition plays an important role in automatic Mandarin speech recognition. Although tone recognition has been investigated for many years, relatively high recognition accuracy are only obtained in isolated words and reading speech [1][2][3][4][5][6]. Various pattern recognition methods were applied to tone recognition, which including Hidden Markov Models (HMM) [4], Gaussian Mixture Model (GMM) [7], Decision-tree Classification [8], and Support Vector Machine (SVM) [9]. Approaches fall into two major categories, namely, embedded tone modeling and explicit tone modeling. In embedded tone modeling, Pitch-related features can be added as extra dimensions in the short-time acoustic feature vector. Tone recognition is done as an integral part of the existed system. On the contrary, in explicit tone modeling, tones are independently modeled and recognized in parallel to the recognition of acoustic units. Then the results are combined in a post-processing stage [7]. Since pitch is a supra-segmental feature, which is spanned across segments and lay on a group of voiced segments, the explicit tone modeling may be more effective for modeling tone variations. For instance, the context-tone concept and supra-tone are proposed to model tone explicitly.

Spontaneous speech, as opposed to planned speech, is a more natural way in which people communicate with each other. It usually contains mispronunciations, emotional status, and other unlinguistic utterances. Besides, the speaking rate is relatively fast, leading to more serious articulation. All the phenomena mentioned above would cause tone contours to deviate from their canonical patterns and bring challenges to the tone recognition. Up to now, few efforts have been made on this respect. Recently, [10] performed some experiments to compare the capability between tone context independent and tone context dependent phoneme sets with embedded tone modeling. The purpose of this paper is to generate a refined tone model which can better describe the complex tone patterns in spontaneous speech. A new context unit is proposed to model tonal context influences, and then we cluster the samples so as to gain actual tone patterns. A kind of similar distance between two tones is also used.

The remainder of the paper is organized as follows. We first describe the tone feature and the model used in this paper. Then, the strategy of reducing "error" data in the training data is illustrated in section 3. In section 4, we introduce a new Real-Context concept and briefly present the clustering process. Some experiment results are given in section 5. And finally, section 6 provides the conclusions and some discussions.

2. Tone feature and tone model

F0 is one of the most important features of Mandarin tone. Although energy and voicing also carry some cues for tone, the cues are not as obvious as that of F0, especially for continuous speech. In present analysis, we mainly use F0 and its first derivatives as tone features. Tone is realized primarily by the F0 movement across the voiced portion of a syllable. F0 contours of four lexical tones are shown in Fig. 1, which are computed by averaging over 1000 utterances spoken by male speakers. These utterances cover most of the tonal syllables used in Mandarin.

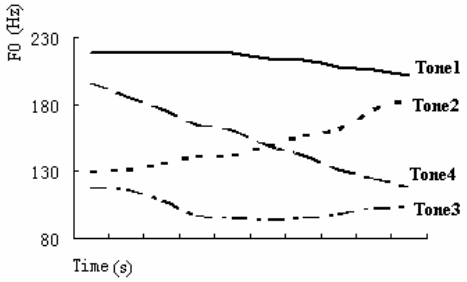


Figure 1: F0 contour of four lexical tones

2.1. Pitch extraction

Previous research suggested that the height and shape of the F0 contours, rather than the exact values at individual points, are critical for the recognition of Mandarin tones. Therefore, it is unnecessary to design a complex system to extract pitch accurately for the tone feature. A fast and robust pitch tracking algorithm (RAPT) [11] is used in our work. Two-pass normalized cross correlation function (NCCF) is calculated to generate F0 candidates, which is expressed as:

$$\phi_{i,k} = \frac{\sum_{j=m}^{m+n-1} s_j s_{j+k}}{\sqrt{e_m e_{m+k}}} \quad (1)$$

$$(k = 0, 1, ..., k-1; i = 0, 1, ..., M-1)$$

Where

$$e_m = \sum_{l=m}^{m+n-1} s_l^2$$

s_j is a sampled speech signal; i is the frame index; k is the lag; n is the sample number in an analysis window; m is the sample number in a frame.

2.2. F0 normalization

F0 is a highly variable acoustic feature affected by a number of linguistic and extra-linguistic factors. The dynamic range of F0 greatly depends on the speaker's gender, age and physiological characteristics. It is also related to the speaker's physical conditions, speaking style and emotional status. In Fig. 2, an example of tone contour in spontaneous speech is demonstrated. Within the sentence, F0 spans a range of more than 50Hz. [10] reported that the pitch of an individual adult speaker can range from 100 to 300Hz. Therefore, F0 should be normalized by the speaker independent system. In this paper, F0 is normalized by the following method:

$$F_i = K*(F0_i - \min F0_i)/(\max F0_i - \min F0_i) \quad (2)$$

where $\min F0_i$ and $\max F0_i$ are the minimum and maximum F0 of a sentence and K is a constant.

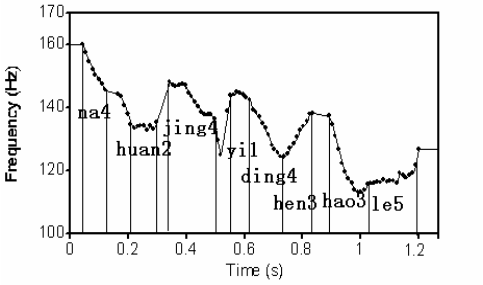


Figure 2: Pitch contour in the spontaneous speech

2.3. Tone model

In Mandarin syllable, the Final is regarded as voiced whereas the Initial is either voiced or unvoiced. Hence, F0 features for tone recognition are only extracted from the Final segments. In present study, the feature vector, which is represented by a fixed-length vector, contains F0 and its first derivatives. Every unit is divided evenly into three portions, and then the mean value is computed from each portion. Our tone model is trained by using tone labels provided by force-alignment results that should reflect the actual tone pronunciation by the speakers instead of the canonical tone marks associated with the character. Initial/Final boundaries are obtained by aligning the syllable labels with the acoustics model.

Gaussian mixture model (GMM) is employed for our tone modeling. It provides a probabilistic output that can be readily integrated into HMM based ASR systems.

The expectation-maximization (EM) algorithm is used for the estimation of GMM parameters.

3. Reducing the "error" data

Except for filled pauses, repairs, hesitations, repetitions, and disfluencies, spontaneous speech also contains other unlinguistic utterances and many noises. All of these phenomena are the main factors that may cause a very low performance of our ASR recognizer and imprecise cutting of Initial/Final boundaries. To make the tone model less affected by the "error" data, we first set a proper threshold to keep the results with high confidence in Initial/Final segmentation, and then use an iterative method to remove the data with very low scores. The methods are presented as follows:

1. Initialize the training data with high confident recognition results.
2. Train the tone model using the training data.
3. Recognize every sample in the training data. The sample, which is recognized to its labeled tone with the lowest score, will be removed from the training data.
4. Decide whether the result satisfies convergence requirement. If it does not, repeat 2 and 3 until convergence

4. Tone variations modeling

As we know, the detection of tone variations is a key point in tone recognition. In spontaneous speech, tone variations are too complicated in actual pronunciations to be described by linguistic rules. Indeed, accurate description of tone variations is the precondition of the accurate tone recognition. A great number of works have discussed this respect [10][12]. For example, [12] suggested that unit selection strategy is needed to extend to incorporate tonal context. Their statistical results showed that the influence of the left tone context is greater than the right one. Through analyzing the results, we find that samples of the tone variations account for a fair proportion in the whole database, which would lead to confusion in tone recognition. In order to devise such a strategy, Model units from left-context (L-C) to super-tone are used in this paper. Further, Real-Context (Real-C) is first proposed to model the tone variations, which is defined as follows:

$$\begin{cases} preF - \beta > 0, context \quad is \quad defined \quad H \\ preF - \alpha < 0, context \quad is \quad defined \quad L \\ other, defined \quad M \end{cases} \tag{3}$$

where α and β are set in advance, $preF$ is the real pitch position of pre-tone last dimension vector. The context will become H-*, M-*, L-*, each of which is a real context. * indicates the current tone. Real-Context, which is opposite to the tonal context, will reflect the context influence of the pre-tone pitch position in reality. And Real-Context has fewer feature vectors than supra-tone. Fig. 3 shows the difference among them. 1 denotes context tone model, 2 is supra-tone model and 3 is Real-Context model.

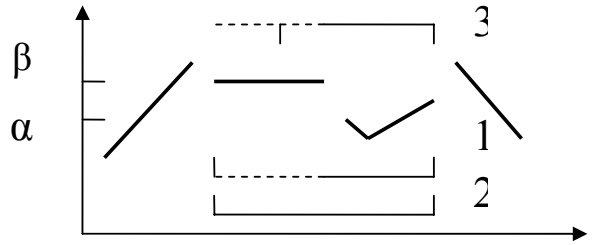


Figure 3: Tone Units

In current study, the method based on clustering is presented to model tone explicitly. Before that, we employ a kind of similarity measurements to compute the distance between two tones, which should reveal the similarity of their pitch contour shapes, and also include their different pitch height. The distance is defined:

$$Dis(X,Y) = \sqrt{\sum_{i=0}^{n} ((x_i - \bar{x}) - (y_i - \bar{y}))^2} + \left|\bar{x} - \bar{y}\right|$$

$$X = (x_0, x_1, ..., x_n), \qquad \bar{x} = \frac{1}{n}\sum_{i=0}^{n} x_i \tag{4}$$

$$Y = (y_0, y_1, ..., y_n), \qquad \bar{y} = \frac{1}{n}\sum_{i=0}^{n} y_i$$

In the clustering process, we first classify the tone data into four groups based on their labels. Second, the hierarchical clustering technique is used to find pitch contour patterns of each tone group. Initially, all samples in the group are assigned to their own clusters. Then the algorithm proceeds iteratively, joining two most similar clusters at one stage, and continuing until the condition is satisfied. The advantage of this method is that the number of clusters does not have to be fixed in advance.

5. Experiments

For all the experiments report in this paper, we use the Mandarin CTS data collected by Hong Kong University of Science and Technology in 2004, made within China and Hong Kong by mostly college students. The training set, a part of the train04, contains phonetically rich training utterances spoken by male and female speakers. The testing set comprises about utterances which are unseen from the training set. The contents are given in table 1.

Table1: Speech database used in this paper

	Num of sentences	Num of speakers	
		male	female
training	13046	50	50
testing	2729	10	10

Our recognition system is HMM-based. The acoustic models are Initial/Final models with both left and right context dependency. The acoustic feature vector is composed of 12 MFCC plus energy, and their first and second order derivatives. The recognition accuracy for base syllable is 58.9% for all the train utterances.

We perform experiments with different tone units. Table 2 shows the tone recognition results of different models. The baseline (L-C) overall accuracy is 40.8%. The best performance, 43.0%, has been attained for tone 4(T4), which has the highest percentage of distribution among all tones. Tone 3(T3) gets the lowest accuracy of 36.7%. Experiments have also been done with right-context, and the results become a little worse. It is also revealed that the Real-C outperforms L-C with 2.3% absolutely and is slightly better than supra-tone models (di-tone).

Table 2: Tone recognition accuracy with different models

Unit	T1(%)	T2(%)	T3(%)	T4(%)	total(%)	gain
L-C	42.4	39.6	36.7	43.0	40.8	ref
di-tone	45.6	40.3	38.4	45.5	42.9	2.1
Real-C	46.2	40.5	38.1	45.8	43.1	2.3

The experimental results of reducing the "error" data and clustering are given in Table 3. Improvement of 3.3% by reducing the "error" data can be attained, but clustering does not exhibit noticeable performance improvement.

Table 3: Reducing the "error" data and clustering

Unit	Reducing the "error" data(gain)	Clustering (gain)
di-tone	45.4 (2.5%)	46.1 (0.7%)
Real-C	46.4 (3.3%)	46.7 (0.3%)

6. Conclusion and discussion

In this paper, we have explored tone recognition in Mandarin spontaneous speech. The proposed new Real-Context concept

would be more helpful in modeling the tonal context influence, as shown by more than 2% improvement absolutely. Meanwhile, a refined tone model is generated by reducing the "error" data from the original data, improving the tone recognition accuracy of 3.3%. Furthermore, through clustering the training data to subsets, which may accurately describe the tone variations, we can achieve a tone recognition accuracy of 46.7%.

Nevertheless, there are still aspects of modification for the proposed methods in our study. Although tone recognition accuracy with 5.9% has been absolutely improved, it is still much lower than that of base syllable. Therefore, refining the tone modeling still requires intensive analyses, for example, tone variation rules may be added into the clustering process. Further, additional experiments are needed by other different database.

7. Acknowledgements

This work is partially supported by Chinese 973 program (2004CB318106), National Natural Science Foundation of China (10574140, 60535030), and Beijing Municipal Science & Technology Commission (Z0005189040391).

8. References

[1] C. J. Chen, R. A. Gopinath, M. D. Monkowski, M. A. Picheny and K. Shen, "New methods in continuous Mandarin speech recognition", *5th European Conference on Speech and Communication and Technology, Vol. 3, 1543-1546, 1997.*

[2] H. Huang, and F. Seide, "Pitch Tracking and Tone Features for Mandarin Speech Recognition", *Proc. ICASSP 2000, vol.3, 1523-1526, 2000.*

[3] C.J. Chen, H.P. Li, L.Q. Shen, G.K. Fu, "Recognize Tone Languages Using Pitch Information on the Main Vowel of Each Syllable", *Proc. ICASSP 2001, Vol. 1, 61-64, 2001.*

[4] W. J. Yang, J. C. Lee, Y. C Chang, and H. C. Wang, "Hidden Markov Model for Mandarin Lexical Tone Recognition," *IEEE Trans. ASSP, Vol. 36, 988-992, 1988.*

[5] G. P. Kong, S. N. Lu, "A VQ study on pitch models of disyllable in Mandarin". *ACTA ACUSTICA*, 25(2), 2000.

[6] E. Chang, J. L. Zhou, C. Huang, S. Di, K. F. Lee, "Large vocabulary mandarin speech recognition with different approaches in modeling tones", *Proc. ICSLP 2000*, 983-986 2000.

[7] Y. Qian, "Use of tone information in Cantonese LVCSR based on Generalized Character Posterior Probability Decoding," Ph. D Dissertation, *The Chinese University of Hong Kong, 2005*

[8] Cao Yang et al. Decision-tree based Mandarin tone model and its application to speech recognition. *Proc. ICASSP 2000, 1759-1762*.

[9] S. D. Dinoj et al. "Tone Recognition in Mandarin using Focus", *INTERSPEECH, 2005*, 3301-3304, 2005.

[10] J. L. Zhou, Y. Tian, Y. Shi, C. Huang, E. Chang, "Tone Articulation Modeling for Mandarin Spontaneous Speech Recognition", *Proc. ICASSP 2004, 997-1000, ICASSP 2004.*

[11] A.D. Talkin, Speech Coding and Synthesis, *Elsevier Science B.V., Amsterdam,* chapter A robust algorithm for pitch tracking (RAPT), 495-518, 1995

[12] Y. Xu, Q.E. Wang, "Pitch target and their realization: Evidence from Mandarin Chinese", *Speech communication, Vol.33, 319-337, 2001.*

Evaluation of a Feature Selection Scheme on ICA-based Filter-Bank for Speech Recognition

Neda Faraji & S.M. Ahadi

Department of Electrical Engineering,
Amirkabir University of Technology, Tehran, Iran
nfaraji@cic.aut.ac.ir sma@aut.ac.ir

Abstract

In this paper, we propose a new feature selection scheme that can contribute to an ICA-based feature extraction block for speech recognition. The initial set of speech basis functions obtained in independent component analysis training phase, has some redundancies. Thus, finding a minimal-size optimal subset of these basis functions is rather vital. On the contrary to the previous works that used reordering methods on all the frequency bands, we have introduced an algorithm that finds optimal basis functions in each discriminative frequency band. This leads to an appropriate coverage of various frequency components and easy extension to other data is also provided. Our experiments show that the proposed method is very useful, specifically in larger vocabulary size tasks, where the selected basis functions trained using a limited dataset, may get localized in certain frequency bands and not appropriately generalized to residual dataset. The proposed algorithm surmounts this problem by a local reordering method in which contribution of a basis function is specified with three factors: class separability power, energy and central frequency. The experiments on a Persian continuous speech corpus indicated that the proposed method has led to 17% improvement in noisy condition recognition rate in comparison to a conventional MFCC-based system.

1. Introduction

A fundamental problem in applied digital signal processing is to find suitable representations for image, audio or other kind of data for applications such as recognition and denoising. Data representations are often based on linear transformations. Standard linear transformations widely used in signal processing are the Fourier, Haar, cosine transform, etc. It would be most useful to estimate the linear transformation from the data itself, in which case the transform could be ideally adapted to the kind of data that is being processed [1]. Independent Component Analysis (ICA) is a data-driven method that can capture the higher order statistics from the signal. ICA can separate independent components from the signals which are mixtures of the unknown sources and it can be applied to image, speech and medical signal processing [2].

In [3], ICA was used for feature extraction of speech signal. The extracted ICA-based features were then applied to an Isolated-word recognition task. Since the ICA algorithm finds the independent components corresponding to the dimensionality of the input, it may result in redundant components. However, the extracted independent components of speech correspond to the sources of speech production. Some of these sources are irrelevant sources. In reality, these may not be useful for speech recognition and should be removed. Identifying the irrelevant and redundant sources and their removal could be carried out by a feature selection method. In [3], two measures have been used for reordering and selecting of basis vectors:

1) The energy of the basis vector.
2) The variance of the basis vector coefficient.

Although the basis vectors selected with these two simple methods outperformed mel-scale filter-bank in capturing the higher order structures of speech, our experiments have shown that two mentioned methods lead to high degradations of recognition rate in some conditions. In reality, the overall ICA-based feature extraction performance is very sensitive to applied feature selection method. We should, therefore, use an effective feature selection method that considers all aspects of a given problem, here speech recognition.

We propose a new feature selection algorithm that minimizes the available problems in two aforementioned methods by obtaining a nearly optimal subset of the initial ICA-based filters with respect to adaptation and generalization issues. The considerable effect of proposed method in improving recognition rate has been approved by our experiments on a Persian continuous speech corpus. Also, the recognition results obtained on both Aurora 2 and Persian tasks show that the ICA-based features are robust in noisy conditions. It should be noted that in the case of Aurora 2 task, the global reordering method was adequate and only the effect of initial number of filters is evaluated.

The paper is organized as follows. In section 2 we briefly review the use of ICA technique in feature extraction of speech. The proposed feature selection method is presented in section 3. Experimental results are discussed in section 4 and conclusions are drawn in section 5.

2. Extracting speech features using ICA

2.1. Extracting basis vectors of speech

The linear model of independent component analysis assumes that the observation is a linear mixture of the independent components and is represented as:

$$\mathbf{x} = \mathbf{A}\mathbf{s} = \sum_{i=1}^{N} \mathbf{a}_i s_i \ , \ \ \mathbf{s} = \mathbf{W}\mathbf{x} \qquad (1)$$

In this model, $\mathbf{a}_i$ is a basis vector that contributes to the generation of the observed data with source coefficient s_i. The estimation of sources and basis functions could be done by maximization of negentropy $J(s_i)$ [1]:

$$max J(s_i) = J(\mathbf{w}_i^T \mathbf{x}) = \left[E\{G(\mathbf{w}_i^T \mathbf{x})\} - E\{G(v)\} \right]^2 \qquad (2)$$

where $\mathbf{w}_i$ is the *i*th row of mixing matrix $\mathbf{W}$, G is a contrast function and v is a standardized Gaussian variable. In this work, we take the exponential function as the contrast function:

$$G(u) = -\exp(-\frac{1}{2}u^2) \qquad (3)$$

To find basis vectors of speech, short-time segments from speech signals are constructed and using an iterative algorithm, negentropy is maximized. We take a fixed point version of the algorithm and an iterative symmetric decorrelation scheme. Finally, the columns of **A** matrix are obtained from the inverse of the estimated **W**. These columns are considered as the initial set of ICA-derived filter-bank.

2.2. Selection of dominant basis vectors

The dimension of the extracted basis vectors is equal to the dimension of the short-time segments[3]. It is desirable to select dominant basis vectors from the initial set so that the feature extraction method is computationally comparable with the conventional one. Also, irrelevant and redundant basis functions are eliminated. We could decide based on the L2-norm of the basis vector or the variance of the basis vector coefficient [3]. The block diagram of ICA-based speech recognition system has been shown in Fig. 1.

3. Proposed feature selection algorithm

The feature selection methods utilized in [3], may omit filters in some frequency bands. We face this effect in certain conditions. For example, most of high energy basis vectors are localized in low frequency bands. Thus, the feature selection method may suppress the high frequency basis vectors. However, when the dimensionality of the training speech frames in ICA algorithm is increased, global reordering methods may result in almost low frequency dominant basis functions. Also, the ICA-derived filters are highly adapted to its normally limited training dataset. It emphasizes on some frequency bands available in ICA training dataset. Thus, the feature selection algorithm may remove less significant filters. These removed filters may have very important role in extracting information from the non-training data. In practice, there should be a trade-off between adaptation and generalization. The flow diagram of the algorithm has been demonstrated in Fig. 2.

3.1. Measure of comparing basis vectors

It is preferable to choose a measure that considers the classification problem more strongly. Thus, we have introduced a weighted version of L2-norm in which L2-norm of a basis vector is weighted with its Proportion of Variance (PoV).

$$\text{measure}(j) = \text{PoV}(j) \times \text{L2-norm}(j) \qquad (4)$$

where *j* indicates *j*th filter. PoV [4] is calculated using the ratio of between-class variance to within-class variance. The PoV measure of a given basis vector, shows its capability in

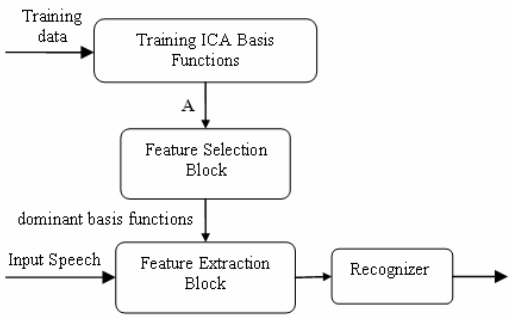


Figure 1: Block Diagram of ICA-based speech recognition system

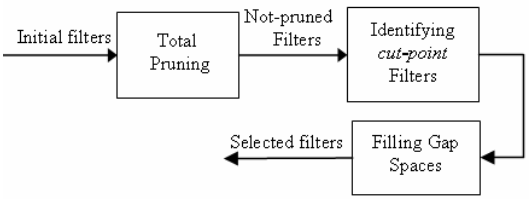


Figure 2 : Proposed Feature Selection Block.

separating different classes, and along with L2-norm directs us to choose those high energy basis vectors that also have high discrimination power. We assume that the number of classes is *L*. Then, the between-class and within-class variances for a given variable x, are defined as follows [5]:

$$S_W = \sum_{i=1}^{L} P_i E\left[(x-\mu_i)(x-\mu_i)^T \,|\, w_i \right] = \sum_{i=1}^{L} P_i \Sigma_i \qquad (5)$$

$$S_B = \sum_{i=1}^{L} P_i \left[(\mu-\mu_i)(\mu-\mu_i)^T \right], \quad \mu = \sum_{i=1}^{L} P_i \mu_i \qquad (6)$$

where μ_i, Σ_i and P_i are the mean, variance and the a priori probability of the *i*th class respectively. Then we write:

$$\text{PoV} = \frac{S_B}{S_W}. \qquad (7)$$

The aforementioned measure in Eq. 4 is very useful when we want to select dominant basis vectors from a set of basis vectors with the same central frequencies. It is probable that a low-power high-frequency basis vector has more capability to discriminate classes than a high-power low-frequency one. In other words, comparing two basis vectors without considering their central frequencies is not correct and finally may result in giving more weight to a certain band in comparison to the others.

This has been the main motivation behind our work on providing an algorithm to select dominant basis vectors automatically. In reality, the problem may be solved with local decisions instead of global ones.

The proposed algorithm includes three parts:

1) Total pruning
2) Identifying the *cut-point* filters
3) Filling gap spaces.

3.2. Total pruning

In this step, each filter competes with the filters whose central frequencies are near its. The competition is local and the difference between central frequencies should be below a threshold value. These filters constitute a group and one or more filters from this group with the largest PoV-weighted L2-norm are selected. It should be noted that a very small threshold value may lead to no filter pruning. Meanwhile, a very large threshold value destroys the locality assumption. We have selected a threshold value equal to 80 Hz. This value is about half of the minimum bandwidth in a conventional mel-scale filter-bank.

3.3. Identifying *cut-point* filters

The aim of this part of algorithm is identifying the critical filters. The critical filters have more distinct characteristics in comparison to the others and would be kept to improve generalization and classification.

The central frequency of a filter is an index that can separate it from the others. A hierarchical clustering algorithm is applied to the central frequency values to reach a certain number of central frequency clusters. The number of clusters is increased by one, if no cluster is found to have a single member. Single member clusters include the filters that have more distinct central frequencies than the other filters. These filters have an important role in generalization.

Concurrently, another clustering is performed on PoV-weighted L2-norm values to reach 3 clusters corresponding to low, medium and high values. In this case, the single member clusters have more distinct PoV-weighted L2-norm values. We are interested in keeping the filters with high PoV-weighted L2-norms. These filters can improve the classification. The filters found from two aforementioned clustering approaches are *cut-point* filters and can divide the whole frequency band into sub-bands.

After finding the initial *cut-point* filters, the two explained clusterings are performed on each sub-band filters. Sub-band clustering is performed from low to high frequency sub-bands. For the algorithm to be well conditioned, we have applied constraints on it. The constraints were put on:

1) The initial number of central frequency clusters in the beginning of the algorithm.
2) The maximum number of clusters in each sub-band.

The large number of clusters in the beginning of the algorithm lead to fast but not accurate converging. We begin the algorithm with the minimum number of central frequency clusters, so that it can find *cut-points* (single-member clusters) gradually.

In sub-band clustering, the low frequency sub-bands are prior to the high frequency sub-bands. If we do not limit the number of clusters in each sub-band, the algorithm finds the *cut-point*s only in low frequency bands. However, the maximum number of clusters in each sub-band specifies the final number of *cut-point* filters found.

3.4. Filling gap space

According to the values of constraints, there would remain some gaps between the *cut-point* filters found. We heuristically fill these gap spaces.

Table 1: The recognition results in single Gaussian HMM and Persian corpus. The number of filters in each sub-band has been selected heuristically.

		Recognition Rate
MFCC		43.57
ICA	**Global reordering**	32.22
	4 sub-band	33.92
	8 sub-band	39.11
	13 sub-band	40.08
	18 sub-band	43.68
	Proposed method	**44.52**

Table 2: The recognition rate in different noisy conditions and 15-Gaussian mixture HMM.

	Recognition Rate	
	ICA	**MFCC**
Clean	72.83	76.43
Babble	27.54	20.01
Car	70.96	50.32
F16	28.72	19.6
Average	**42.4**	**29.97**

Table 3: The recognition rate of ICA-based features in AURORA 2 task (12 Ceps.coeff+C_0 in ICA features, 12 Ceps.coeff+log Energy in baseline system)

	Recognition Rate		
	35	**50**	**baseline**
Set A	65.44	63.94	61.13
Set B	64.85	64.1	55.57
Set C	60.28	65.57	66.68
Average	**63.52**	**64.53**	**61.12**

4. Experiments and results

4.1. Experiments on Farsi continuous speech corpus

The Farsi continuous speech corpus, FARSDAT was used in this work [6]. This corpus is the only continuous speech corpus of Farsi, which is available to public. It consists of 6000 sentences from 300 speakers, each uttering 20 sentences selected from a set of 392 available sentences. We have used 1819 sentences from 91 speakers for building a 3-state phoneme-based HMM system. Also, the test set includes 888 sentences from 46 speakers. Training and test have been carried out using HTK [7] and with different number of Gaussian mixtures in each state. The noise was then added to the speech in different SNRs. The noise data was extracted from the NATO RSG-10 corpus [8]. We have considered babble, car and F16 noises and added them to the clean signal at 20, 15, 10, 5 and 0 dB SNRs. Our experiments were carried out using MFCC (for comparison purposes) and ICA features. The features in two cases were computed using 25 msec. frames with 10 msec. of frame shifts. Pre-emphasis coefficient was set to 0.97 and a Hamming window was applied. The feature vectors for the two methods were composed of 12 cepstral and a log-energy parameter.

For extracting MFCC features, a 24-channel mel-scale filter-bank was used. Also, for training the initial set of ICA-based filter-bank, the 100 sample short-time segments with frame shifts of 30 samples were used. These segments were

constructed using 110 sentences from 11 speakers. After training, the proposed feature selection block selected 24 dominant filters from a total of 100 filters. The selected filters were then replaced mel-scale filters in feature extraction block.

In step 1 of feature selection algorithm, threshold value was set to 80Hz and 50 filters were pruned. From the 50 remaining filters, 20 *cut-point* filters were selected in step 2. This final number of *cut-point* filters was obtained with these parameters:

- The initial number of central frequency clusters was set to 4 at the beginning of the algorithm.
- The maximum number of clusters in each sub-band was set to the number of sub-band members minus one.
- The initial number of clusters in each sub-band clustering was set to the minimum value between 4 and half of the sub-band members.

Finally, 4 filters filled gap spaces to get 24 filters. In all steps, the comparison of filters was carried out by PoV-weighted L2-norm measure. Also, the PoV values of 100 initial filters were calculated using 119 phoneme-labeled sentences.

Table 1 lists the recognition results of ICA-based features using sub-band reordering methods, in which the whole frequency band was divided into some sub-bands and in each sub-band, the L2-norm measure was used for selecting dominant filters. The number of filters in each sub-band is selected according to the number of mel-scaled filters in that sub-band. The results have been brought from clean condition and 1-mixture phoneme-based HMM recognizer. Also, the recognition result of MFCC features is given for comparison. The indicated results show the considerable effect of used feature selection method in overall result. As seen in Table. 1, proposed feature selection algorithm has increased the recognition rate about 1.6% in comparison to MFCC.

Table 2 lists the recognition results obtained in clean and noisy conditions, on a 15-component HMM-based system using the proposed feature selection method. As indicated in this table, the performance of ICA-based features is worse than MFCC in clean condition when we used 15-Gaussian HMM system. This effect is the natural result of lower-variance ICA features. The results also show the robustness of ICA features in noisy conditions. The average values mentioned in this table are calculated over the results obtained from 0 dB to 20 dB SNRs , omitting the clean ones.

The results obtained can be listed as follows:

1) The feature selection method is very important in ICA-based feature extraction.
2) The ICA-based features improve the robustness of speech recognition system.
3) When the number of Gaussian components is increased, the recognition rate of ICA-based features is degraded in comparison to MFCC features.

4.2. Experiments on AURORA 2 speech task

In this section, we briefly review the result of experiments on AURORA 2 task [9], adopting global reordering method and various initial number of ICA-based filters. It should be noted that ICA training phase was carried out using 22 files from 22 selected speakers in this task.

The results are reported in Table 3 and the following conclusions can be listed:

1) The global reordering is adequate in this smaller size task.
2) The ICA-based features are more robust than MFCC features.
3) The initial number of ICA-derived filters is effective in obtained recognition rate.

5. Conclusions

In this paper, we tried to emphasize on the significant role of the feature selection algorithm on the overall performance of ICA-based speech recognition systems. The ICA-derived filters are adapted to processing data, specifically with data used in training ICA basis functions. Therefore, they are able to extract maximum higher-order information from data. The insufficient feature selection algorithm can remove some of the essential filters and lead to degradation of recognition rate. However, if an appropriate algorithm is used for selecting dominant filters in ICA feature extraction block, the recognition rate in noisy conditions shows great improvements in comparison to the baseline. This effect also originates from adapting of ICA filters with data. It seems that the ICA-based features contain the maximum amount of data information, while extract minimum information of noise data. The initial number of filters is also an important parameter in obtaining filters with appropriate frequency resolutions and then better recognition rate.

6. Acknowledgement

This work was in part supported by a grant from the Iran Telecommunications Research Center (ITRC).

7. References

[1] A. Hyvarinen, J. Karhunen, E. Oja, *Independent Component Analysis*. John Wiley & Sons, New York: 2001.

[2] M. Kotani, Y. Shirata, S. Maekawa, S. Ozawa, K. Akazawa, "Application of independent component analysis to feature extraction of speech," in *Proc. IJCNN, Vol. 5, pp. 2981-2985, 1999.*

[3] J.H Lee, H.Y Jung, T.W Lee, S.Y Lee "Speech feature extraction using independent component analysis," in *Proc. ICASSP .Vol. 3, pp. 1631-1634, 2000.*

[4] E.K. Ekenel, N.Sankur, "Feature selection in the independent component subspace for face recognition," in *Pattern Recognition Letters*, *Vol. 25, pp. 1377-1388, June 2004.*

[5] N. Malayath, H. Hermansky, "Data-driven spectral basis functions for automatic speech recognition," in *Speech Communication*, *Vol. 40, Issue 4, pp. 449-466, June 2003.*

[6] D.Gharavian, S.M.Ahadi, " Evaluation of the effect of stress on formants in Farsi vowels," in *Proc. ICASSP*, *vol. 1, pp. 661-664, Montreal, May 2004.*

[7] The hidden Markov model toolkit available from http://htk.eng.cam.ac.uk.

[8] Available from http://spib.rice.edu/spib/select_noise.html

[9] H.G. Hirsch, D. Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ASRU*, *pp. 181-188, September 2000.*

A Robust Endpoint Detection Algorithm Based on Identification of the Noise Nature

Denilson C. Silva

Program of Electrical Engineering - COPPE
Signal Processing Laboratory
Federal University of Rio de Janeiro, Rio de Janeiro, Brazil
denilson@lps.ufrj.br

Abstract

The endpoint detection of speech is still a big problem in situations of speech recognition in noisy environments. While traditional methods concentrate on finding speech in noise, the proposed technique is based on noise identification through HMMs, associated with both *SNR* and euclidean distance of the log-energy calculated on a frame-by-frame basis. Computer experiments confirm that the proposed algorithm gives rise to a considerable improvement on the precision of endpoint detection, specially in severely adverse conditions where the *SNR* is very low.

1. Introduction

In many applications of speech signal processing, determination of the endpoints of utterances is necessary. The traditional methods of endpoints detection based on both energy and zero crossing rate work very well with clean speech [1]. When we have utterances with fricatives, for instance, the endpoint detection might become complicated if the delimitation process happens in a noisy environment.

Several works have been seeking to solve the subject of the endpoint detection in noisy environments [2, 3], but the obtained results are extremely sensitive to the signal-to-noise ratio (*SNR*).

In this article a method is proposed for endpoint detection in utterance for speech recognition based on the principle of identification of the noise nature that contaminates the signal, through a classification on each frame using hidden Markov models (HMMs), delimiting the intervals with speech starting from the identification of frames with noise only. The *SNR* and the euclidean distance of the log-energy of each frame are also used to accomplish a refinement in the detection. The proposed method results in a significant precision improvement, particularly for very adverse conditions, as computer experiments show.

This article is organized as follows. In Section 2, the proposed endpoint detection algorithm is introduced. In Section 3, the database is described. In Section 4, results of the accomplished tests as well as comparisons with the traditional method based on both energy and zero crossing rate are shown. Finally, in Section 5, we present the conclusions.

2. Proposed method

The proposed endpoints detection method is accomplished through three processes of decision beginning with a frame-by-frame analysis: the identification of the noise nature [4] [5], *SNR* and the euclidean distance of the log-energy (Figure 1). We define that a frame with positive flag $(+1)$ contains noise only, while a frame with negative flag (-1) contains both speech and noise.

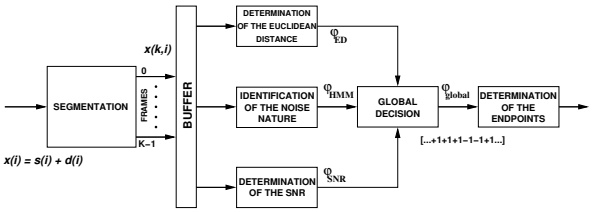


Figure 1: *Proposed method.*

2.1. Identification of the noise nature

Consider the noisy signal $x(i)$ a composition of the clean speech $s(i)$ with the additive noise $d(i)$, that is, $x(i) = s(i) + d(i)$.

The signal $x(i)$ is segmented into K frames with N size and overlap of L samples:

$$x(k,i) = x(k(N-L)+i) \tag{1}$$

where $0 \leq k \leq K-1$ and $0 \leq i \leq N-1$.

The process of identification of the noise nature is accomplished through a classification of the frames of the noisy signal ($x(k,i)$), using HMMs, among the types of noise involved in the training. The settings used on tests were 18 parameters per subframes of 23 ms, 50% of overlap, Hamming windowing, left-right HMM with skip allowed, four states, four models and 128 centroids. The stages of the identification are explained below.

2.1.1. Extraction of parameters

The extracted parameters on each subframe were: one of spectral entropy (as described in [3]), one of zero crossing rate and 16 of log-energy, described in Table 1.

The pth log-energy parameter, $LogE(k,p)$, is defined as the energy contained in the pth subband for frame k.

2.1.2. Criterion of decision

The noisy utterances are segmented in frames of 46 ms and overlap of 80%. The frames are identified by HMM, in agreement with trained noise models. The results are registered as

Table 1: Algorithm for extraction of log-energy parameters.

1. Consider the noisy signal spectrum, $\Gamma(k, j)$, calculated by FFT.
2. Calculate the log-spectrum ($\Psi(k, j)$):
$\Psi(k, j) = 10 \log 10 \left(\frac{\Gamma(k,j)}{f_s m} \right)$, where $0 < j < F - 1$, F is the number of frequency components, f_s is the sampling frequency and m is the number of points used in the FFT;
3. Estimate the log-spectral envelope ($\Phi(k, j)$):
a. Initialization:
$\Phi(k, 0) = \Psi(k, 0)$
$\Phi(k, F - 1) = \Psi(k, F - 1)$
b. Iteration:
for $j = 1$ to $F - 2$ **do**
  if $\Psi(k, j) > \Psi(k, j - 1)$ and
  $\Psi(k, j) > \Psi(k, j + 1)$ **then**
    $\Phi(k, j) = \Psi(k, j)$
  else
    $\Phi(k, j) = 0$
  end if
end for
c. Interpolate the values between each pair of adjacent non-null values by Newton's divided difference method [6].
5. Divide the log-spectral envelope into P subbands with F' frequencies in each subband.
6. Extract P log-energy parameters ($LogE(k, p)$).
$LogE(k, p) = \sum_{j=0}^{F'-1} \Phi(k, (pF' + j))$, where $0 < p < P - 1$

noise frame. In the following, a count is accomplished to verify which noise type received the largest number of classifications in all signal (*noise signal*).

$\varphi_{HMM}(k)$ is the flag attributed to the *k*th frame.

$$\varphi_{HMM}(k) = \begin{cases} -1, & \text{if } \textit{noise frame} \neq \textit{noise signal} \\ +1, & \text{if } \textit{noise frame} = \textit{noise signal} \end{cases}$$

2.2. Determination of the *SNR*

The *SNR* of each frame is calculated in accordance with the algorithm in Table 2 and a flag $\varphi_{SNR}(k)$ is assigned.

$th1$ received empirically the value of 1.25, if $\xi < 10$dB, and 2.50, if $\xi \geq 10$dB. M and Q were attributed values 15 and 5, respectively.

ξ is the estimated *SNR* for all signal, calculated as:

$$\xi = 10 \log 10 \left(\frac{\sum_{k=0}^{K'-1} \hat{\sigma}_x^2(k) - \sum_{k=0}^{K'-1} \hat{\sigma}_d^2(k)}{\sum_{k=0}^{K'-1} \hat{\sigma}_d^2(k)} \right) \quad (2)$$

where both $\hat{\sigma}_x^2(k)$ and $\hat{\sigma}_d^2(k)$ are calculated such as $\sigma_x^2(k)$ and $\sigma_d^2(k)$, from algorithm in Table 2, without overlap of the K' frames, in other words, $x(k, i) = x(kN + i)$, for $0 \leq k \leq K' - 1$.

2.3. Determination of the euclidean distance

The euclidean distance is evaluated based on vectors obtained from the log-energy parameters of $x(i)$. The flag assignment algorithm can be found in Table 3.

Table 2: Algorithm for determination of the flags by *SNR*.

1. Calculate the variance of noisy signal $x(k, i)$ ($\sigma_x^2(k)$).
$\sigma_x^2(k) = \frac{1}{N} \sum_{i=0}^{N-1} [x(k, i) - \mu(k)]^2$
a. Here $\mu(k) = \frac{1}{N} \sum_{i=0}^{N-1} x(k, i)$
2. Estimate the variance of noise using a smoothing filter [7] ($\sigma_d^2(k)$).
a. Considering M frames inside the initial interval of the utterance, without the presence of the speech signal $s(i)$, calculate the relative *SNR* ($\Xi(k)$).
$\Xi(k) = \frac{\sigma_x^2(k)}{\frac{1}{M} \sum_{n=0}^{M-1} \sigma_x^2(n)}$
b. Calculate the $\alpha(k)$ parameter.
$\alpha(k) = 1 - min(1, \Xi(k)^{-Q})$
c. Estimate noise.
$\sigma_d^2(k) = \alpha(k)\sigma_d^2(k - 1) + (1 - \alpha(k))\sigma_x^2(k)$
3. Define flag.

$$\varphi_{SNR}(k) = \begin{cases} -1, & \text{se } \sigma_x^2(k)/\sigma_d^2(k) \geq th1 \\ +1, & \text{se } \sigma_x^2(k)/\sigma_d^2(k) < th1 \end{cases}$$

$\varphi_{ED}(k)$ is the flag attributed to *k*th frame by euclidean distance criterion. $th2$ received empirically the value of 1.4.

2.4. Determination of the endpoints

After all frames receive three flags, they are labeled again, receiving one flag only, ($\varphi_{global}(k)$):

if $\varphi_{HMM}(k) = \varphi_{SNR}(k) = \varphi_{ED}(k) = +1$ **then**
  $\varphi_{global}(k) = +1$
else
  $\varphi_{global}(k) = -1$
end if

The global flags are analyzed starting from frame 0 in direction to the last, looking for the first sequence of 15 frames with negative flags, where the first frame of the sequence characterizes the ending of a possible initial interval containing only noise. Starting from the last frame in direction to the frame 0, a sequence of 15 frames with negative flag is also sought, where the last frame of the sequence characterizes the beginning of a possible final interval containing only noise.

3. Database

The utterances used in this article were collected of the database described in [8], where we have 10 command and control isolated words. The sampling rate is 11025 Hz.

The noise database was collected from [9], with original sampling rate of 19980 Hz, and resampling to 11025 Hz. Four noise types were selected: WHITE, PINK, VOLVO (car interior) and BABBLE.

The noisy speech was formed through the addition of selected noise to clean speech with *SNR* from 0 to 20dB.

4. Experimental Results

The performance could be appraised comparing the results obtained in detection with obtained reference values of manual

Table 3: Algorithm for determination of the flags by euclidean distance.

1. Calculate the euclidean distance $ED_x(k)$, on each frame, between the log-energy vectors of signal $x(i)$ and a reference vector.
a. Calculate the reference vector, $LogE_{ref}(p)$
$LogE_{ref}(p) = \frac{1}{M}\sum_{k=0}^{M-1} LogE(k,p)$
b. Calculate the distance.
$ED_x(k) = \sqrt{\sum_{p=0}^{P-1}(LogE(k,p) - LogE_{ref}(p))^2}$
2. Estimate ED_d.
$ED_d = \max_{0\leq k\leq M-1}\{ED_x(k)\}$
3. Define flag.

$$\varphi_{ED}(k) \quad = \quad \begin{cases} -1, & \text{if } ED_x(k)/ED_d \geq th2 \\ +1, & \text{if } ED_x(k)/ED_d < th2 \end{cases}$$

Table 4: Table containing average percents of error reduction rate

	BABBLE	PINK	VOLVO	WHITE
Beginning	6.82%	12.23%	−2.65%	12.05%
Ending	20.81%	26.24%	0.70%	24.88%

clipping.

Initially it was made a training of discrete HMMs of [8], with 500 segments of 100 ms for each of the four types of noise and 50% of overlap. An experiment was undertaken with 121 corrupted utterances, setting both the *SNR* and the noise, observing the percent of mistake in detection along the several values of *SNR* both at beginning (ε_b) and at ending (ε_e), where B and E are, respectively, the beginning and the ending by manual clipping, $\mathcal{E}_b$ and $\mathcal{E}_e$ are, respectively, the detected points by the proposed system.

$$\varepsilon_b = \frac{|B - \mathcal{E}_b|}{E - B} \times 100\% \tag{3}$$

$$\varepsilon_e = \frac{|E - \mathcal{E}_e|}{E - B} \times 100\% \tag{4}$$

The average reduction of the error rate in detection, comparatively to the method described in [1], is shown in Table 4. We can see that there was a considerable error reduction in every case, except for the car interior noise, where no significant change was observed.

The results for each type of noise of the traditional [1] and proposed technique can be found in Tables 5, 6, 7 and 8. It is also noticed that the best performance of the introduced detector happens in low-*SNR* zone.

Figure 2 exemplifies the endpoint detection on low-energy sounds. We can see that the proposed method kept all useful information signal.

5. Conclusions

In this paper a robust endpoint detection method was proposed based on identification of the noise nature using HMMs, associated to *SNR* and to euclidean distance of log-energy parameters, that were used to adjust the detection in segments with low-energy sounds. With this method we try to detect the frames that contain only noise and to separate them from the useful information of the signal. Computer experiments with 121 utterances corrupted by four different noise types and with varying levels of *SNR* from 0 to 20dB were presented. It can be seen from the results that in severe adverse conditions, when the *SNR* is very low, that the proposed method offers a considerably more precise detection of endpoints as compared to the traditional technique. The proposal of an algorithm that, based on the estimation of *SNR* and type of noise, can decide

Table 5: Percentual error of the endpoint detection in babble noise environment

SNR	Traditional		Proposed	
	beginning	ending	beginning	ending
0dB	40.0%	80.67%	11.96%	20.55%
5dB	15.83%	39.41%	9.38%	10.78%
10dB	8.95%	19.11%	7.28%	8.66%
15dB	6.54%	10.85%	6.87%	6.33%
20dB	5.34%	6.04%	7.04%	5.71%

Table 6: Percentual error of the endpoint detection in pink noise environment

SNR	Traditional		Proposed	
	beginning	ending	beginning	ending
0dB	68.52%	114.26%	15.20%	36.55%
5dB	19.07%	53.69%	11.87%	20.35%
10dB	14.04%	34.13%	11.18%	17.32%
15dB	9.79%	14.66%	10.08%	9.95%
20dB	7.10%	6.63%	9.03%	8.04%

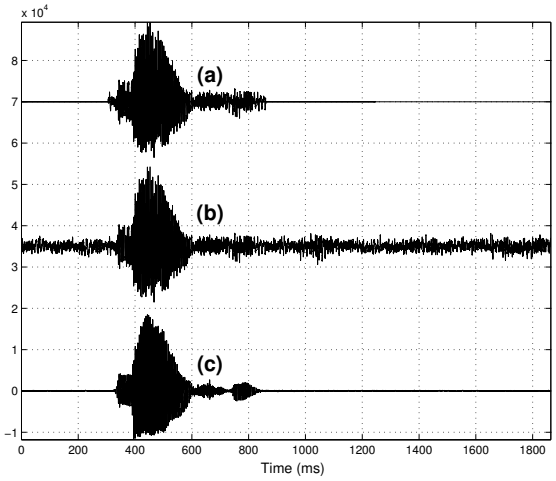


Figure 2: Endpoint detection based on identification of the noise nature. (a) Endpoint detection by the proposed method. (b) Corrupted signal by babble noise at 5dB *SNR* and (c) Clean speech referring to "baixo" (which means "down") utterance.

Table 7: Percentual error of the endpoint detection inside car

SNR	Traditional		Proposed	
	beginning	ending	beginning	ending
0dB	9.67%	15.02%	7.52%	5.64%
5dB	7.43%	10.86%	8.22%	6.08%
10dB	7.55%	6.71%	9.38%	6.16%
15dB	8.40%	5.04%	12.66%	8.24%
20dB	7.09%	4.53%	15.61%	12.53%

Table 8: Percentual error of the endpoint detection in white noise environment

SNR	Traditional		Proposed	
	beginning	ending	beginning	ending
0dB	62.44%	105.94%	13.04%	29.29%
5dB	18.31%	50.79%	11.49%	21.94%
10dB	13.68%	33.22%	9.87%	16.44%
15dB	9.61%	15.19%	8.05%	9.25%
20dB	6.93%	6.57%	8.29%	10.37%

which is the most suitable technique for end point detection and recognition in adverse conditions is subject of ongoing research.

6. References

[1] Teruszkin, R., Consort, T. A. and Resende Jr., F. G. V., "Endpoint detection analysis for an implementation of a speech recognition system applied to robot control", In Proc. SAWCAS, November, 2001.

[2] Bou-Ghazale, S. E. and Assaleh, K., "A robust endpoint detection of speech for noisy environment with application to automatic speech recognition", In IEEE International Conference on Acoustics, Speech, and Signal Processing, vol.4, pp. 3808-3811, May, 2002.

[3] Jia-Lin Shen, Jeih-Weih Hung and Lin-Shan Lee, "Robust entropy-based endpoint detection for speech recognition in noisy environments", In International Conference on Spoken Language Processing, Sydney, November, 1998.

[4] Silva, D. C. and Resende Jr., F. G. V., "Identification of the Noise Nature based on HMMs" (in Portuguese), In Simpósio Brasileiro de Telecomunicações, September, 2004.

[5] Silva, D. C., "Identification of the Noise Nature with Application in Robust Speech Recognition" (in Portuguese), Masters Thesis, Federal University of Rio de Janeiro, February, 2005.

[6] Hildebrand, F. B., Introduction to Numerical Analysis, McGraw-Hill, New York, Second Edition, 1974.

[7] Lin, L., Holmes, W. H. and Ambikairajah, E., "Subband noise estimation for enhancement using a perceptual Wiener filter", In IEEE International Conference on Acoustics, Speech, and Signal Processing, vol.1, pp.80-83, April, 2003.

[8] Teruszkin, R., Resende Jr., F. G. V., Villas-Boas, S. B. and Lizarralde, F., "Object-oriented speech recognition library applied to robot control" (in Portuguese), In Congresso Brasileiro de Automática, September, 2002.

[9] http://spib.rice.edu/pib/select_noise.html, "Signal Processing Information Base (SPIB)",September, 2002.

EMD Analysis of Speech Signal in Voiced Mode

Aïcha Bouzid[1], Noureddine Ellouze[2]

[1]Institute of Electronic and Communication of Sfax, Sfax, Tunisie
[2]National School of Engineers of Tunis, Tunis, Tunisie
N.Ellouze@enit.rnu.tn, bouzidacha@yahoo.fr

Abstract

Like almost all natural phenomena, speech is the result of many nonlinearly interacting processes; therefore any linear analysis has the potential risk of underestimating, or even missing, a great amount of information content. Recently the technique of Empirical Mode Decomposition (EMD) has been proposed as a new tool for the analysis for nonlinear and nonstationary data. We applied EMD analysis to decompose speech signal into intrinsic oscillatory modes. Besides, the LPC analysis of each mode provides an estimation of formants.

1. Introduction

Speech signal, as with many real-world signals, are nonstationary, making Fourier analysis unsatisfying since the frequency contents changes across the time. In time-frequency analysis, we analyse the frequency content across a small span of time and then move to another time position [1] and [2]. The major drawback of most time- frequency transforms is that the rectangular tiling of the time frequency plane does not match the shape of many signals.

On the other hand, basis decomposition techniques such as Fourier decomposition or the wavelet decomposition have also been used to analyse real world signals [3]. The main drawback of these approaches is that the basis functions are fixed, and do not necessarily match varying nature of signals.

In this paper, we use the empirical mode decomposition (EMD), first introduced by N. E. Huang and al. in 1998 [4]. This technique adaptively decomposes a signal into oscillating components. The different components match the signal itself very well. Because the approach is algorithmic, it does not allow expressing the different components in closed form. The EMD is in fact type of adaptive wavelet decomposition whose sub bands are built as needed to separate the different components of the signal.

EMD was applied to a number of real situations [5], [6] and [7], motivating us to consider work on naturally speech decomposition in order to delimit EMD limitations and possibilities.

The out line of the present paper is as follows. Firstly we introduce the new non linear decomposition technique known as empirical mode decomposition. Then we apply this technique to decompose a simple signal consisting of a sum of three pure frequencies. The second section presents results of this approach applied to speech signal decomposition. Computing the LPC analysis of different intrinsic mode functions provides measure of formant speaker. Last section concludes this work.

2. Empirical mode decomposition

The empirical mode decomposition is a signal processing technique proposed to extract all the oscillatory modes embedded in a signal without any requirement of stationarity or linearity of the data. The goal of this procedure is to decompose a time series into components with well defined instantaneous frequency by empirically identifying the physical time scales intrinsic to the data that is the time lapse between successive extrema [8].

Each characteristic oscillatory mode extracted, named Intrinsic Mode Function (IMF), and satisfies the following properties: an IMF is symmetric, has unique local frequency, and different IMFs do not exihibit the same frequency at the same time. In other words the IMFs are characterized by having the number of extrema and the number of zero crossings equal (or different at most by one), and the mean value between the upper and lower envelope equal to zero at any point.

The algorithm operates through six steps [4]:

1) Identification of all the extrema (maxima and minima) of the series x(t).
2) Generation of the upper and lower envelope via cubic spline interpolation among all the maxima and minima, respectively.
3) Point by point averaging of the two envelopes to compute a local mean series m(t).
4) Subtraction of m(t) from the data to obtain a IMF candidate d(t)=x(t)-m(t).
5) Check the properties of d(t):
   - If d is not a IMF (i.e it does not satisfy the previously defined properties), replace x(t) with d(t) and repeat the procedure from step 1
   - If d is a IMF, evaluate the residue m(t)=x(t)-d(t)

Repeat the procedure from step 1 to step 5 by sifting the residual signal.

The sifting process ends when the residue satisfies a predefined stopping criterion.

By construction, the number of extrema is decreased when going from one residual to the next (thus guaranteeing that the complete decomposition is achieved in a finite number of steps), and the corresponding spectral supports are expected to decrease accordingly. Selection of modes rather corresponds to an automatic and adaptative (signal dependent) time variant filtering [9] and [10].

At the end of the algorithm, we have:

$$x(t) = \sum_{i=1}^{n} d_i(t) + m_n(t) \quad (1)$$

where $m_n(t)$ is the residue and d_i is the intrinsic mode function at mode i that has the same numbers of zero crossing and extrema; and is symmetric with respect to the local mean.

Another way to explain how the empirical mode decomposition works is that it picks out the highest frequency oscillation that remains in the signal. Thus, locally, each IMF contains lower frequency oscillations than the one extracted just before. This property can be very useful to pick up frequency changes, since a change will appear even more clearly at the level of a IMF [5].

Figure 1 shows the starting point of signal decomposition and the IMF candidate obtained after little iteration.

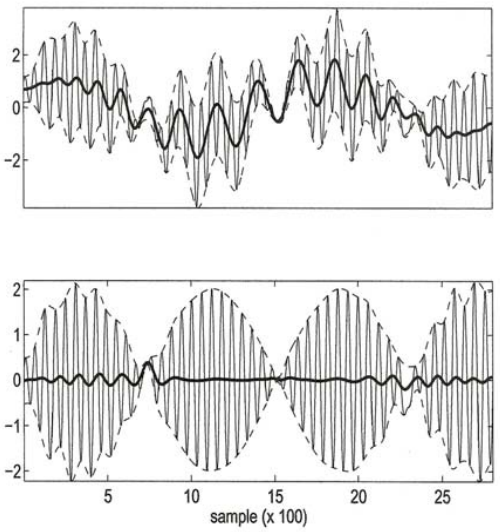


Figure 1: At the top: the original signal with upper and lower envelope. The thick line represents the point by point mean value of the envelopes. Below: the signal d after little iteration. The iteration continue until becomes IMF.

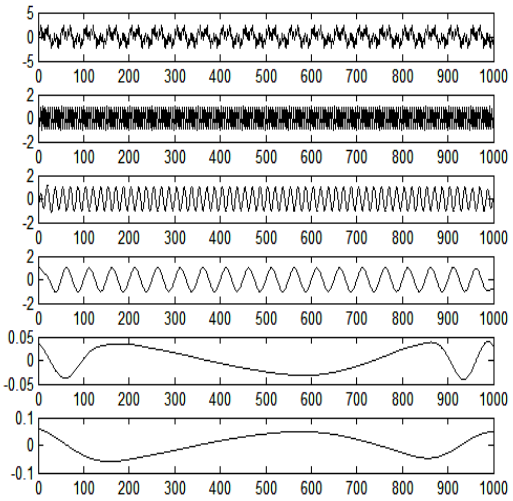


Figure 2: Decomposition of sum of 3 sinus signal of frequency 100Hz, 300 Hz and 1000 Hz by 5 first IMFs.

Figure 2, shows a signal which is the sum of three pure frequencies having the following frequencies: 100 Hz, 300 Hz and 1000 Hz, and the five IMFs followed by the residue. The signal is composed by 1000 samples with a sampling frequency of 20 kHz.

We can see that each component has the same number of zero crossings as extrema and is symmetric with respect to zero line. We note that the first mode which corresponds naturally to the highest frequency shows clearly the 1 kHz frequency present in the signal. Consequently the second mode depicts 300Hz frequency and the third one corresponds to the lowest frequency i.e. 100Hz.

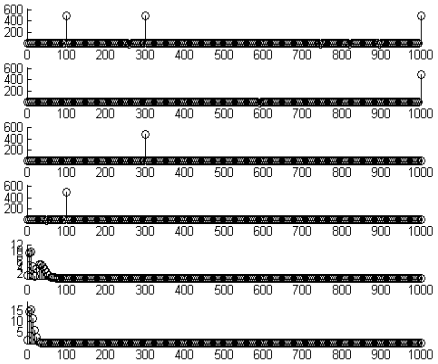


Figure 3: Spectral analysis of composite signal (frequency 100Hz, 300 Hz and 1000 Hz) and its 5 first IMFs.

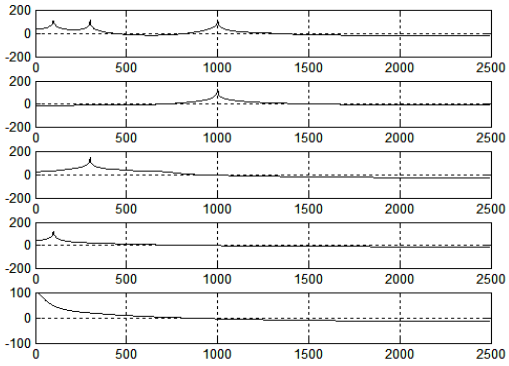


Figure 4: LPC analysis of composite signal and its IMFs

Fourier analysis of the composite signal and its IMFs as depicted in Figure 3, shows that the highest frequency is identified from the first IMF and the lowest one is given by the third IMF.

We compute also the LPC analysis of signal and the 3 IMFs using autocorrelation method. As expected, LPC analysis shows peaks at 1 kHz, 300 Hz and 100 Hz.

This analysis demonstrates once again the efficiency of the proposed method in decomposing the signal in spectral domain [12]. The proposed decomposition detects all frequencies constituting the signal separetly.

The EMD procedure, according to the above specifications, is used in the next section for the decomposition of speech signal issued from Keele database as described in the next section.

3. EMD analysis of voiced speech signal

In the previous section we illustrate the efficiency of the empirical mode decomposition of a typical signal which is the sum of pure frequencies in detecting these frequencies. This approach is used to decompose the speech signal in order to analyze its formant frequencies.

We take as an example of speech signal, a vowel /o/ pronounced by a female speaker f1, extracted from the Keele University database and sampled at 20 kHz. Figure 5 shows the different modes obtained from the empirical mode decomposition of the vowel /o/ and the residue of the last algorithm step.

In our approach, we proceed to an LPC analysis of the IMFs represented in figure 5 and its comparison to results of the same analysis operated on speech signal. The results are depicted in figures 6 and 7.

The LPC analysis achieved for the first IMF shows a curve that fits approximately curve corresponding to speech signal but the peaks for IMF are sharper. The first analyzed IMF doesn't depict the low frequency composition of the signal. In fact it concerns the highest frequency.

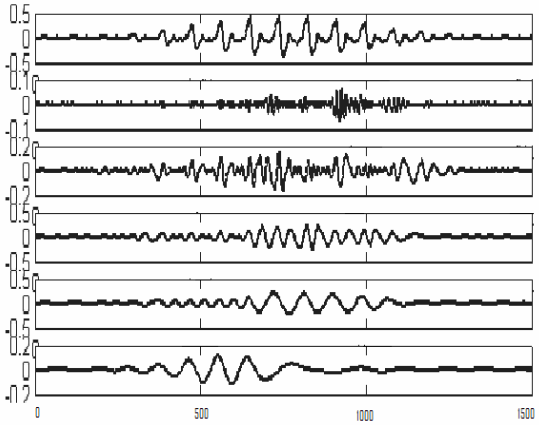


Figure 5: Illustration of the EMD: vowel /o/ speaker f1 and first five IMFs

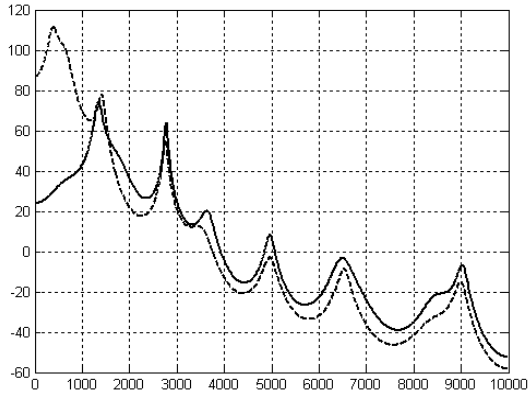


Figure 6: LPC analysis of vowel /o/ speaker f1 (dashed line) and of the first signal's IMF (solid line).

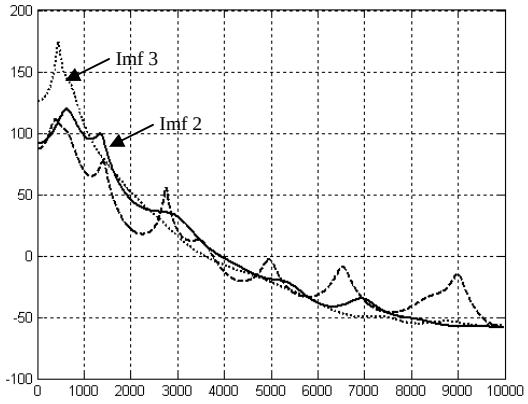


Figure 7: LPC analysis of vowel /o/ speaker f1 and IMFs of the signal (second and third).

This result, mode by mode, in a frequency profile can be interpreted as the frequency response of some equivalent filter. As evidenced in figures 6 and 7, the collection of all such filters tend to estimate the different resonant frequencies of the vocal tract.

An other example for speech signal is given to emphasize the efficiency of this method. It's about a vowel /a/ expressed by a male speaker m2. The achieved EMD is depicted in figure 8.

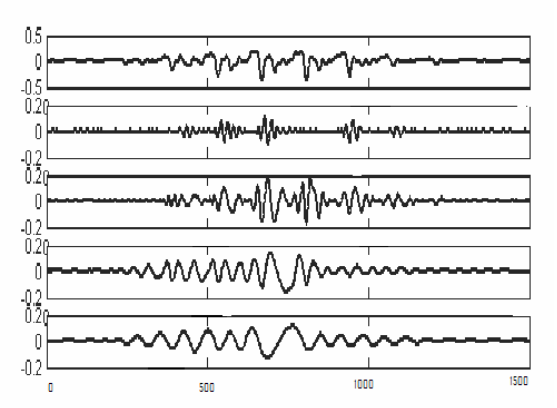


Figure 8: Illustration of the EMD: vowel /a/ speaker m2 and the different IMFs.

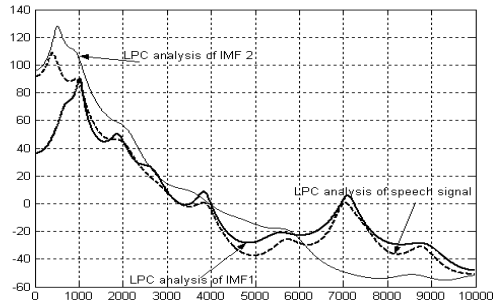


Figure 9: LPC analysis vowel /a/ speaker m2 and the two first IMFs. Solid line, concerns LPC analysis of the first IMF, the dashed line represents vowel /a/ of speaker m2, and the third curve concerns the second IMF.

Figure 9 depicts the LPC analysis of the corresponding speech signal and its 3 first IMFs. We note that peaks given by IMFs are more distinguishable than those releated to speech.

4. Conclusion

In this work, we have proposed a new methodology to decompose a speech signal into different oscillatory modes and to extract the resonant frequencies of the vocal tract i.e. formants from the LPC analysis of different intrinsic mode functions called IMFs. LPC analysis of IMFs shows the frequency components. If we represent all the LPC analysis, we may obtain a complete description of the speech production model. A, study of the residue can be considered and compared to the frequency representation of the glottal pulse.

Besides, we can look for a new time-frequency attributes obtained from the EMD analysis and based on an instantaneous frequency calculation of each component of the decomposition.

5. References

[1] P. Flandrin, Temps-fréquence, Hermes, 1993.

[2] L. Cohen, Time-frequency analysis, Prentice Hall, 1995.

[3] I. Daubechies, Ten lectures on wavelet, SIAM, 1992.

[4] N. E. Huang, Z. Shen, S. R. Long, M. L. Wu, H. H. Shih, Q. Zheng, N. C. Yen, C. C. Tung and H. H. Liu, The empirical mode decomposition and Hilbert spectrum for nonlinear and non-stationary time series analysis, Proc. Roy. Soc. London A, Vol. 454, pp. 903-995,1998.

[5] I. M. Chagnollean and R. G. Baranniuk, Empirical mode decomposition based time-frequency attributes, 69th SEG Meeting, Houston, United States, November 1999.

[6] R. Fournier, Analyse stochastique modale du signal stabilométrique. Application à l'étude de l'équilibre chez l'Homme, Thèse de Doctorat, Univ. Paris XII Val de Marne, 2002.

[7] E. P. Sousa et al., Assessment of cardiovascular autonomic control by the empirical mode decomposition , 4th Int. Workshop on Biosignal Interpretation, Como (I), pp. 123-126, 2002.

[8] R. Balocchi, D. Menicucci, E. Santarcangelo, L. Sebastiani, A. Gemignani, B. Ghelarducci, and M. Varanini, Deriving the respiratory sinus arrhythmia from the heartbeat time series using empirical mode Decomposition, Qauntitative Biology, October 2003.

[9] P. Flandrin et P. Gonçalvès, sur la decomposition modale empirique, GRETSI 2003, Paris, Semptembre 2003.

[10] G. Rilling, P. Flandrin and P. Gonçalvès, On empirical mode decomposition and its algorithms, IEEE-EURASIP Workshop on nonlinear signal and image processing NSIP-03, Grado(I), 2003.

[11] J. C. Nunes, O. Niang, Y. Bouaoune, E. Deléchelle and Ph. Bunel, Décomposition empirique multimodale bidimensionnelle modifiée pour l'analyse d'images, GRETSI 2003, Paris, Septembre 2003.

[12] P. Flandrin, G. Rilling and P. Gonçalvès, Empirical mode decomposition as a filter bank, IEEE signal processing letters, Vol.11, No.2: pp. 112-114.

Estimation of Speech Features of Glottal Excitation by Nonlinear Prediction

Karl Schnell, Arild Lacroix

Institute of Applied Physics, Goethe-University Frankfurt
Max-von-Laue-Str. 1, D-60438 Frankfurt am Main, Germany
schnell@iap.uni-frankfurt.de

Abstract

Analysis of speech signals can be performed with the aid of linear or nonlinear statistics using appropriate prediction algorithms. In this contribution, speech features are treated using the results of a nonlinear prediction based on Volterra series. Features are investigated representing the prediction gain by nonlinear statistics and representing individual coefficients of the nonlinear components. The features are estimated quasi continuously resulting in a feature signal. Additionally, to obtain features which are highly sensitive to segmentation shifting, an asymmetric window function is integrated into the prediction algorithm. The analyses of speech signals show that the estimated features correlate with the glottal pulses. Furthermore, the investigations show that using the first individual nonlinear coefficient as a feature is advantageous over using the prediction gain.

1. Introduction

Speech analysis is usually performed using linear models and statistics. However, nonlinear components are also contained in the speech signal [1]. The voiced excitation is caused by vibrations of the vocal folds which can be described by a nonlinear oscillator; additionally nonlinear fluid dynamics are effective. Nonlinear systems and operators, like the energy operator, can be used for speech analysis [2],[3]. In this contribution nonlinear components of the speech signal are estimated by nonlinear prediction. The nonlinear system of a Volterra series is used for the prediction. The estimation can be achieved by an adaptive algorithm like LMS or RLS [4]. Another approach for the estimation is to minimize the prediction error of individual signal segments, which can be applied to coding [5] or speech generation [6]. For speech analysis the integration of an appropriate window function can be relevant [7]. In [7] speech features based on the prediction gain are discussed. In this contribution, features of nonlinear coefficients of the predictor are proposed delivering feature signals advantageously for analysis. Additionally, a post-processing of the feature signal is carried out accentuating the regions of glottal closures.

2. Nonlinear Prediction

The nonlinear predictor based on Volterra systems estimates a signal value $x(n)$ by a linear combination of last signal values $x(n-k)$ and, additionally, by a linear combination of products of last signal values. Here, without loss of generality systems are treated with the first and second order Volterra kernels only, leading to the prediction error $e = x(n) - \hat{x}(n)$:

$$= x(n) - \sum_{k=1}^{N} h_1(k) \cdot x(n-k) - \sum_{i=1}^{M} \sum_{k=1}^{i} h'_2(i,k) \cdot x(n-i)x(n-k) . \quad (1)$$

e is the prediction error and $\hat{x}(n)$ is the estimation of $x(n)$. The coefficients h_1 represent the linear components whereas h'_2 represent the nonlinear components; h'_2 are coefficients of the second-order kernel h_2, which can be assumed symmetrically $h'_2(i,k) = h_2(i,k)$ for $i = k$ and $h'_2(i,k) = 2 \cdot h_2(i,k)$ for $i \neq k$. For speech analysis the speech signal is segmented in frames. Due to the segmentation a window function $w(n)$ is integrated into the estimation of the nonlinear prediction. If the window function is applied directly to the signal $x(n)$ the prediction error results in

$$e = w(n)x(n) - \sum_{k=1}^{N} h_1(k) \cdot \underline{w(n-k)}x(n-k) - \sum_{i=1}^{M} \sum_{k=1}^{i} h'_2(i,k) \cdot \underline{w(n-i)w(n-k)}x(n-i)x(n-k)$$

leading to different weights of the components, especially between the linear and nonlinear components. For this reason the window function has to be applied to the error $e(n)$ yielding the weighted error $e_w(n) = w(n) \cdot e(n)$. Applying to eq. (1) results in

$$e_w(n) = w(n) \cdot x(n) - \sum_{k=1}^{N} h_1(k) \cdot w(n) \cdot x(n-k) - \sum_{i=1}^{M} \sum_{k=1}^{i} h'_2(i,k) \cdot w(n) \cdot x(n-i)x(n-k) . \quad (2)$$

The predictor coefficients are determined by minimizing the weighted error

$$\sum_n e_w(n)^2 \rightarrow \min , \quad (3)$$

which is explained in the following section.

2.1. Vector based nonlinear prediction

The prediction is applied to a segment of the speech signal, so that it is convenient to describe the signals by vectors. For

that purpose the analyzed weighted signal $u(n) = w(n) \cdot x(n)$ is described by the vector

$$\boldsymbol{u} = \left(w(0) \cdot x(0), w(1) \cdot x(1), \ldots, w(K) \cdot x(K)\right)^{\mathrm{T}}$$

of length $L = K + 1$. Since the prediction error $e_w(n)$ contains last values $x(n-k)$, additionally the vectors $\boldsymbol{u}_i$ and $\boldsymbol{u}_{i,k}$ containing the shifted signals with fixed weights are defined by

$$\begin{aligned} \boldsymbol{u}_i &= (w(0) \cdot x(-i), w(1) \cdot x(1-i), \ldots, w(K) \cdot x(K-i))^{\mathrm{T}} \\ \boldsymbol{u}_{i,k} &= \left(w(0)x(-i)x(-k),\ w(1)x(1-i)x(1-k), \ldots\right)^{\mathrm{T}}. \end{aligned} \quad (4)$$

The estimation of the weighted signal values $u(n)$ can be described by the vector $\hat{\boldsymbol{u}}$ with

$$\hat{\boldsymbol{u}} = \sum_{i=1}^{N} h_1(i) \cdot \boldsymbol{u}_i + \sum_{i=1}^{M} \sum_{k=1}^{i} h'_2(i,k) \cdot \boldsymbol{u}_{i,k} \quad . \quad (5)$$

By these definitions the prediction problem can be described by the vector equation $\boldsymbol{e}_w = \boldsymbol{u} - \hat{\boldsymbol{u}}$. Since the error depends on the order N of linear coefficients and order M of nonlinear coefficients, the error $\boldsymbol{e}_w \rightarrow \boldsymbol{e}_w^{N,M}$ is extended by the superscripts N and M:

$$\boldsymbol{e}_w^{N,M} = \boldsymbol{u} - \sum_{i=1}^{N} h_1(i) \cdot \boldsymbol{u}_i - \sum_{i=1}^{M} \sum_{k=1}^{i} h'_2(i,k) \cdot \boldsymbol{u}_{i,k} \ , \quad (6)$$

respectively

$$\boldsymbol{e}_w^{N,M} = \begin{pmatrix} e_w^{N,M}(0) \\ e_w^{N,M}(1) \\ \vdots \\ e_w^{N,M}(K) \end{pmatrix} = \begin{pmatrix} w(0)x(0) \\ w(1)x(1) \\ \vdots \\ w(K)x(K) \end{pmatrix} - h_1(1) \begin{pmatrix} w(0)x(-1) \\ w(1)x(0) \\ \vdots \\ w(K)x(K-1) \end{pmatrix} \cdots$$

$$- h'_2(1,1) \begin{pmatrix} w(0)x^2(-1) \\ w(1)x^2(0) \\ \vdots \\ w(K)x^2(K-1) \end{pmatrix} - h'_2(1,2) \begin{pmatrix} w(0)x(-1)x(-2) \\ w(1)x(0)x(-1) \\ \vdots \\ w(K)x(K-1)x(K-2) \end{pmatrix} \cdots .$$

Equation (6) represents a vector based description of eq. (2). From the equations (5) and (6) it can be seen that the optimal prediction $\hat{\boldsymbol{u}}$ is an expansion of $\boldsymbol{u}$ by the vectors $\boldsymbol{u}_i$ and $\boldsymbol{u}_{i,k}$. For this expansion the vectors $\boldsymbol{u}_i$ and $\boldsymbol{u}_{i,k}$ are transformed into an orthogonal basis $\{\boldsymbol{v}_m\}$ with the dot products $\langle \boldsymbol{v}_m, \boldsymbol{v}_k \rangle = 0$. This is performed by the Gram-Schmidt orthogonalization. Since the vectors of the basis $\{\boldsymbol{v}_m\}$ are orthogonal, the optimal coefficients b_m in description of the basis $\{\boldsymbol{v}_m\}$ can easily be obtained by

$$b_m = \langle \boldsymbol{u}, \boldsymbol{v}_m \rangle / |\boldsymbol{v}_m|^2 \ ,$$

yielding an expansion with the vectors $\boldsymbol{v}_m$. Finally the coefficients b_m of basis $\{\boldsymbol{v}_m\}$ are converted back into the original basis of $\{\boldsymbol{u}_i, \boldsymbol{u}_{i,k}\}$. The resulting coefficients minimize the Euclidean norm $|\boldsymbol{e}_w^{N,M}|$ representing a least square estimation.

Since in eqs. (2), (6) signal values outside of the frame appear, represented by negative arguments of $n-k$, the vector lengths are truncated in such a way that only values inside of the analyzed segment appear in the vectors.

3. Speech Features

The results of the nonlinear prediction can be utilized to define speech features. One approach is to consider the prediction gain by the nonlinear components. The gain can be described by the ratio between the prediction errors with and without nonlinear components, which is used in [7]. The logarithmic error ratio leads to the feature definition:

$$F_{\mathrm{gain}}^{N,M} = \log\left(\frac{\left|\boldsymbol{e}_w^{N,0}\right|}{\left|\boldsymbol{e}_w^{N,M}\right|}\right) \ .$$

Nonlinear coefficients $h'_2(i,k)$ are considered for the prediction error of the denominator, which can be seen from the superscript M. Since the nonlinear coefficients contribute only to a decrease of the prediction error, the feature $F_{\mathrm{gain}}^{N,M}$ has positive values.

Another approach for defining features is to consider individual values of the estimated predictor coefficients, especially these of the nonlinear components. Here the value of the nonlinear coefficients $h'_2(i,k)$ of the prediction of orders N and M are used leading to the feature

$$F_{i,k}^{N,M} = h'_2(i,k) \ .$$

3.1. Feature signals

The feature F is obtained from the results of the nonlinear prediction. To consider the time-dependence of the feature, the speech signal is segmented into overlapping frames analyzed individually. Applying the nonlinear prediction to each segment yields the corresponding values of the speech feature F. To measure the features quasi continuously in time, the displacement of the segments is chosen to one sample. Hence, the sequence of the feature values which are estimated from the segments represents a feature signal $F(n)$. Each value $F(n)$ is obtained from the nonlinear prediction of a segment. The estimation is influenced by the type of the window function w of the prediction algorithm. If a Hann-window is used, the feature F can be estimated smoothly in time, however, the time resolution of the feature estimation is blurred. This behaviour is caused by the shape of the window; the Hann window is insensitive to small changes of the segmentation since its values tend continuously towards zero to the left and right side. In contrast to that, an asymmetric window with a value greater one at one side is sensitive to small changes of the segmentation and can deliver a more precise time resolution. The window w_a defined by

$$\begin{aligned} w_a(k) &= 1 \quad \text{for} \quad k = 0 \ldots K/4 - 1 \\ w_a(k) &= \left(0.5\left(1 + \cos\left(\frac{\pi(k - K/4)}{K - K/4}\right)\right)\right)^2 \quad \text{for} \quad k = \frac{K}{4} \ldots K \end{aligned}$$

delivering a strong discontinuity between the left-side values and values outside of the window, which can be assumed as zero. The asymmetric window is shown in fig. 1.

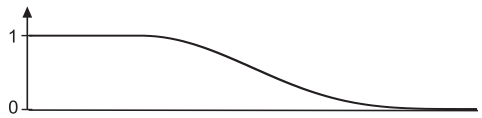


Figure 1: Asymmetric window function w_a .

4. Analysis of Speech

For the analysis of individual sounds and speech utterances, speech signals with a sampling rate of 16 kHz are investigated. The speech signals are segmented and analyzed as described in the previous section. Fig. 2 shows the estimated feature signals $F_{\text{gain}}^{16,1}(n)$ and $F_{1,1}^{16,1}(n)$ from the analysis of the vowel /a/; additionally, the original speech waveform and the LPC-residual is shown. The main impulses of the residual of fig. 2(b) indicate the abrupt glottal closures, which are denoted as the glottal closure instances (CGI). It can be seen that the feature signal has peaks correlating with those of the residual. Hence, the peaks of the feature signals indicate the CGI. The high time resolution of the feature signal results from the asymmetric window. In the case of voiced fricatives often many impulses occur in the residual, which can be seen from fig. 3 showing the analysis of the voiced fricative /z/. The LPC-residual shows a more or less unperiodic structure and especially the high incidence of pulses makes it hard to detect the impulses corresponding to the glottal closures. In contrast to that, the feature signals are more periodic and have fewer pulses. The analyses show that the feature signal $F_{1,1}^{16,1}(n)$ shows even mostly only one dominant positive pulse per period corresponding to the glottal closure; therefore the feature $F_{1,1}^{16,1}(n)$ is advantageous in comparison to the prediction gain delivering often more potential pulses per period. One reason for that is given in the following: For the prediction with order $M=1$ only the nonlinear coefficient $h'_2(1,1)$ is effective. The prediction gain depends on the absolute value of the coefficient, whereas the feature $F_{1,1}^{16,1}(n)$ preserves the information about the sign of the coefficient $h'_2(1,1)$. Analysis results show that the glottal closures cause impulses with positive sign. Other regions of the feature signal show also impulses or bulges, however, for $F_{1,1}^{16,1}(n)$ they have usually negative sign. Hence, by the feature signal $F_{1,1}^{16,1}(n)$ the impulses of glottal closure can be separated from the other regions with the aid of the sign of $h'_2(1,1)$. In comparison to that the feature $F_{\text{gain}}^{16,1}(n)$ cannot distinguish since the information of the sign is lost in the prediction gain.

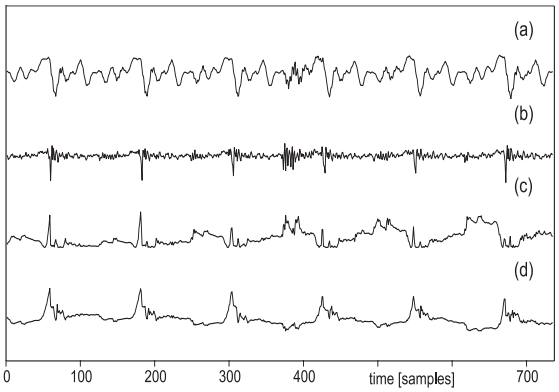


Figure 2: Analysis of the vowel /a/: (a) analyzed speech signal, (b) corresponding LPC-residual, (c) feature signal of prediction gain $F_{\text{gain}}^{16,1}(n)$, (d) feature signal $F_{1,1}^{16,1}(n)$.

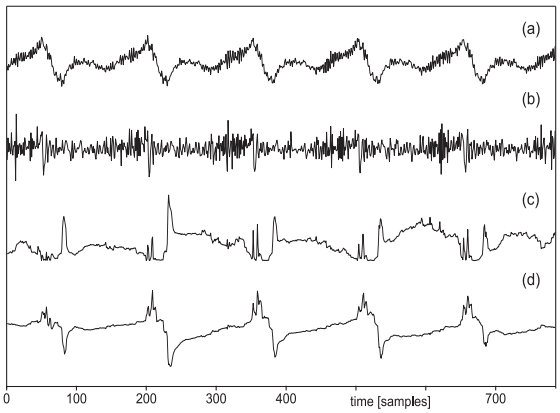


Figure 3: Analysis of the voiced fricative /z/: (a) analyzed speech, (b) corresponding LPC-residual, (c) feature signal of prediction gain $F_{\text{gain}}^{16,1}(n)$, (d) feature signal $F_{1,1}^{16,1}(n)$.

Overall, the analyses show that especially the feature signal $F_{1,1}^{16,1}(n)$ is suitable for detection of regions of glottal closures not only for stationary speech signals, but also for speech utterances. A post-processing of the feature signal is useful to mark the regions of glottal closures. At first, fluctuations of the mean value differing from zero should be compensated; additionally, the power of the feature signal should be balanced achieving a constant envelope of the amplitude. Therefor a short-time estimation of the mean of the feature signal is subtracted to each feature value. After that, each feature value is divided by a short-time estimation of the power of the feature signal resulting in the modified feature signal $\tilde{F}_{1,1}^{16,1}(n)$.

Figure 4 shows the analysis of the German word [nUl]. The curves 4(c) and (d) show the initial feature and the modified feature signal $\tilde{F}_{1,1}^{16,1}(n)$; variations of the mean and the power of the feature signal are balanced. After that the modified

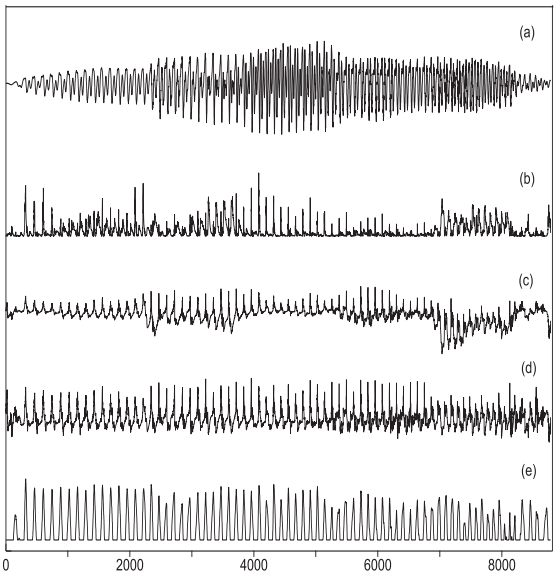


Figure 4: Analysis of word [nUl]: (a) analyzed speech signal, (b) feature signal of prediction gain $F_{\text{gain}}^{16,1}(n)$, (c) feature signal $F_{1,1}^{16,1}(n)$, (d) processed feature $\tilde{F}_{1,1}^{16,1}(n)$, (e) derived feature signal $f'(n)$.

feature signal is convolved by a finite pattern-signal g depicted in fig. 5 which has a pointed shape resulting in the signal

$$f(n) = \tilde{F}_{1,1}^{16,1}(n) * g(n) ;$$

the mean value of the signal g is zero. The convolution implies dot products with time-shifted segments. If the segment is similar to the pointed shape, a high value results.

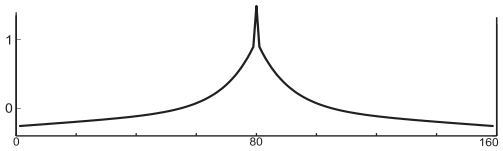


Figure 5: Pattern-signal g.

Since only positive correlations with the pointed shape are of interest, negative values of f are set to zero by

$$f'(n) = (f(n) + \text{sgn}(f(n)) \cdot f(n)) / 2 .$$

The curve 4(e) shows the derived feature signal $f'(n)$ representing the positive values of the convolution results of the modified feature signal of the utterance [nUl]. The peaks indicate regions of glottal pulses.

In figure 6 the analysis result for the utterance [vaIma] of the German word “Weimar” is shown. It can be seen that the corresponding feature signal $f'(n)$ represents a sequence of pulses, which is disturbed only occasionally by artifacts.

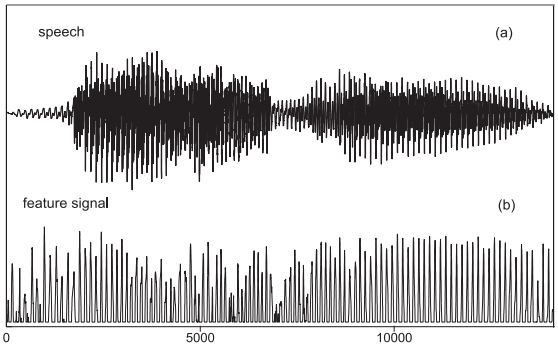


Figure 6: Analysis of word [vaIma]: (a) analyzed speech signal, (b) derived feature signal $f'(n)$.

5. Conclusions

Speech features based on nonlinear prediction are proposed and discussed for the analysis of speech. The features are correlated with the voiced excitation and especially with the glottal pulses. For analysis of real speech, one important feature of estimation algorithms is their robustness. Concerning this, features based on the first nonlinear prediction coefficient have been proven advantageously in comparison to the prediction gain. The deciding reason for this fact is that the informational content of the sign of the coefficient is useful. By the use of that feature signal with an additional post-processing the algorithm is applicable to analyse real speech.

6. References

[1] M. Faundez et al., "Nonlinear Speech Processing: Overview and Applications", in *Int. J. Control Intelligent Syst., vol. 30, no. 1*, pp. 1–10, 2002.

[2] P. Maragos, T. Quatieri, and J. Kaiser, "Speech Nonlinearities, Modulations, and Energy Operators", in *Proc. ICASSP'91*, 1991, pp. 421-424.

[3] L. Atlas and J. Fang, "Quadratic Detectors for General Nonlinear Analysis of Speech", in *Proc. ICASSP'92, vol. II*, 1992, pp. 9–12.

[4] E. Mumolu, A. Carini, and D. Francescato, "ADPCM With Non Linear Predictors", in *Proc. EUSIPCO'94*, 1994, pp. 387–390.

[5] J. Thyssen, H. Nielsen, and S. D. Hansen, "Non-linear Short-term Prediction in Speech Coding", in *Proc. ICASSP'94, vol. I*, 1994 pp. 185–188.

[6] K. Schnell and A. Lacroix, "Modeling Fluctuations of Voiced Excitation for Speech Generation Based on Recursive Volterra Systems", contribution in *Nonlinear Analyses and Algorithms for Speech Processing – NOLISP'05, LNAI Vol. 3817*, pp. 338-347, Springer 2005.

[7] K. Schnell and A. Lacroix, "Weighted Nonlinear Prediction Based on Volterra Series for Speech Analysis", in *Proc. EUSIPCO'06*, Florence 2006.

An efficient VAD based on a Generalized Gaussian PDF

O. Pernía, J.M. Górriz, J. Ramírez and C.G. Puntonet and I. Turias

Department of Signal Theory
University of Granada, Granada, Spain
gorriz@ugr.es

Abstract

The emerging applications of wireless speech communication are demanding increasing levels of performance in noise adverse environments together with the design of high response rate speech processing systems. This is a serious obstacle to meet the demands of modern applications and therefore these systems often needs a noise reduction algorithm working in combination with a precise voice activity detector (VAD). This paper presents a new voice activity detector (VAD) for improving speech detection robustness in noisy environments and the performance of speech recognition systems. The algorithm defines an optimum likelihood ratio test (LRT) involving Multiple and correlated Observations (MCO). An analysis of the methodology for $N = \{2, 3\}$ shows the robustness of the proposed approach by means of a clear reduction of the classification error as the number of observations is increased. The algorithm is also compared to different VAD methods including the G.729, AMR and AFE standards, as well as recently reported algorithms showing a sustained advantage in speech/non-speech detection accuracy and speech recognition performance.

1. Introduction

The emerging applications of speech communication are demanding increasing levels of performance in noise adverse environments. Examples of such systems are the new voice services including discontinuous speech transmission [1, 2, 3] or distributed speech recognition (DSR) over wireless and IP networks [4]. These systems often require a noise reduction scheme working in combination with a precise voice activity detector (VAD) [5] for estimating the noise spectrum during non-speech periods in order to compensate its harmful effect on the speech signal.

During the last decade numerous researchers have studied different strategies for detecting speech in noise and the influence of the VAD on the performance of speech processing systems [5]. Sohn *et al*. [6] proposed a robust VAD algorithm based on a statistical likelihood ratio test (LRT) involving a single observation vector. Later, Cho *et al* [7] suggested an improvement based on a smoothed LRT. Most VADs in use today normally consider hangover algorithms based on empirical models to smooth the VAD decision. It has been shown recently [8, 9] that incorporating long-term speech information to the decision rule reports benefits for speech/pause discrimination in high noise environments, however an important assumption made on these previous works has to be revised: *the independence of overlapped observations*. In this work we propose a more realistic one: *the observations are jointly gaussian distributed with non-zero correlations*. In addition, important issues that need to be addressed are: *i*) the increased computational complexity mainly due to the definition of the decision rule over large data sets, and *ii*) the optimum criterion of the decision rule. This work advances in the field by defining a decision rule based on an optimum statistical LRT which involves multiple and *correlated* observations. The paper is organized as follows. Section 2 reviews the theoretical background on the LRT statistical decision theory. Section 4 considers its application to the problem of detecting speech in a noisy signal. Finally in Section 4.1 we discuss the suitability of the proposed approach for pair-wise correlated observations using the experimental data set AURORA 3 subset of the original Spanish SpeechDat-Car (SDC) database [10] and state some conclusions in section 6.

2. Multiple Observation Probability Ratio Test

Under a two hypothesis test, the optimal decision rule that minimizes the error probability is the Bayes classifier. Given an observation vector $\hat{\mathbf{y}}$ to be classified, the problem is reduced to selecting the hypothesis (H_0 or H_1) with the largest posterior probability P($H_i|\hat{\mathbf{y}}$). From the Bayes rule:

$$L(\hat{\mathbf{y}}) = \frac{p_{\mathbf{y}|H_1}(\hat{\mathbf{y}}|H_1)}{p_{\mathbf{y}|H_0}(\hat{\mathbf{y}}|H_0)} \begin{array}{c} > \\ < \end{array} \frac{P[H_0]}{P[H_1]} \Rightarrow \begin{array}{l} \hat{\mathbf{y}} \leftrightarrow H_1 \\ \hat{\mathbf{y}} \leftrightarrow H_0 \end{array} \quad (1)$$

In the LRT, it is assumed that the number of observations is fixed and represented by a vector $\hat{\mathbf{y}}$. The performance of the decision procedure can be improved by incorporating more observations to the statistical test. When N measurements $\hat{\mathbf{y}}_1$, $\hat{\mathbf{y}}_2$, ..., $\hat{\mathbf{y}}_N$ are available in a two-class classification problem, a multiple observation likelihood ratio test (MO-LRT) can be defined by:

$$L_N(\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, ..., \hat{\mathbf{y}}_N) = \frac{p_{\mathbf{y}_1,\mathbf{y}_2,...,\mathbf{y}_N|H_1}(\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, ..., \hat{\mathbf{y}}_N|H_1)}{p_{\mathbf{y}_1,\mathbf{y}_2,...,\mathbf{y}_N|H_0}(\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, ..., \hat{\mathbf{y}}_N|H_0)} \quad (2)$$

This test involves the evaluation of an *N*-th order LRT which enables a computationally efficient evaluation when the individual measurements $\hat{\mathbf{y}}_k$ are independent. However, they are not since the windows used in the computation of the observation vectors $\mathbf{y}_k$ are usually overlapped. In order to evaluate the proposed MCO-LRT VAD on an incoming signal, an adequate statistical model for the feature vectors in presence and absence of speech needs to be selected. The joint probability distributions under both hypotheses are assumed to be jointly gaussian independently distributed in frequency and in each part (real and imaginary) of vector with correlation components between each pair of frequency observations:

$$L_N(\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2, ..., \hat{\mathbf{y}}_N) = \prod_{p\in\{R,I\}} \{ \prod_{\omega} \frac{p_{y_1^\omega, y_2^\omega, \ldots, y_N^\omega|H_1}(\hat{y}_1^\omega, \hat{y}_2^\omega, \ldots, \hat{y}_N^\omega|H_1)}{p_{y_1^\omega, y_2^\omega, \ldots, y_N^\omega|H_0}(\hat{y}_1^\omega, \hat{y}_2^\omega, \ldots, \hat{y}_N^\omega|H_0)} \}_p \quad (3)$$

This is a more realistic approach that the one presented in [9] taking into account the overlap between adjacent observations. We use following joint gaussian probability density function for each part:

$$p_{\mathbf{y}_\omega|H_s}(\hat{\mathbf{y}}_\omega|H_s)) = K_{H_s,N} \cdot \exp\{-\frac{1}{2}(\hat{\mathbf{y}}_\omega^T (C^N_{\mathbf{y}_\omega,H_s})^{-1}\hat{\mathbf{y}}_\omega)\} \quad (4)$$

for $s = 0, 1$, where $K_{H_s,N} = \frac{1}{(2\pi)^{N/2}|C^N_{\mathbf{y}_\omega,H_s}|^{1/2}}$, $\mathbf{y}_\omega = (\mathrm{y}_1^\omega, \mathrm{y}_2^\omega, \ldots, \mathrm{y}_N^\omega)^T$ is a zero-mean frequency observation vector, $C^N_{\mathbf{y},H_s}$ is the *N*-order covariance matrix of the observation vector under hypothesis H_s and $|.|$ denotes determinant of a matrix. The model selected for the observation vector is similar to that used by Sohn *et al.* [6] that assumes the discrete Fourier transform (DFT) coefficients of the clean speech (S_j) and the noise (N_j) to be asymptotically independent Gaussian random variables. In our case the observation vector consist of the real and imaginary parts of frequency DFT coefficient at frequency ω of the set of m observations.

3. Evaluation of the LRT

In order to evaluate the MCO-LRT, the computation of the inverse matrices and determinants are required. Since the covariances matrices under $H_0 \& H_1$ are assumed to be tridiagonal symmetric matrices[1], the inverses matrices can be computed as the following:

$$[C^{-1}_{\mathbf{y}_\omega}]_{mk} = [\frac{q_k}{p_k} - \frac{q_N}{p_N}]p_m p_k \quad N-1 \geq m \geq k \geq 0 \quad (6)$$

where N is the order of the model and the set of real numbers q_n, p_n $n = 1 \ldots \infty$ satisfies the three-term recursion for $k \geq 1$:

$$0 = r_k(q_{k-1}, p_{k-1}) + \sigma_{k+1}(q_k, p_k) + r_{k+1}(q_{k+1}, p_{k+1}) \quad (7)$$

with initial values:

$$\begin{array}{ll} p_0 = 1 & \text{and } p_1 = -\frac{\sigma_1}{r_1} \\ q_0 = 0 & \text{and } q_1 = \frac{1}{r_1} \end{array} \quad (8)$$

In general this set of coefficients are defined in terms of orthogonal complex polynomials which satisfy a Wronskian-like relation [11] and have the continued-fraction representation[12]:

$$\frac{q_n(z)}{p_n(z)} = \frac{1}{(z-\sigma_1)-} \ominus \frac{r_1^2}{(z-\sigma_2)-} \ominus \ldots \ominus \frac{r_{n-1}^2}{(z-\sigma_n)} \quad (9)$$

where $\ominus$ denotes the continuos fraction. This representation is used to compute the coefficients of the inverse matrices evaluated on $z = 0$. In the next section we show a new VAD based on this methodology for $N = 2$ and 3, that is, this robust speech detector is intended for real time applications such us mobile communications. The decision function will be described in terms of the correlation and variance coefficients which constitute a correction to the previous LRT method [9] that assumed uncorrelated observation vectors in the MO.

[1]The covariance matrix will be modeled as a tridiagonal matrix, that is, we only consider the correlation function between adjacent observations according to the number of samples (200) and window shift (80) that is usually selected to build the observation vector. This approach reduces the computational effort achieved by the algorithm with additional benefits from the symmetric tridiagonal matrix properties:

$$[C^N_{\mathbf{y}_\omega}]_{mk} = \left[\begin{array}{lll} \sigma^2_{y_m}(\omega) \equiv E[|y_m^\omega|^2] & if & m = k \\ r_{mk}(\omega) \equiv E[y_m^\omega y_k^{\omega]} & if & k = m+1 \\ 0 \quad other & case & \end{array}\right] \quad (5)$$

where $1 \leq i \leq j \leq N$ and $\sigma^2_{y_i}(\omega)$, $r_{ij}(\omega)$ are the variance and correlation frequency components of the observation vector $\mathbf{y}_\omega$ (denoted for clarity σ_i, r_i) which must be estimated using instantaneous values.

4. Application to voice activity detection

The use of the MO-LRT for voice activity detection is mainly motivated by two factors: *i*) the optimal behaviour of the so defined decision rule, and *ii*) a multiple observation vector for classification defines a reduced variance LRT reporting clear improvements in robustness against the acoustic noise present in the environment. The proposed MO-LRT VAD is described as follows. The MO-LRT is defined over the observation vectors $\{\hat{\mathbf{y}}_{l-m}, \ldots, \hat{\mathbf{y}}_{l-1}, \hat{\mathbf{y}}_l, \hat{\mathbf{y}}_{l+1}, \ldots, \hat{\mathbf{y}}_{l+m}\}$ as follows:

$$\ell_{l,N} = \sum_\omega \frac{1}{2}\left\{ {\mathbf{y}_\omega}^T \Delta_N^\omega \mathbf{y}_\omega + ln\left[\frac{|C^N_{\mathbf{y}_\omega,H_0}|}{|C^N_{\mathbf{y}_\omega,H_1}|}\right]\right\} \quad (10)$$

where $\Delta_N^\omega = (C^N_{\mathbf{y}_\omega,H_0})^{-1} - (C^N_{\mathbf{y}_\omega,H_1})^{-1}$, $N = 2m+1$ is the order of the model, l denotes the frame being classified as speech (H_1) or non-speech (H_0) and $\mathbf{y}_\omega$ is the previously defined frequency observation vector on the sliding window.

4.1. Analysis of JGPDF Voice Activity Detector for $N = 2$

In this section the improvement provided by the proposed methodology is evaluated by studying the most simple case for $N = 2$. In this case, assuming that squared correlations ρ_1^2 under $H_0 \& H_1$ and the correlation coefficients are negligible under H_0 (noise correlation coefficients $\rho_1^n \to 0$) vanish, the LRT can be evaluated according to:

$$\ell_{l,2} = \frac{1}{2}\sum_\omega L_1(\omega) + L_2(\omega) + 2\sqrt{\gamma_1\gamma_2}\left[\frac{\rho_1^s}{\sqrt{(1+\xi_1)(1+\xi_2)}}\right] \quad (11)$$

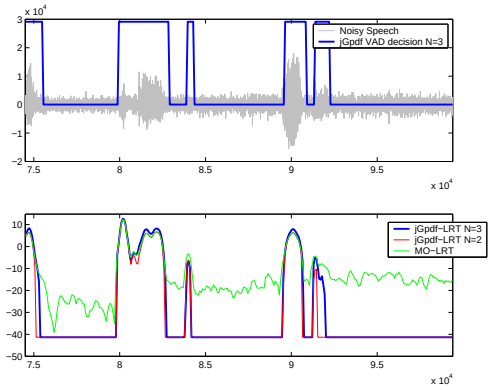
where $\rho_1^s = r_1^s(\omega)/(\sqrt{\sigma_1^s\sigma_2^s})$ is the correlation coefficient of the observations under H_1, $\gamma_i \equiv (y_i^\omega)^2/\sigma_i^n(\omega)$ and $\xi_i \equiv \sigma_i^s(\omega)/\sigma_i^n(\omega)$ are the SNRs a priori and a posteriori of the DFT coefficients, $L_{\{1,2\}}(\omega) \equiv \frac{\gamma_{\{1,2\}}\xi_{\{1,2\}}}{1+\xi_{\{1,2\}}} - \ln(1+\xi_{\{1,2\}})$ are the independent LRT of the observations $\hat{\mathbf{y}}_\mathbf{1}, \hat{\mathbf{y}}_\mathbf{2}$ (connection with the previous MO-LRT [9]) which are corrected with the term depending on ρ_1^s, the new parameter to be modeled, and l indexes to the second observation. At this point frequency ergodicity of the process must be assumed to estimate the new model parameter ρ_1^s. This means that the correlation coefficients are constant in frequency thus an ensemble average can be estimated using the sample mean correlation of the observations $\hat{\mathbf{y}}_\mathbf{1}$ and $\hat{\mathbf{y}}_\mathbf{2}$ included in the sliding window.

4.2. Analysis of JGPDF Voice Activity Detector for $N = 3$

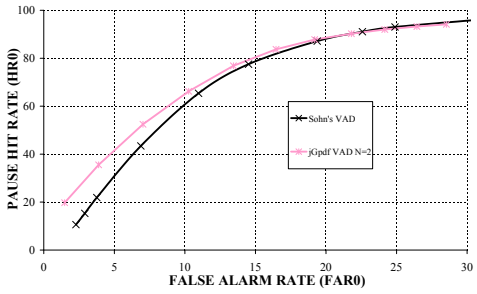
In the case for $N = 3$ the properties of a symmetric and tridiagonal matrix come out. The likelihood ratio can be expressed as:

$$\ell_{l,3} = \sum_\omega \ln\frac{K_{H_1,3}}{K_{H_0,3}} + \frac{1}{2}\hat{\mathbf{y}}_\omega^T \Delta_3^\omega \hat{\mathbf{y}}_\omega \quad (12)$$

where $\ln\frac{K_{H_1,3}}{K_{H_0,3}} = \frac{1}{2}\left[\ln\left[\frac{1-(\rho_1^2+\rho_2^2)^{H_0}}{1-(\rho_1^2+\rho_2^2)^{H_1}}\right] - \ln\ \prod_{i=1}^3(1+\xi_i)\ \right]$, and Δ_3^ω is computed using the following expression under



(a)



(b)

Figure 1: a) JGPDF-VAD vs. MO-LRT decision for $N = 2$ and 3. b) ROC curve for JGPDF VAD with $l_h = 8$ and Sohn's VAD [6] using a similar hang-over mechanism.

hypotheses $H_0 \& H_1$:

$$\hat{\mathbf{y}}_\omega^T (C^3_{\mathbf{y}_\omega, H_s})^{-1} \hat{\mathbf{y}}_\omega = \frac{1}{1-(\rho_1^2+\rho_2^2)} \left[\frac{1-\rho_2^2}{\sigma_1} (y_1^\omega)^2 + \frac{(y_2^\omega)^2}{\sigma_2} \dots \right] \left[+ \frac{1-\rho_1^2}{\sigma_3} (y_3^\omega)^2 - 2\rho_1 \frac{y_1^\omega y_2^\omega}{\sqrt{\sigma_1 \sigma_2}} - 2\rho_2 \frac{y_2^\omega y_3^\omega}{\sqrt{\sigma_2 \sigma_3}} + 2\rho_1 \rho_2 \frac{y_1^\omega y_3^\omega}{\sqrt{\sigma_1 \sigma_3}} \right] \quad (13)$$

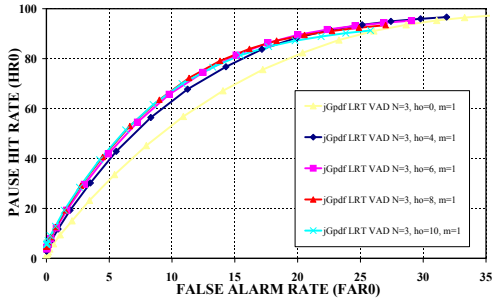
Assuming that squared correlations under $H_0 \& H_1$ and the correlations under H_0 vanish, the log-LRT can be evaluated as the following:

$$\ell_{l,3} = \frac{1}{2} \sum_\omega \sum_{i=1}^3 L_i(\omega) + \frac{2\sqrt{\gamma_1 \gamma_2} \rho_1^s}{\sqrt{(1+\xi_1)(1+\xi_2)}} + \frac{2\sqrt{\gamma_2 \gamma_3} \rho_2^s}{\sqrt{(1+\xi_2)(1+\xi_3)}} - \frac{2\sqrt{\gamma_1 \gamma_3} \rho_1^s \rho_2^s}{\sqrt{(1+\xi_1)(1+\xi_2)^2(1+\xi_3)}} \quad (14)$$

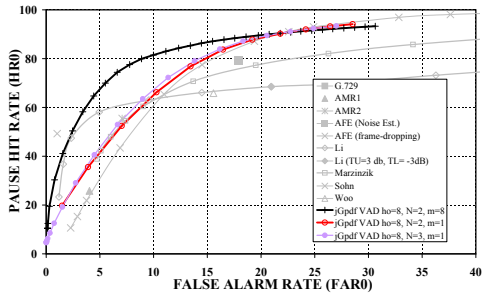
5. Experimental Framework

The ROC curves are frequently used to completely describe the VAD error rate. The AURORA 3 subset of the original Spanish SpeechDat-Car (SDC) database [10] was used in this analysis. The files are categorized into three noisy conditions: quiet, low noisy and highly noisy conditions, which represent different driving conditions with average SNR values between 25dB, and 5dB. The non-speech hit rate (HR0) and the false alarm rate (FAR0= 100-HR1) were determined in each noise condition.

Using the proposed decision functions (equations 14 and 11) we obtain an almost binary decision rule as it is shown in figure 1(a) which accurately detects the beginnings of the voice



(a)



(b)

Figure 2: a) ROC curve analysis of the jGpdf-VAD ($N = 3$) for the selection of the hang-over parameter l_h. b) ROC curves of the jGpdf-VAD using contextual information (eight MO windows for $N = 2$) and standards and recently reported VADs.

periods. In this figure we have used the same level of information in both methods ($m = 1$). The detection of voice endings is improved using a hang-over scheme based on the decision of previous frames. Observe how this strategy cannot be applied to the independent LRT [6] because of its hard decision rule and changing bias as it is shown in the same figure. We implement a very simple hang-over mechanism based on contextual information of the previous frames, thus no delay obstacle is added to the algorithm:

$$\ell^h_{l,N} = \ell_{l,N} + \ell_{l-l_h,N} \quad (15)$$

where the parameter l_h is selected experimentally. The ROC curve analysis for this hang-over parameter is shown in figure 2(a) for $N = 3$ where the influence of hang-over in the zero hit rate is studied with variable detection threshold. Finally, the benefits of contextual information [9] can be incorporated just averaging the decision rule over a set of multiple observations windows (two observations for each window). A typical value for $m = 8$ produces increasing levels of detection accuracy as it is shown in the ROC curve in figure 2(b). Of course, these results are not the optimum ones since only pair-wise dependence is considered here. However for a small number of observations the proposed VAD presents the best trade-off between detection accuracy and delay.

6. Conclusion

This paper showed a new VAD for improving speech detection robustness in noisy environments. The proposed method is developed on the basis of previous proposals that incorporate long-term speech information to the decision rule [9]. However, it is not based on the assumption of independence between observations since this hypothesis is not realistic at all. It defines a statistically optimum likelihood ratio test based on multiple and correlated observation vectors which avoids the need of smoothing the VAD decision, thus reporting significant benefits for speech/pause detection in noisy environments. The algorithm has an optional inherent delay that, for several applications including robust speech recognition, does not represent a serious implementation obstacle. An analysis based on the ROC curves unveiled a clear reduction of the classification error for second and third order model. In this way, the proposed VAD outperformed, at the same conditions, the Sohn's VAD, as well as the standardized G.729, AMR and AFE VADs and other recently reported VAD methods in both speech/non-speech detection performance.

6.1. Computation of the LRT for $N = 2$

From equation 4 for $N = 2$ we have that the MCO-LRT can be expressed as:

$$\ell_{l,2} = \sum_{\omega} \ln \frac{K_{H_1,2}}{K_{H_0,2}} + \frac{1}{2} \hat{\mathbf{y}}_{\omega}^T \Delta_2^{\omega} \hat{\mathbf{y}}_{\omega} \tag{16}$$

where:

$$\ln \frac{K_{H_1,2}}{K_{H_0,2}} = \frac{1}{2} ln \quad \frac{|C^N_{\mathbf{y}_{\omega},H_0}|}{|C^N_{\mathbf{y}_{\omega},H_1}|} \Bigg) = \frac{1}{2} \frac{\sigma_1^{H_0}\sigma_2^{H_0} - (r_1^{H_0})^2}{\sigma_1^{H_1}\sigma_2^{H_1} - (r_1^{H_1})^2} \tag{17}$$

and $C_{\mathbf{y}_{\omega}}$ is defined as in equation 5. If we assume that the voice signal is observed in additive independent noise, that is for $i = 1, 2$:

$$\begin{array}{ll} H_1: & \sigma_i^{H_1} = \sigma_i^n + \sigma_i^s \\ H_0: & \sigma_i^{H_0} = \sigma_i^n \end{array} \tag{18}$$

and define the correlation coefficient $\rho_1^{H_s} \equiv \frac{r_1^{H_1}}{\sqrt{\sigma_1^{H_1}\sigma_2^{H_1}}}$ and the a posteriori SNR $\xi_i \equiv \frac{\sigma_i^s}{\sigma_i^n}$, we have that:

$$\ln \frac{K_{H_1,2}}{K_{H_0,2}} = \frac{1}{2} \left[\ln \quad \frac{1-(\rho_1^{H_0})^2}{1-(\rho_1^{H_1})^2} \quad - \ln \quad \prod_{i=1}^{2}(1+\xi_i) \right) \Bigg] \tag{19}$$

On the other hand, the inverse matrix is expressed in terms of the orthogonal complex polynomials $q_k(z), p_k(z)$ as:

$$(C^2_{\mathbf{y}_{\omega},H_s})^{-1} = \left(\begin{array}{cc} \left[\frac{q_0}{p_0} - \frac{q_2}{p_2}\right] p_0 p_0 & \left[\frac{q_1}{p_1} - \frac{q_2}{p_2}\right] p_0 p_1 \\ \left[\frac{q_1}{p_1} - \frac{q_2}{p_2}\right] p_0 p_1 & \left[\frac{q_1}{p_1} - \frac{q_2}{p_2}\right] p_1 p_1 \end{array} \right)_{H_s} \tag{20}$$

where $p_0 = 1$, $q_0 = 0$, $p_1 = -\sigma_1/r_1$ and $q_2/p_2 = \sigma_2/(r_1^2 - \sigma_1\sigma_2)$ under hypothesis H_s. Thus the second term of equation 16 can be expressed as:

$$\hat{\mathbf{y}}_{\omega}^T \Delta_2^{\omega} \hat{\mathbf{y}}_{\omega} = (y_1^{\omega})^2 (\Delta_2^{\omega})_{00} + (y_2^{\omega})^2 (\Delta_2^{\omega})_{11} + 2 y_1^{\omega} y_2^{\omega} (\Delta_2^{\omega})_{01} \tag{21}$$

where $(\Delta_2^{\omega})_{00} = \frac{\sigma_2^{H_0}}{\sigma_2^{H_0}\sigma_1^{H_0} - (r_1^{H_0})^2} - \frac{\sigma_2^{H_1}}{\sigma_2^{H_1}\sigma_1^{H_1} - (r_1^{H_1})^2}$, $(\Delta_2^{\omega})_{11} = \frac{\sigma_1^{H_0}}{\sigma_2^{H_0}\sigma_1^{H_0} - (r_1^{H_0})^2} - \frac{\sigma_1^{H_1}}{\sigma_2^{H_1}\sigma_1^{H_1} - (r_1^{H_1})^2}$ and $(\Delta_2^{\omega})_{01} = \frac{r_1^{H_0}}{(r_1^{H_0})^2 - \sigma_2^{H_0}\sigma_1^{H_0}} - \frac{r_1^{H_1}}{(r_1^{H_0})^2 - \sigma_2^{H_0}\sigma_1^{H_0}}$. Finally, if we define the a priori SNR $\gamma_i \equiv (y_i^{\omega})^2/\sigma_i^n(\omega)$ and neglect the squared correlation functions under both hypotheses we have equation 11.

7. References

[1] A. Benyassine, E. Shlomot, H. Su, D. Massaloux, C. Lamblin, and J. Petit, "ITU-T Recommendation G.729 Annex B: A silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 64–73, 1997.

[2] ITU, "A silence compression scheme for G.729 optimized for terminals conforming to recommendation V.70," *ITU-T Recommendation G.729-Annex B*, 1996.

[3] ETSI, "Voice activity detector (VAD) for Adaptive Multi-Rate (AMR) speech traffic channels," *ETSI EN 301 708 Recommendation*, 1999.

[4] ——, "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms," *ETSI ES 201 108 Recommendation*, 2002.

[5] R. L. Bouquin-Jeannes and G. Faucon, "Study of a voice activity detector and its influence on a noise reduction system," *Speech Communication*, vol. 16, pp. 245–254, 1995.

[6] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 16, no. 1, pp. 1–3, 1999.

[7] Y. D. Cho, K. Al-Naimi, and A. Kondoz, "Improved voice activity detection based on a smoothed statistical likelihood ratio," in *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, 2001, pp. 737–740.

[8] J. M. Górriz, J. Ramírez, J. C. Segura, and C. G. Puntonet, "An effective cluster-based model for robust speech detection and speech recognition in noisy environments," *Journal of Acoustical Society of America*, vol. 120, no. 470, pp. 470–481, 2006.

[9] J. M. Górriz, J. Ramirez, J. C. Segura, and C. G. Puntonet, "An improved mo-lrt vad based on a bispectra gaussian model," *Electronic Letters*, vol. 41, no. 15, pp. 877–879, 2005.

[10] A. Moreno, L. Borge, D. Christoph, R. Gael, C. Khalid, E. Stephan, and A. Jeffrey, "SpeechDat-Car: A Large Speech Database for Automotive Environments," in *Proceedings of the II LREC Conference*, 2000.

[11] N. Akhiezer, *The Classical Moment Problem*. Edimburgh: Oliver and Boyd, 1965.

[12] H. Yamani and M. Abdelmonem, "The analytic inversion of any finite symmetric tridiagonal matrix," *J. Phys. A: Math Gen*, vol. 30, pp. 2889–2893, 1997.

Index of authors