INCORPORATING KNOWLEDGE ON SEGMENTAL DURATION IN *HMM*-BASED CONTINUOUS SPEECH RECOGNITION

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam, op gezag van de
Rector Magnificus prof. dr J. J. M. Franse ten overstaan
van een door het college van dekanen ingestelde commissie
in het openbaar ter verdedigen in de Aula der Universiteit
op maandag 14 april 1997 te 15.00 uur

door

Xue Wang

geboren te Harbin, China

Preface

Spoken language research, even though originating from the very need of everyday-life, has traditionally been mainly theoretical and descriptive. It is modern technology and present-day sophistication, that finally make these sciences applicable in building useful devices such as talking and listening machines intended for the use by ordinary people and by the handicapped. We all feel privileged that we are able to witness this great development in modern science. I feel especially privileged to be able to see this happening in some detail, thanks to the unique combination of my background as an electronic engineer, and the research interests of the institute that cover speech communication. I am very grateful to the faculty of Arts and to the Institute of the Phonetic Sciences, University of Amsterdam, for accommodating me, and for giving me this very unique opportunity to work at the research project presented in this thesis.

I am greatly indebted to my supervisor Prof. Louis Pols, who guided me through this whole research project. Discussions with him provided me with insight both into the essence of the research in broad outline, and into the many details of the approaches. I thank my tutor Dr. Louis ten Bosch for his many ideas and for providing ways of thinking about how mathematics can be useful for speech technology, a very demanding interdisciplinary area indeed.

I also thank my colleagues Ton Wempe and David Weenink, who helped me in both my work and my life outside the Institute. I would like to express my thanks to Paul Boersma, who made it possible for me to present the most beautiful drawings in this thesis with his program Praat. I thank Dr. Paul van Alphen, for having introduced me to the speech recogniser REXY that he developed. My very warm thanks go to Mrs. Willemien Eelzak, who took care of me in many different ways during my stay at the Institute. I appreciate the discussions with Prof. Shuzhen Wu of Beijing University, China, who also helped me in writing the summary in Chinese.

I thank Dr. Florien Koopmans-van Beinum for having taken care of part of the development of my research and my education, both at the Institute and through the ERASMUS programme. I thank Dr. Valérie Hazan and Dr. Mark Huckvale for being my hosts during my ERASMUS stay at University College London.

I thank Dr. Johan de Veth and Dr. Bert Cranen for helping me with the software used for the tests presented in Chapter 7 of this thesis.

I very much appreciate the useful comments from all members of the thesis committee: Prof. Loe Boves of Nijmegen University, Prof. Yves Kamp of Leuven University, Prof. Sieb Nooteboom of Utrecht University, and Prof. Remko Scha and Dr. Rob van Son of the University of Amsterdam. I am also

grateful to Loe Boves and Yves Kamp for their participation in the discussions at the early stages of my research project.

My last (but not least) thanks go to Jing, Wendy and Ling, for the love, care and joy they gave me, during the past years when I couldn't give them much in return.

I would like to offer my sincere apologies to all the readers of this thesis, who may be troubled in any respects with my writing. But I still dare to hope that some of you will enjoy reading (parts of) this book.

Xue Wang
Purmerend, North Holland
February, 1997

Table of Contents

To my parents

献给我的父母

1

GENERAL INTRODUCTION

Abstract

This chapter gives a short description of the state of the art in automatic speech recognition (ASR). ASR is presented as a technique in which knowledge about speech has been gradually incorporated. The currently most successful statistical ASR still leaves room for improvement by more appropriate incorporation of knowledge about speech. In this thesis, knowledge about segmental duration is chosen, since this knowledge is particularly lacking in the current ASR systems. The scope and methodology of the current study are discussed. The speech recognisers and databases used in this study are described. An outline of the thesis content is given along with the development of the thesis project.

1.1 Introduction

This thesis addresses various possibilities of incorporating knowledge about segmental duration in speech into a statistically based speech recogniser. The main goal of the current study is to obtain insight into the complicated nature of segmental duration and its role in a speech recogniser.

In Section 1.2, the state of the art in automatic speech recognition (ASR) will be briefly reviewed, as well as the problem of further improvement. In Section 1.3, the statistically based recognition technique chosen for this study will be briefly introduced. The research activities aimed at the improvement of recognition performance by incorporating "knowledge", and the reasons why knowledge about duration was chosen for the current study, will be discussed. Section 1.4 defines the scope, methodology, and technical environment for the current study. Section 1.5 presents the arrangement of the thesis contents, along with the logical development of the thesis project.

1.2 Automatic speech recognition

Automatic speech recognition (ASR) is a method for recognising spoken messages by computers. ASR already finds useful applications, such as *voice commands* used to control a TV set or a personal computer, especially when manual control is impossible or undesirable. The current state of the art in ASR can be exemplified with the following two applications. One is the ability to recognise spoken texts of a single speaker nearly error-free in a real-time dictation task (e.g., Gauvain et al., 1994; Markowitz, 1995), and the other is the ability to recognise texts of very large or unlimited vocabulary, such as excerpts from the Wall Street Journal, read by many speakers, with only a few percent word errors (Pallett et al., 1995; Ljolje et al., 1995; Woodland et al., 1995). In these applications, noise-free speech is used.

However, there still is a long way to go (Cole et al., 1995) before meeting the ultimate challenge for ASR, such as a high recognition score under *adverse* conditions (e.g., noisy or with competing speakers). The main reason for this is that, in many respects, human speech communication is a much more complicated process (e.g., Allen, 1996; O'Shaughnessy, 1987; Moore, 1994) than what is implemented in the current ASR systems. However, improvement of ASR performance is probably best achieved through technical development, instead of by mimicking humans.

1.3 HMM and knowledge in ASR

The first success in ASR (Jelinek, 1976; Lee, 1989) for difficult real-life tasks (e.g., continuous speech or large vocabulary) came with the introduction of the hidden Markov model (HMM), a mathematical tool used to model speech

units (Rabiner, 1988 & 1989, Rabiner & Juang, 1986). HMM lately had to compete or was combined with artificial neural nets (NN) (Bourlard & Wellekens, 1990; Hochberg et al., 1995; Lippmann, 1987; Morgan & Bourlard, 1995). Nowadays nearly all large-scale recognisers are based on either or both these techniques. Technically, it is the combination of the ever-increasing power of modern computers and some basic algorithms for HMM and NN, that makes ASR practically possible. This study concentrates on HMM-based ASR only.

1.3.1 Research activities in HMM-based ASR

Research activities trying to improve the performance of HMM-based ASR can be listed in the following way:

- Modification of the basic mathematical assumptions of HMM or introduction of structures that are additional to the conventional HMM (Deng, 1992 & 1996; Deng et al., 1992; Levinson, 1986; Ostendorf & Roukos, 1989; Sun & Deng, 1994; Sun et al., 1994; Wellekens, 1987);
- Modification of the criterion for HMM training (Ephraim et al., 1989; Ephraim & Rabiner, 1990; Gauvain & Lee, 1994; Juang & Katagiri, 1992; Juang & Rabiner, 1990; Juang et al., 1995; Lee & Gauvain, 1995; Lee & Mahajan, 1990; Merhav & Lee, 1993);
- Improvement of particular system components such as acoustic front-end processing (Ljolje, 1994b) and language models (e.g., Shih et al., 1995);
- Incorporation of extra knowledge that is missing in conventional ASR (Burshtein, 1995; Dai et al., 1994; Deng & Braam, 1994; Gupta et al., 1987 & 1992; Kenny et al., 1991). Our approach belongs here.

Although state-of-the-art ASR cannot yet function as well as human beings, it already leads to many useful applications. Examples of applications are voice-input information inquiry systems (e.g., Gauvian et al., 1996; Strik et al., 1996). Substantial research efforts are actually invested in the development of such applications rather than in ASR technique itself.

Another useful application of ASR is to use a recogniser for automatic segmentation (of phonemes, for instance). There is an increasing demand for automatic or semi-automatic segmentation of large speech databases, since manual segmentation is very time-consuming and therefore expensive. HMM, sometimes combined with other techniques, provides reasonable to high quality automatic segmentation (Brugnara et al., 1993; Ljolje & Riley, 1991; Pauws et al., 1994; Svendsen & Kvale, 1990; Vorstermans et al., 1995 & 1996).

1.3.2 Knowledge representation in ASR

In the ASR technology of the last two decades, mainly two approaches exist: the rule-based approach (Zue, 1985) and the statistical pattern recognition approach based on HMM (Jelinek, 1976). We will argue that in both approaches human knowledge about speech is used, but that this knowledge is *represented* in different ways[1]. The rule-based approach represents the knowledge in the form of explicit rules that describe the relation between the acoustic realisations of speech and the linguistic units. The HMM-based approach defines the system structure given by the HMM, and it further collects *statistics* about the acoustic realisations of speech and puts them onto the HMM structures[2]. In other words, *statistical knowledge* is used. Currently, the HMM approach outperforms the rule-based approach. Our interpretation for this situation is that the current rules are not good enough to accommodate the complicated variations in speech patterns, whereas the statistical approach provides a rich *possibility* to explain the variations. It is also possible to combine the two approaches. For example, in the Dutch NWO Priority Programme of an advanced telephone-based information system (Boves et al., 1995), two approaches are compared and/or combined. The "grammar-based" approach uses rules, and the "corpus-based" approach directly uses the subtrees from the corpus, to parse input strings.

However, the present situation does not imply that the better performing HMM-based systems already have all the knowledge about the speech variations built in. HMMs are mathematical models. The internal structure of HMMs themselves is *not* based on any specific knowledge of speech. The fact that the HMM structure allows for a rich modelling ability does not imply that the modelling of all aspects of speech is achieved perfectly with the standard HMM, and that no more knowledge can be added. It is the topic of the current study to explore additional possibilities to incorporate our knowledge about speech into a conventional HMM-based recogniser.

Since the first technical attempt (e.g., Davis et al., 1952), ASR has become much more sophisticated. The whole process of ASR development can be seen as gradually incorporating different pieces of speech knowledge into the system by applying proper techniques. It can be seen that the representation forms of "knowledge" should be more general than the explicit rules, and should include descriptions of speech at various levels. In any conventional recogniser, for instance, the definition of the phoneme inventory requires knowledge about the phonemes of the language. Other examples of

---

[1]It is also the author's belief that the ultimate form of the knowledge representation in ASR will not be rule-based only. Therefore, the terminology in this study does not follow the convention of exclusively calling the rule-based systems knowledge systems.

[2]More precisely, this includes not only the HMM structure, but also a highly organised structure of the whole recogniser (Levinson, 1985), such as the separation of the language models and acoustic models.

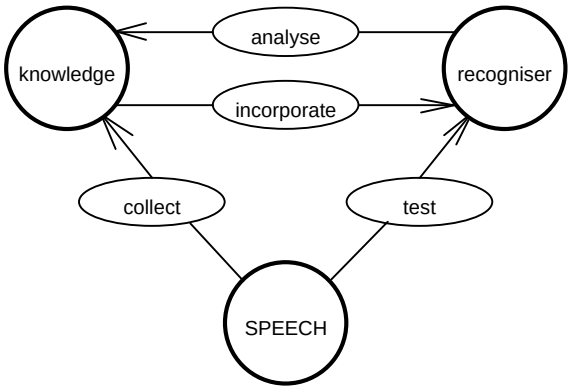


Figure 1.1 Conceptual illustration of knowledge incorporation in ASR. Each ellipse is an activity of speech scientists. For example, we *analyse* the recogniser as how well the speech knowledge is already *incorporated*.

incorporating knowledge are the imitation of the spectral resolution mechanisms in the human ear by a mel-scale filterbank, and the use of dynamic features as inspired by perception studies. There can be two types of allocations of knowledge: in the system structures, and in the parameter values. Adding new pieces of knowledge to an existing system can then be implemented as either adding new structures (e.g., Chapter 5 of Lee, 1989), or modifying the parameter values (e.g., various training algorithms). These are actually the two implementations of this study (see end of Section 1.5).

The process of incorporating knowledge about speech in a recogniser can be illustrated in Figure 1.1. It can be seen that this is a data-driven (from speech) bottom-up approach. Knowledge about speech is collected from the regularities in speech, and is further incorporated into a speech recogniser. The behaviour of an existing recogniser is analysed (also tested by direct speech input) to see how much and how well a certain piece of knowledge is already contained in it. If this is insufficient, further incorporation is needed.

1.3.3 Lack of durational knowledge in HMM-based ASR

It is observed that the knowledge about one specific feature of speech is not incorporated well in most state-of-the-art HMM-based ASR, namely the knowledge about the duration of the phonetic segments, particularly the phones (acoustic realisations of phonemes plus other non-speech sounds and pauses). Therefore, in this study, we chose to concentrate on durational knowledge. The detailed reasons for this choice are as follows:

1. There exists a huge body of knowledge in the literature about segmental duration and duration-related phenomena, that resulted from extensive research in this field (Crystal & House, 1988a & 1988b; Van Heuven & Pols, 1993; Nooteboom, 1970; Pitrelli, 1990; Pitrelli & Zue, 1989; Riley, 1992; Van Santen, 1992; Van Santen & Olive, 1990). A very simple

example of durational knowledge is that well-articulated stressed vowels systematically have a longer duration than vowels in other situations;
2. There is a very often cited notion about the improper duration modelling ability of HMMs (e.g., Russell & Moore, 1985). This is basically because the (standard) HMM states are defined at the frame level whereas phone duration is a segmental level measure. More details of our own research results on this topic will be given in Chapter 4 and 5 of this thesis;
3. Major efforts presented so far in literature to repair the improper durational behaviour of HMM have been either computationally too expensive (hidden semi-Markov model or HSMM, Levinson, 1986; Mitchell et al., 1995), or too simple (minimal duration, e.g., Gupta et al., 1992). Neither approach found a good match with the intrinsic complexity of the segmental duration itself. There is a sustained interest in duration modelling to improve recognition (Anastasakos et al., 1995; Dumouchel & O'Shaughnessy 1995; Gauvain et al., 1994; Magrin-Chagnolleau et al., 1995; Marcus & Zue, 1991; Nicol et al., 1992);
4. It is phonetically interesting to see the impact of incorporating knowledge on segmental duration (being a "simple" one-dimensional phonetic parameter) on the performance of ASR. Most phoneticians will believe that this knowledge should improve the system performance. On the other hand, it is also interesting to check whether the durational knowledge as presented in textbooks fully reflects the regularities extracted from a specific natural speech database, for the purpose of incorporation into ASR.

1.4 Scope, method and technical environment

1.4.1 Scope and method

The **scope** of the current study covers various possible approaches about incorporating durational knowledge into HMM-based speech recognition. However, the most commonly used HSMM will *not* be studied in detail (see Chapter 4). Attempts have been made to relate duration modelling to one practical application domain of ASR, namely large vocabulary and continuous speech, by using such a database for both duration analyses and recognition tests. It is not the aim of the current study to make our recogniser work in "real-time". Therefore the efficiency of the algorithms will be of secondary importance. Furthermore, only "clean" speech will be used. In the whole thesis "monophone" HMMs are used (rather than triphones or even larger units), in order to keep duration modelling manageable, as well as to make the effect of duration modelling tangible (more reasons for this will be elaborated in Chapter 4).

The general **method** in the current study could be described as "to look for methods" (a methodology study). The actual ways of incorporating knowledge depend both on the type of knowledge and on the structures in the existing HMM system[3]. The speech knowledge will be sought from the regularities in the speech database. Above all, each method has to be tested in terms of its contribution to recognition performance.

The basic approach of this study is mathematical and technical, which can be explained by the background of the author as well as by the complicated nature of both durational knowledge and of the HMM. For the sake of smooth reading along the main theme, however, mathematical and technical details are postponed to the appendices at the end of each chapter, which can be skipped. Furthermore, the main topic of the current study is *not* necessarily the technical improvement of the ASR performance. The current study aims at obtaining insight into the interactions between the two complicated systems (phonetics and HMM), using a specifically *chosen* parameter (segmental duration) as a probe. It is then not expected that manipulation of this parameter alone will lead to an *optimal* recognition performance, because most other issues in the standard recogniser are not tackled in our study. Therefore (Bourlard et al., 1996) it will not be the purpose of this study to compete with the performance score of our system against the most advanced systems. This does not mean, however, that we do not carefully look at the performance scores of our own system, because *changes* in scores may provide insight. Scores are reported and analysed for each test.

1.4.2 Recognition systems used

In this study, we have basically used two recognition systems for different purposes. The first one is called REXY, a discrete-density small-vocabulary system developed by Van Alphen (1992) at the Institute of Phonetic Sciences of the University of Amsterdam. Another system is based on a flexible tool for building very easily any continuous-density recogniser, called HTK (HMM Tool kit, version 2.0A), which was developed at Cambridge University (Young, 1992). REXY is only used in Chapter 3. The general specifications needed for a recogniser will be introduced in Chapter 2, and the detailed specifications of our systems will be given in later chapters in which the recognisers are used.

1.4.3 Speech databases used

It is a common practice in most ASR research to use pre-recorded speech in the training and testing phases, instead of live speech directly from a speaker. This recorded speech material, together with its orthographic

[3]This justifies the use of word "incorporating" in the thesis title (not just "adding").

Table 1.1. The three speech databases used and their main features.

database	language	#speakers	amount of material	vocabulary	labelling
TIMIT	American English	630	6,300 sentences	6,100 words	phone & word
Bloemendal	Dutch	1	1.5 hours	3,511 words	phone & word[4]
REXY	Dutch	1	3x100+34 sentences	230 words	phone & word

transcription and sometimes its phonetic annotation, is stored in the form of a speech database. The use of well-acknowledged databases is important for comparing results. For the current study we have used three databases with different features (Table 1.1), however with the common feature that they only contain continuously read speech. More details about TIMIT are given in Appendix 1.1 since this is the database used most in this study.

1.5 History of the thesis work and outline of the thesis

The starting point of this thesis work was the following. HMMs have a set of parameters that also govern their durational behaviour to a certain extent (Ten Bosch, 1991). HMMs trained with speech data of one speaking rate are most suitable for the test of speech data at that same rate (see indirectly Pallett et al., 1994). Using various rate-specific sets of HMMs for different rates may then improve recognition performance. We anticipated that in this way some relations could be found between the HMM parameters and the acoustic-phonetic parameter *segment duration*. This idea was directly implemented in a pilot study, using a small part of the Dutch "Bloemendal" database which was spoken at two speaking rates (Van Son & Pols, 1990) to train two sets of HMMs. We did not proceed along this line because it turned out that the amount of training material available for two speaking rates was too small. More substantially, such an implementation does not pinpoint the problem *which HMM parameters* are relevant to duration or rate.

The development of the actual thesis work has gone through several stages and this is reflected in the arrangement of the thesis chapters. First, we realised that the problem of duration modelling has two complicated aspects, namely the phonetic segmental duration and the recognition system (algorithms) based on HMM. The basics of the HMM as used in this study are introduced in Chapter 2. Thereafter, the front-end processing of HMM-based ASR is discussed in Chapter 3, with our attempts for optimisation, and some considerations for duration modelling. Then the durational probability density function (dpdf) of the standard HMM, is derived and discussed in Chapter 4, with an emphasis on the dpdf at the level of HMMs with multiple states, instead of the dpdf at the state level. In Chapter 5, the durational knowledge is incorporated in terms of the dpdf of context-free monophone

[4]Only for about half of the material.

HMMs. This is achieved in a training procedure constrained by the phone durational statistics of the actual speech data. Both Chapter 4 and 5 discuss context-independent (CI) duration modelling by HMMs themselves. This appeared to be insufficient; therefore attempts to model context-dependent (CD) duration are discussed in subsequent chapters. In Chapter 6, CD durational statistics are collected by detailed analyses of speech data, as influenced by various contextual factors, using the TIMIT database. In Chapter 7, CD durational statistics are incorporated into the HMM-based recognition system in the post-processing phase of recognition. Finally in Chapter 8 we summarise how we actually incorporated the durational knowledge into the recogniser, in the current study.

Technically, in this study, the CI durational knowledge is incorporated into *new* HMM parameter values (through constrained training) using the *old* structure of the conventional HMM-based recogniser (Chapter 5). The CD durational knowledge is incorporated into *new* structures of a duration model outside of the HMMs, as well as in a post-processing mechanism (Chapter 7).

Appendix 1.1 Description of the TIMIT database

The TIMIT acoustic-phonetic continuous speech corpus is developed mainly by the Massachusetts Institute of Technology (MIT) and Texas Instruments (TI). It contains a total of 6,300 sentence utterances, 10 read by each of 630 speakers from eight major dialect regions (abbreviated as "dr") of the USA, where the speakers lived during their childhood years, except for dr8.

dr1:	New England	dr5:	Southern
dr2:	Northern	dr6:	New York City
dr3:	North Midland	dr7:	Western
dr4:	South Midland	dr8:	Army Brat (moved around)

The whole database is designed to have a (suggested) division into a train set and a test set. The number of speakers is not equally distributed among the dialect regions, nor the gender (30% female), nor the two sets (about 1/4 in test set). These distributions are tabulated as follows:

	train set			test set			whole database		
dr	# male	# female	total	# male	# female	total	# male	# female	total
1	24	14	38	7	4	11	31	18	49
2	53	23	76	18	8	26	71	31	102
3	56	20	76	23	3	26	79	23	102
4	53	15	68	16	16	32	69	31	100
5	45	25	70	17	11	28	62	36	98
6	22	13	35	8	3	11	30	16	46
7	59	18	77	15	8	23	74	26	100
8	14	8	22	8	3	11	22	11	33
total	326	136	462	112	56	168	438	192	630

The text material consists of three types of sentences. There are two dialect calibration sentences (sa), 450 phonetically-compact sentences (sx), and 1,890 phonetically-diverse sentences (si). These sentences were read by different number of speakers:

sentence type	# sentences	# speakers per sentence	total	# sentences per speaker
dialect (sa)	2	630	1,260	2
compact (sx)	450	7	3,150	5
diverse (si)	1,890	1	1,890	3
total	2,342		6,300	10

All the text materials are meaningful short sentences containing about 40 phonemes each. Examples of three sentences, one for each type, are as follows (the numbers are purely administrative):

```
sa1:      She had your dark suit in greasy wash water all year.
si1027:   Even then if she took one step forward he could catch her.
sx127:    The emperor had a mean temper.
```

Since the same two sa sentences were read by all 630 speakers, they are usually not used for the purpose of training and testing automatic speech recognisers. So in this study we will only use a total of 5,040 sx and si utterances. For convenience (e.g., for testing the parameter settings in the recogniser), a subset of 192 utterances is defined by TIMIT, called a "core test set", which consists of all five sx and all three si utterances from two male and one female speaker from each of the 8 dialect regions. These 24 speakers in the core test set were chosen in such a way that each speaker read a set of five *different* sx texts. The "test set" is extended from the core test set by including all the speakers who read any of the sx texts in the core test set. This (extended) test set contains 1,343 utterances (2,373 different words), and the training set contains the remaining 3,696 utterances (4,891 different words). The union of the two sets consists of 6,100 different words. The above division of training and test sets guarantees that no sentence text appears in both sets, and that no speaker appears in both sets. Furthermore, no speaker spoke the same sentence text as any other speakers in the core test set.

The transcription is provided at both word and phoneme levels, labelled mainly by hand (see the documentation on the CD-ROM). The segment beginning and ending pointers are given in terms of number of samples. The speech is recorded in quiet and sampled at 16 kHz with a resolution of 16 bit per sample. A total of 61 phonetic symbols are used which are represented in ASCII form (called TIMITBET[5]). Furthermore, a lexicon is provided in which each word in the TIMIT database has a single norm pronunciation, with the positions of the primary and secondary stresses marked. No actual word stress or sentence accent realisations are provided.

[5]A list of the TIMITBET symbols is shown in Table 3.9 of Chapter 3 in this thesis.

2

HMM: ITS BASICS AND ITS RELATION WITH PHONETIC SEGMENTAL DURATION

Abstract

In this chapter the basics of standard HMM are introduced, as a preparation for further chapters to come. The basic concepts and assumptions, the parameter set, the essential problems and important algorithms of HMM, are presented. Then the basic system components of a recogniser, and task and evaluation issues of HMM-based recognition are discussed. Finally a relation between the HMM parameters and the modelled durational pdf is given for the simplest case of a single state model.

2.1 Introduction

In order to incorporate durational knowledge into HMM-based automatic speech recognition (ASR), we first have to understand what HMM is. Based on our knowledge of HMM, we are able to understand, and probably modify some of its durational behaviour. In Section 2.2, general mathematical basics of HMM, and technical issues related to HMM-based ASR, are presented. In Section 2.3, HMM set-ups specific to this study are given. Finally in Section 2.4 a simple relation is given between HMM parameters and a measure of segmental duration. Most of the material in this chapter is not the result of the author's original research, but of author's understanding of the existing techniques (presented in a compact way), written in preparation for the next chapters. Only the standard HMMs and the most used ASR techniques are presented.

2.2 Basics of HMM and statistical speech recognition

The hidden Markov model (HMM) is the basic model frequently used to model speech in a recogniser (Rabiner & Juang, 1993). The internal structure of HMM is not derived from any knowledge of speech. However, HMMs are used in a speech recogniser to calculate quantities related to speech (a *calculation model*).

2.2.1 Mathematical formulation of HMM as used in speech recognition

In order to understand the HMM, we must first look at a Markov model and a stochastic process in general. A stochastic process specifies certain probabilities of some events and the relations between the probabilities of the events in the same process at different times. A process is called Markovian if the probability at one time is only conditioned on a finite history. Therefore a Markov process is a simple stochastic process suitable for engineering purposes. *State* is a concept used to help understand the time evolution of a Markov process. *Being in a certain state at a certain time* is then the basic event in a Markov process. The Markov process used in most ASR practice is based on the following further simplifications of the general Markov process:

1. only defined at *discrete* time steps $t = 1,2,\cdots,T$;
2. only with a *finite* number of states $\{s_t\} = \{i\}$, $i = 1,2,\cdots,n$;
3. the history is only *one step* long (a first-order Markov process) $P(s_t|s_{t-1},s_{t-2},\cdots,s_{t-k}) = P(s_t|s_{t-1})$.

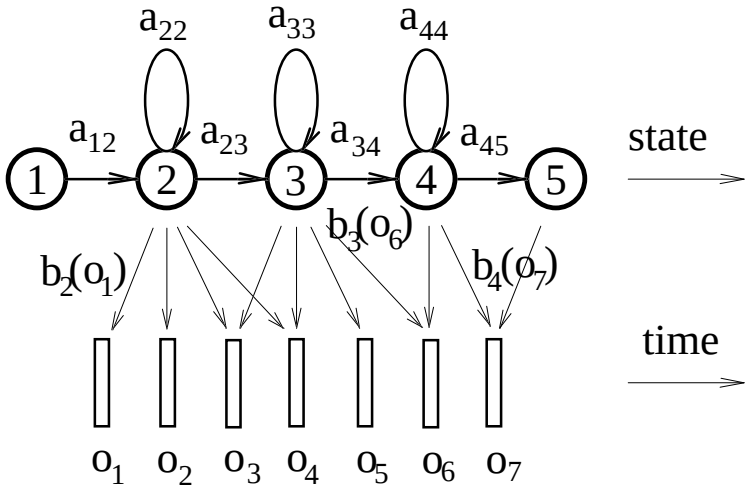


Figure 2.1 Schematic drawing of a hidden Markov model with 5 states. Each state is denoted as a circle with an index number. All between-state transitions (probability a_{ij}) and selfloops (probability a_{ii}) are represented as thick arrows. The events of observation of some example acoustic vectors from all except the first and the last states are indicated by thin straight arrows (probability $b_j(\boldsymbol{o}_t)$). In order to be consistent with the notations in this chapter, all states except the first one have observation probabilities assigned to them.

A whole Markov process thus produces a sequence of states $S = s_0 s_1 \cdots s_T$. The basic issues in using HMM in speech recognition will be discussed in the following sub-subsections.

2.2.1.1 Two basic sets of parameters

The HMM is an extension of a Markov model. The set of states and two basic sets of parameters together specify an HMM, which characterises the stochastic process (Figure 2.1). The first set of parameters of the HMM are called *transition probabilities* and are defined as

$$a_{ij} = P(s_t = j | s_{t-1} = i), \qquad (i, j = 1, 2, \cdots, n).$$

This can also be written in matrix form $A = \{a_{ij}\}$. For the Markov process itself, when the previous state is known, there is a certain probability to transit to each of the other states.

In order to make a link to the speech signals, a (probabilistic) function is attached to each state[1]. Firstly the acoustic speech signal is converted into a time sequence of *observation vectors* $\boldsymbol{o}_t$ defined in an acoustic space[2]. The sequence of vectors is called an *observation sequence* $\boldsymbol{O} = \boldsymbol{o}_1 \boldsymbol{o}_2 \cdots \boldsymbol{o}_T$, with each $\boldsymbol{o}_t$ a static representation of speech at t. What the probability function

[1]We take a formalism with *state-assigned* observation probabilities in this study, whereas another equivalent formalism uses *transition-assigned* probabilities.

[2]The actual technique to perform this conversion and the actual composition of the vectors will be discussed in Chapter 3.

calculates is called the *observation probability* and is given by the second set of the HMM parameters, defined as

$$b_j(\mathbf{o}_t) = P(\mathbf{o}_t \mid s_t = j), (t = 1,2,\cdots,T; j = 1,2,\cdots,n),$$

with its matrix form $B = \{b_j\}$. The composition of the parameters $\lambda = (A, B)$ defines an HMM (In the HMM literature there is another set of parameters, the probability that the HMM starts at initial time $\boldsymbol{\Pi} = \{\pi_j\}$. However, for cases like ours where all the HMM always start at the first state, $s_0 = 1$, this Π can be included in A).

2.2.1.2 One fundamental paradigm

The basic way of using HMM in speech recognition is to model different well defined phonetic units w_l (e.g., words in the usual sense, but these can also be any other units) in an inventory $\{w_l\}$ for the recognition task, with a set of HMMs (each with parameter λ_l). To recognise a word w_k from an unknown $\boldsymbol{O}$ is to find basically

$$w_k = \arg\max_l \{P(w_l \mid \mathbf{O})\},$$

The probability P is usually calculated indirectly using Bayes' rule (e.g., Scharf, 1991)

$$P(w_l \mid \mathbf{O}) = \frac{P(\mathbf{O} \mid w_l) P(w_l)}{P(\mathbf{O})}.$$

Here $P(\boldsymbol{O})$ is constant for a given $\boldsymbol{O}$ over all possible w_l. The *a priori* probability $P(w_l)$ only concerns the *language model* of the given task, which we assume here to be constant too. Then the problem of recognition is converted to calculation of $P(\boldsymbol{O}|w_l)$. In the simplest whole-word recogniser we use λ_l to model w_l, therefore we actually need to calculate $P(\boldsymbol{O}|\lambda_l)$ (More general recognition systems, in which λ_l does not model w_l, can be found later in Subsection 2.2.2 and in Rabiner & Juang (1993)). To be able to calculate $P(\boldsymbol{O}|\lambda_l)$, one needs two basic assumptions.

2.2.1.3 Two basic assumptions

The **first assumption** is the Markovian assumption itself. Because the transition probability a_{ij} is only conditioned on one previous state, a_{ij} is a constant. The *transitions* occurring at different times and in different states are then independent, and this therefore gives rise to the calculation of the probability of a state sequence $S = s_0 s_1 \cdots s_T$ to be a product

$$P(S|\lambda) = \prod_{t=1}^{T} P(s_t | s_{t-1}) = \prod_{t=1}^{T} a_{s_{(t-1)} s_t}.$$

It can be seen from the definition of observation probability b that the observation sequence $\boldsymbol{O}$ is also a stochastic process. However, it is in general not Markovian (e.g., Baum et al., 1970). The process $\boldsymbol{O}$ is a probabilistic function of the Markov process, therefore the mathematical property of $\boldsymbol{O}$ depends also on the actual form of b. From now on it is clear that the adjective *hidden* in the term HMM means that the Markov model itself is hidden as seen from the side of speech. This can be further seen in Figure 2.1. When an HMM models an observation sequence of speech, each state can in general generate *any* $\boldsymbol{o}_t$ (with some constraints), but each with a different probability. It is unknown which state generates which vector. Therefore, the actual state sequence $S = s_0 s_1 \cdots s_T$ that generates a given $\boldsymbol{O}$ is unknown. We can also say that the *underlying* Markov process is hidden. We can spell out the term HMM as *a model underlain with a hidden Markov process* (the underlying process is Markovian, and the process of the HMM is not).

Despite the fact that the probabilistic property of $\boldsymbol{O}$ is unknown, one usually also assumes that the events of observing $\boldsymbol{o}_t$ at different t are independent (this is necessary for all the HMM-based calculations). This **second assumption** for HMM leads to, for a given S,

$$P(\boldsymbol{O}|S,\lambda) = \prod_{t=1}^{T} b_{s_t}(\boldsymbol{o}_t).$$

With $P(S|\lambda)$ and $P(\boldsymbol{O}|\lambda)$ as calculated above using the two assumptions, the joint probability of $\boldsymbol{O}$ and S being generated by the model λ can be calculated

$$P(\boldsymbol{O},S|\lambda) = P(S|\lambda)P(\boldsymbol{O}|S,\lambda).$$

However, in reality, the state sequence S is unknown. Then one has to sum over all S in order to get $P(\boldsymbol{O}|\lambda)$. An efficient way to do this will be given later on when we discuss the two relevant algorithms.

2.2.1.4 Three essential problems

In order to use HMM in ASR, a number of practical problems have to be solved. Historically, the difficulties with these problems have delayed the use of HMM in ASR until the last two decades or so, although the theory of Markov processes has existed for a much longer time (see, e.g., Jelinek, 1976 for earlier references). Juang & Rabiner (1992) provided one of the best ways to list the major problems:

1. **The evaluation problem:** One has to evaluate the value $P(\boldsymbol{O}|\lambda)$ given only $\boldsymbol{O}$ and λ, but not S. Without an efficient algorithm, one has to sum over n^T possible S with a total of $2T \cdot n^T$ calculations, which is impractical.
2. **The estimation problem:** The values of all λ_l in a system have to be determined from a set of sample data. This is called *training*. The problem

is how to get an optimal set of λ_l that leads to the best recognition result, given a training set.

3. **The decoding problem:** Given a set of well trained λ_l and an $\boldsymbol{O}$ with an unknown identity, one has to find $P(\boldsymbol{O}|\lambda_l)$ for all λ_l. In the *recognition* process, for each single λ_l, one hopes, instead of summing over all S, to find a single S_M that is most likely associated with $\boldsymbol{O}$. S_M also provides the information of boundaries between the concatenated phonetic or linguistic units that are most likely associated with $\boldsymbol{O}$. The term *decoding* refers to finding the way that $\boldsymbol{O}$ is coded onto S.

In both the training and recognition processes of a recognition system, problem 1 is involved. The solutions to these problems are presented next.

2.2.1.5 Two important algorithms

The two important algorithms that solve the essential problems are both named after their inventors: the Baum-Welch algorithm (Baum et al., 1970) for parameter estimation in training, and the Viterbi algorithm (see, e.g., Juang & Rabiner, 1992) for decoding in recognition (in some recognisers the Viterbi algorithm is also used for training).

The essential part of the **Baum-Welch algorithm** is a so-called expectation-maximisation (EM) procedure, used to overcome the difficulty of incomplete information about the training data (the unknown state-sequence). In the most commonly used implementation of the EM procedure for speech recognition, a maximum-likelihood (ML) criterion is used. The solutions for the ML equations give the closed-form formulae for updating HMM parameters given their old values (see Appendix 2.1, and Chapter 5 for the special case of a linear HMM). In order to obtain good parameters, a good initial set of parameters is essential, since the Baum-Welch algorithm only gives a solution for a local optimum (e.g., Merialdo, 1993). However, for speech recognition, such a solution often leads to sufficiently well performance.

The basic shortcoming of the ML training is that maximising the likelihood that the model parameters generate the training observations is not directly related to the actual goal of reducing the recognition error, which is to maximise the discrimination between the classes of patterns in speech. In addition to the ML-based training, other training algorithms also have been developed, based on criteria different from ML and other mathematical treatments (Ephraim & Rabiner, 1990). Although they sometimes provide better results than ML, they are more difficult to understand and to implement. Therefore the ML is still the most widely used one (under certain conditions, ML yields equivalent results as the other algorithms (Lee & Gauvain, 1995)). There is also an ML-training algorithm which uses only "dominant" state sequences, instead of all the sequences in the Baum-Welch

algorithms (Merhav & Ephraim, 1991a, 1991b), aiming at reduced training cost at an acceptable accuracy.

The **Viterbi algorithm** essentially avoids searching through an unmanageably large space of HMM states to find the most likely state sequence S_M by using step-wise optimal transitions (see Appendix 2.1). In most cases, the state sequence S_M yields satisfactory results for recognition (Juang & Rabiner, 1992). But in other cases, S_M does not give rise to state sequence corresponding to the most correct words. Other algorithms exist to find solutions better than S_M of the Viterbi algorithm (e.g., the N-best algorithms (Schwartz & Chow, 1990)), however at the cost of an increased searching effort.

2.2.2 Basic system components

In a modern HMM-based speech recogniser, there are four major, relatively independent, components (their functions may have mutual interactions):

1. Front-end processing;
2. Acoustic HMM models;
3. Language models;
4. Search strategy.

Their roles will be briefly discussed here.

The **front-end** converts the digitised speech samples into a sequence of observation vectors $\boldsymbol{O} = \boldsymbol{o}_1\boldsymbol{o}_2\cdots\boldsymbol{o}_T$. Each $\boldsymbol{o}_t$ is a vector of some 10 to 40 components, obtained by some basic signal-processing analysis applied on the speech samples. The details of $\boldsymbol{o}_t$ can be found in Section 3.2 of Chapter 3.

The **acoustic HMM models** are used to model the acoustic patterns of speech units. Each HMM in a system then models various acoustic realisations of the same unit. Usually an inventory of basic units is chosen for a certain task. Sometimes the basic modelling units are not the same as the units for recognition. For example, in the common practice of continuous speech recognition, the units for modelling are phones (phonemes plus some non-speech sounds) or triphones, whereas the units for recognition are words. The technical reason for that is the following. When the vocabulary is large, modelling each word with an HMM will make the system very large and yet the discrimination between the words is not optimal. This is because the total discrimination power is limited given the defined acoustic space. Some different words share the same basic unit such as a phone, whereas each word HMM makes a duplicate of the region in the acoustic space for that phone, which is a waste. Therefore an alternative is to define the HMMs for the basic units like the phones while finding ways in the recogniser to compose the words based on these basic units. This often used approach is called a *sub-word-unit based* system. The collection of the allowed

concatenations of the basic units for all the words in the system lexicon is called a *pronunciation dictionary,* and such pronunciations can be altered following some *phonological rules*.

The link between the HMM states and the acoustic space, i.e., the observation probability function b, has basically two possible forms: a *discrete-* or a *continuous*-density (DDHMM or CDHMM). A DDHMM system uses more memory, but less computation time, than a CDHMM does. The b parameters of both DDHMM and CDHMM can be estimated by the Baum-Welch algorithm. More details can be found in Subsection 3.2.3 of Chapter 3.

The **language model** (LM) in a speech recogniser defines the allowable transitions between the words. It contains mostly information similar to syntactical constraints in a natural language, but can contain other types of information as well. In implementation, usually two types of LM are used, i.e., the statistical regular grammars and the deterministic regular grammars. The often used statistical grammar, is a word n-gram. For example, a bigram (n=2) specifies the probability of observing the current word given the identity of the immediate previous word. The parameters of such a model should also be estimated from the text material of the speech database. The probabilities of those between-word transitions unseen in the training corpus should be filled with certain small values to allow these transitions occurring in the test corpus. The other type of a deterministic regular grammar, specifies all the legal transitions between words[3]. It can be estimated from a training text, or provided by a closed recognition task in which only sentences obeying a small set of syntax rules are allowed.

Language modelling plays a substantial role in speech recognition. This is because of the limited power of the acoustic modelling and the huge number of possible word-combinations which would be unlimited without an LM. The result is that the search space is greatly reduced and that the recognition scores are increased. The design of the mathematical structures of LM, methods for its parameter estimation, and improvement of efficiency when using the LM, attract active research (e.g., Shih et al., 1995), especially for large and close-to-real-life recognition tasks.

Language match factors (LMF) are artificial factors set to balance between the probabilities $P(\boldsymbol{O}|w_l)$ calculated with the acoustic models and $P(w_l)$ from the language models (see Sub-subsection 2.2.1.2 about the basic paradigm) occurring at transitions between words w_l. This is needed because the two models are defined separately, while the probability of the whole utterance

[3]In practice, a deterministic regular grammar can have a similar *form* as a bigram, being a matrix of between-word transition probabilities. Then all the allowable transitions have a probability of $1/N$ where N is the number of succeeding words, and all the forbidden transitions have a zero probability. Such a grammar is also called a *word-pair* grammar.

$P(U)$ consists of the two, and this $P(U)$ governs the recognition process. Usually the LMF consists of two parameters p and q and $P(U)$ is calculated[4]

$$P(U) = \prod_{w_l \in U} p[P(\boldsymbol{O}|w_l)]^q P(w_l),$$

where p is called *word insertion penalty* and q is called *language model scaling factor*. Adjusting p and q will result in different trade-offs between insertion and deletion of words (see also Subsection 2.2.3). Values of p and q have to be determined empirically for optimal performance of each (kind of) recognition task.

Given the above three system components (front-end, acoustic model and LM), the **search strategy** is still of prime importance, for both accuracy and efficiency, and for a good trade-off between them (Austin et al., 1991; Haeb-Umbach & Ney, 1994; Katoh & Kodha, 1994; Kenny et al., 1993). This component becomes necessary because of the huge s*earch space* which consists of combinations of all possible connections between the phones and between the words in a large-vocabulary recognition task. An exhaustive search is neither technically possible nor necessary, since many connections are not plausible. A search algorithm often consists of a definition of some sub-structures in the whole search space (e.g., word alternative candidates at a decision point conditioned on a finite-length word history), some measures of the sub-structures to distinguish between the retained and discarded parts of the sub-structures (e.g., likelihood difference between the alternatives), and the actual procedural design of the algorithm. A search algorithm is often designed and optimised on specific types of recognition tasks. The Viterbi algorithm in the previous sub-section is the most used search algorithm. We will come back to searching problems in Chapter 7 in combination with the duration modelling problems.

Interactions between the various system components mentioned above could exist, but these are very difficult to specify. In this respect it can be observed that, when comparing various successful recognisers, independently developed in different research laboratories over the world, one sees that each of them emphasises specific aspects or specific system components, while leaving the other parts without special effort. However, the performance of these systems are quite comparable (e.g., Pallett et al., 1995). This indicates, at least in part, that modelling of the same aspects can be achieved in different system components. This may also indicate that the potential of each of the components has not yet been fully explored. However, if the potential of one component is exhaustively explored, this leaves less potential to improve the system performance by further exploring other components.

[4]The large dynamic range of different kinds of probabilities will often exceed the finite numerical range of digital computers. Therefore all the probabilities in a recogniser are actually calculated in logarithmic form. Multiplication becomes addition, which is faster.

This is true given the current way to divide the system into components. For example, the word-correct score (77%) of a DDHMM system (Chapter 3) with a relatively inaccurate acoustic model can be significantly improved to nearly 100% by casting a strong language model, while the improvement may be smaller for a CDHMM system with good acoustic modelling (94% without language model).

The present study of integrating durational knowledge into ASR involves modifications of nearly all four system components mentioned above, although the LM will be barely touched. This implies that our approach does not actually alter the overall structure of a conventional HMM-based recogniser.

2.2.3 Tasks of recognition and segmentation; evaluation issues

As we have seen so far, the state of the art in speech recognition is still far from perfect. Therefore the formal specification of (the difficulty of) a task is important for evaluating the performance of recognisers[5]. The usual specifications for evaluations are (e.g., Pols, 1994):

1. **speaker dependency**: If the recogniser is trained to recognise only the speech of a single speaker, it is called *speaker-dependent*, since other speakers have to train the recogniser separately. If speech of many speakers is used to train and to test the system, the system is assumed to operate *speaker-independently*. If the system is trained by a basic set of speakers, and in testing, it adapts its behaviour to each speaker's voice, this is called a *speaker-adaptive* system (e.g., Leggetter & Woodland, 1994).
2. **vocabulary size**: The number of words that the recogniser knows. Unknown words will also be (wrongly) 'recognised' as one of the words in the system, or as "garbage". For small to medium-sized vocabulary (a few hundred words), a measure called *perplexity*[6] may be used to further distinguish between tasks with the same vocabulary size, but with a different averaged number of words to choose from at a decision point. For large-vocabulary and real-life materials (e.g. newspaper articles), however, perplexity is rarely used since this value is about the same for all such tasks.

---

[5]If otherwise the performance of the recognisers is near perfect, the evaluation process can simply take all the difficult aspects for recognition simultaneously (e.g., noisy, very-large vocabulary, continuous speech of multiple speakers). For the present-day recognisers, however, such an evaluation will produce very low scores.

[6]For a bigram in which for given word i the probability is $p_{j|i}$ to transit to word j, the perplexity is 2^H, where $H = -\sum_i \sum_j p_{j|i} \log_2(p_{j|i}) / W$ is the *per-word entropy* and W the total number of words. For a word-pair grammar in which word i can be followed by n_i different words, H is simplified to $H = \sum_i \log_2(n_i) / W$. For a "no grammar" task, i.e., each word may follow any words, the perplexity is just W (Bahl et al., 1983; Press et al., 1989).

3. **continuous speech versus isolated words**: If the speaker is asked to make short pauses between words, we are dealing with an isolated-word recogniser. If the speaker can speak naturally, it is a continuous-speech recogniser).
4. **clean versus corrupted speech**: Most laboratory recognisers are evaluated with pre-recorded clean speech. Recognising (real life) speech, corrupted with noise, reverberation, and/or having reduced bandwidth (e.g., through a telephone line), i.e., under more practical conditions, will be more difficult and will usually require additional techniques.

For evaluating **recognition performance**, one uses four kinds of percentages as in the following table, for W words in the text materials of the whole test set to be compared with, W_C correct, W_S substituted, W_D deleted, and W_I inserted words, respectively:

word correct score	word accuracy score	word error rate	total word error rate
$\frac{W_C - W_S - W_D}{W} \times 100$	$\frac{W_C - W_S - W_D - W_I}{W} \times 100$	$\frac{W_S + W_D}{W} \times 100$	$\frac{W_S + W_D + W_I}{W} \times 100$

The last two percentages are more suitable for evaluating performance with word correct scores approaching 100 %.

The **automatic segmentation performance** of a recogniser is also an aspects for evaluation. One measures the deviation of the unit boundaries versus the correct boundary location (as defined by the hand labelling). One counts the total number of occurrences of the deviation to both left and right directions within a pre-defined threshold, e.g., 20 *ms*. Then one evaluates the accuracy of segmentation by the percentage of such occurrences.

2.3 Basic HMM setups in this study

Whereas the detailed and the specific system setups will be given in Chapter 3, some of the basic setups for the HMM will be given below.

For any HMM-based recogniser, one has to define the basic phonetic units to be modelled by HMMs. In a sub-word-based recogniser, one can choose either *context-dependent* (CD) or *context-independent* (CI) basic units. Phones are often used as CI units (referred to as *monophones*). A single CI-HMM for a phone *f* stores information about all occurrences of *f* in all possible contexts. Since acoustic variations of a phone exist due to contextual effects, modelling all the different acoustic variants of a phone by just a single monophone HMM may be insufficient. But this is task-dependent.

On the other hand, CD units are defined on a *central phone* together with its left and right contexts. An often used CD unit is the *triphone*. A triphone models the same acoustic segment as a phone; however different triphone HMMs are made for the phone of a given identity with different immediate

left and right context phones. One hopes to model some phonetic contextual effects by CD units. It is evident that the total number of CD units will be much larger than that of CI units. Therefore, this is again a problem of trade-off between modelling accuracy and system complexity. In this study, we basically use CI monophones, whereas some context effect will be modelled by duration models (Chapter 7), instead of the usual triphone modelling.

Within a single HMM, not only the number of states, but also the *topology* (the connections between the states) needs to be specified. For our recogniser based on HTK, we made a choice to use linear models instead of the so-called K.F. Lee model (Lee, 1989) with parallel branches (the topology of it will be shown in Chapter 4), for three reasons: (1) The K.F. Lee model requires much task-specific tuning on both the parameter values and the structure (special tying), therefore its optimal performance is very task-dependent. In large scale recognisers developed after Lee (1989), nearly always linear HMMs are used. (2) We made an informal pilot test with the small REXY database to compare between a seven-state left-to-right K.F. Lee topology and a five-state linear topology with only the upper branch of the K.F. Lee model. The result is that the simpler linear model out-performed the more complicated one. (3) For our purpose of duration modelling, a linear model gives more direct insight into the relation between the segmental duration and the HMM parameters (more detailed discussion on this will be given in Chapter 4), than would a model with parallel branches.

2.4 Durational measure of HMM as a segmental feature

The durational behaviour of an HMM is usually characterised by a durational pdf $P(d)$. For a single state i (or an HMM with only one selfloop), the value $P(d)$ is the probability of the event of staying in i for exactly d steps. This event is in fact the joint event of taking the selfloop for $(d-1)$ times and taking the out-going transition (with probability $1-a_{ii}$) just once (Figure 2.2). Given the Markovian assumption, and from probability theory (e.g., Papoulis, 1990), $P(d)$ is simply the product of all the d probabilities:

$$P(d) = a_{ii}^{d-1}(1-a_{ii}).$$

It can be seen that this is a geometrically decaying function of d. It has been

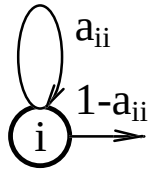


Figure 2.2 A single state i of an HMM with selfloop probability a_{ii}.

argued (e.g., Russell & Moore, 1985) that this is a source of inaccurate duration modelling with the HMMs since no actual physical events in speech obey this function. Durational pdf's for HMMs with more than one state will be discussed in Chapter 4. It can also be seen that $P(d)$ is not a basic quantity of an HMM directly characterised by a single parameter, but a derived measure of an HMM. $P(d)$ is defined on a sequence of time frames corresponding to a segment, so it is a segmental feature.

Appendix 2.1 Baum-Welch and Viterbi algorithms

The major formalism of the algorithms described in this appendix follows a combination of the treatments in Juang and Rabiner (1992), Kamp (1991) and Young (1992), to achieve best clarity and generality. It is assumed that the HMM is in state 1 at $t = 0$ ($s_0 = 1$) and in state n at $t = T$ ($s_T = n$).

A2.1.1 Baum-Welch algorithm

The Baum-Welch algorithm for estimating HMM parameters $\lambda = (A, B)$ with a given set of speech training data $\{\boldsymbol{O}\}$ follows the method of maximum likelihood (ML). This means that the values of λ will be chosen so that the likelihood $P(\boldsymbol{O}|\lambda)$ is maximised for the given $\{\boldsymbol{O}\}$. Here the whole set of training data $\{\boldsymbol{O}\}$ can be explicitly written as $\{\boldsymbol{O}^r\}, r = 1, 2, \cdots R$ (R is the total number of observation sequences $\boldsymbol{O}^r$), and each $\boldsymbol{O}^r = \boldsymbol{o}_1^r \boldsymbol{o}_2^r \cdots \boldsymbol{o}_{T_r}^r$, where each $\boldsymbol{o}$ is an observation vector. Given the two basic assumptions of HMM, we have

$$P(\boldsymbol{O}|\lambda) = \sum_S P(\boldsymbol{O}, S|\lambda) = \sum_S P(S|\lambda) P(\boldsymbol{O}|S, \lambda) = \sum_S \prod_{t=1}^{T} a_{s_{(t-1)}s_t} b_{s_t}(\boldsymbol{o}_t), \tag{1}$$

where $S = s_0 s_1 \cdots s_T$. Maximisation of $P(\boldsymbol{O}|\lambda)$ (or its logarithm) directly over λ is difficult. In the Baum-Welch algorithm, an auxiliary function

$$Q(\lambda, \bar{\lambda}) = \sum_S P(\boldsymbol{O}, S|\lambda) \log P(\boldsymbol{O}, S|\bar{\lambda}) \tag{2}$$

is incremented instead in an iterative procedure, where each iteration begins with an existing λ and ends at a new value $\bar{\lambda}$. The whole iteration process begins with some initial values and stops when the improvement decreases below a threshold[7]. Such an indirect maximisation of $P(\boldsymbol{O}|\lambda)$ via Q is satisfactory in speech recognition practice.

[7]The procedure of the Baum-Welch algorithm has been given in a more general scheme known as the *expectation-maximisation* (EM) algorithm (Dempster et al., 1977). In the EM algorithm, the first step (E-step) is to find the Q function which is a *conditional expectation*, and the second step (M-step) is to maximise Q. The EM scheme can be used to solve a larger class of problems than the Baum-Welch ML algorithm.

The fact that increasing $P(\boldsymbol{O}|\lambda)$ can be equivalently achieved by increasing Q has been proven (for clarity see, e.g., Kamp, 1991), namely

$$\log\frac{P(\boldsymbol{O}|\bar{\lambda})}{P(\boldsymbol{O}|\lambda)} \geq Q(\lambda,\bar{\lambda}) - Q(\lambda,\lambda).$$

Then the problem of maximising $P(\boldsymbol{O}|\lambda)$ is replaced by maximising $Q(\lambda,\bar{\lambda})$ with respect to $\bar{\lambda}$. When the summation over S in (2) is written explicitly as transitions at each time step t and between all pairs of states (i, j) of the HMM, and when the probability is written for the A and B parts separately using (1), (2) can be written as

$$Q(\lambda,\bar{\lambda}) = \sum_{r=1}^{R}\sum_{t=1}^{T_r}\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_{t-1}^{r}(i,j)\log\bar{a}_{ij} + \sum_{r=1}^{R}\sum_{t=1}^{T_r}\sum_{j=1}^{n}\gamma_{t}^{r}(j)\log\bar{b}_{j}(\boldsymbol{o}_t^r), \tag{3}$$

where n is the number of states, and the likelihood of the whole training set $\{\boldsymbol{O}^r\}$ is considered. The two γ values, usually called "counts", are the *a posteriori* probabilities:

$$\gamma_{t-1}^{r}(i,j) = P(s_t = j, s_{t-1} = i \mid \mathbf{O}^r, \lambda) = \frac{P(\mathbf{O}^r, s_t = j, s_{t-1} = i \mid \lambda)}{P(\mathbf{O}^r \mid \lambda)};$$
$$\gamma_{t-1}^{r}(i) = P(s_{t-1} = i \mid \mathbf{O}^r, \lambda) = \frac{P(\mathbf{O}^r, s_{t-1} = i \mid \lambda)}{P(\mathbf{O}^r \mid \lambda)} = \sum_{j=1}^{n}\gamma_{t-1}^{r}(i,j).$$

Note that, in order to relate the two γ values, the time index for the second γ has been shifted to $t-1$. The actual way to calculate these counts will be given later. Each individual term in (3) has the form $\sum_{j=1}^{n} w_j \log y_j$, i.e., a function of y_j, which under the unity constraint $\sum_{j=1}^{n} y_j = 1$ (e.g., for A-parameters $\sum_{j=1}^{n} a_{ij} = 1, i = 1,2,\cdots,n$) has the general solution (a single point) for the maximum:

$$y_j = \frac{w_j}{\sum_{i=1}^{n} w_i}, \qquad j = 1,2,\cdots,n.$$

Particularly, the solution to calculate new A-parameters from the old ones (called re-estimation formula) is

$$\bar{a}_{ij} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_{t-1}^{r}(i,j)}{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_{t}^{r}(i)}.$$

Re-estimation formulae for B-parameters depend on the actual form of B. As an example, we look at an often used mixture of M Gaussian densities:

$$b_j(\mathbf{o}_t) = \sum_{m=1}^{M} C_{jm}\mathbf{N}(\mathbf{o}_t;\ \ _{jm}, \Sigma_{jm}),$$

where ***N*** denotes "normal", i.e., the Gaussian function, and C_{jm} is the mixture weight of m-th mixture of state j. In order to obtain the re-estimation formulae for the parameters of the Gaussian density, we need an *a posteriori* probability of the m-th mixture, also calculated from the old HMM parameters:

$$P_j(m \mid \mathbf{o}_t^r) = \frac{C_{jm}\mathbf{N}(\mathbf{o}_t^r;\ _{jm}, \Sigma_{jm})}{\sum_{m=1}^{M} C_{jm}\mathbf{N}(\mathbf{o}_t^r;\ _{jm}, \Sigma_{jm})}.$$

the new B-parameters are

$$\overline{C}_{jm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_t^r(j)P_j(m \mid \mathbf{o}_t^r)}{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_t^r(j)};$$

$$\overline{\ }_{jm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_t^r(j)P_j(m \mid \mathbf{o}_t^r)\mathbf{o}_t^r}{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_t^r(j)P_j(m \mid \mathbf{o}_t^r)};$$

$$\overline{\Sigma}_{jm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_t^r(j)P_j(m \mid \mathbf{o}_t^r)(\mathbf{o}_t^r - \overline{\ }_{jm})(\mathbf{o}_t^r - \overline{\ }_{jm})^{\tau}}{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_t^r(j)P_j(m \mid \mathbf{o}_t^r)}.$$

where τ denotes transpose.

The γ terms will be calculated efficiently with a "forward-backward algorithm". In essence, this algorithm makes an efficient organisation of the many different state sequences into step-wise transitions. This algorithm is practically instead of theoretically necessary, just like the importance of FFT for calculating DFT.

Two probabilities are defined of observing part of the sequence ***O***, called the *forward probability* α and the *backward probability* β:

$$\alpha_i(t) = P(\mathbf{O}_1^t, s_t = i); \beta_j(t) = P(\mathbf{O}_{t+1}^T \mid s_t = j).$$

These are calculated recursively:

$$\alpha_j(0) = \begin{cases} 1, j = 1; \\ 0, j \neq 1; \end{cases}$$
$$\alpha_j(t) = \left[\sum_{i=1}^{n}\alpha_i(t-1)a_{ij}\right]b_j(\mathbf{o}_t), t > 1;$$

$$\beta_i(T) = 1, \qquad 1 \le i \le n;$$
$$\beta_i(t) = \sum_{j=1}^{n} a_{ij}b_j(\boldsymbol{o}_{t+1})\beta_j(t+1), \quad t < T.$$

The γ terms are then calculated

$$\gamma_t(i,j) = \frac{\alpha_i(t)a_{ij}b_j(\boldsymbol{o}_{t+1})\beta_j(t+1)}{\alpha_n(T)};$$
$$\gamma_t(i) = \frac{\alpha_i(t)\beta_i(t)}{\alpha_n(T)}.$$

The term "Baum-Welch" algorithm is often confused with "forward-backward", probably because in ASR technique, the former always uses the latter, for efficient calculations. As we have seen, the latter is only a small part of the former.

A2.1.2 Viterbi algorithm

The Viterbi algorithm finds a best state sequence for a given $\boldsymbol{O}$, in the following 4 steps.

1. Initialisation:

$$\delta_0(i) = \begin{cases} 1, i = 1; \\ 0, i > 1; \end{cases}$$

2. Recursion: for $1 \le t \le T$ and $1 \le j \le n$:

$$\delta_t(j) = \max_{1 \le i \le n} \left[\delta_{t-1}(i)a_{ij}\right] b_j(\mathbf{o}_t);$$
$$\psi_i(j) = \arg\max_{1 \le i \le n} \left[\delta_{t-1}(i)a_{ij}\right];$$

3. Termination (maximal probability P^* of $\boldsymbol{O}$ and best exiting state i_T^*):

$$P^* = \max_{1 \le i \le n} \delta_T(i);$$
$$i_T^* = \arg\max_{1 \le i \le n} \delta_T(i);$$

4. Back-tracking of state sequence: for $t = T-1, T-2, \cdots, 1$:

$$i_t^* = \psi_{t+1}(i_{t+1}^*).$$

In essence, the decision of choosing the "best" state sequence is achieved in two steps (2 and 4). In step 2, the best state coming to each of the states at current time is registered, however which current state *is* the best is left for the next time step to choose. Only in the back-tracking, the single best path is linked from the end to the beginning. The Viterbi algorithm avoids comparing between all the possible paths (to achieve the necessary efficiency) by only looking at the state-transitions registered as optimal. This is equivalent to a thorough comparison through all the paths; therefore it is optimal in the sense that the path with the highest accumulated score is guaranteed to be found.

3

OPTIMISATION ON PRE-PROCESSING FOR SPEECH RECOGNITION*

Abstract

This chapter is a small deviation from the main theme of the thesis: duration modelling. Pre-processing of the speech signal is vitally important for optimal recognition. In order to compare the recognition performance with and without duration modelling, one needs a baseline recogniser. It is important then that the pre-processing part of the baseline recogniser is optimised. In this chapter, optimal representations by means of the usual pre-processing methods in HMM-based recognition systems, as well as a further processing step after the usual one, are discussed. This further processing is based on removing the correlation between the components of the observation vectors, by means of linear transformations. The impact of such optimisations on recognition performance is first shown with both discrete- and continuous-density recognisers using filterbank parameters. Then this is tested with cepstrum coefficients on the continuous-density system, for which a further refined technique of component transformation is implemented. The best pre-processing schemes shown in this chapter will be used as the baseline system in subsequent chapters.

*This chapter is a substantially revised and extended version of Wang, Ten Bosch & Pols (1993).

3.1 Introduction

Pre-processing is the first part of the recogniser that the speech signals have to go through. Only if the useful speech information can be extracted and represented in a suitable form in the pre-processing part, can the rest of the recogniser work properly. Research in the area of optimising the pre-processing has been active since decades ago (e.g., Rabiner & Schafer, 1978) until very recently (Ljolje, 1994b; Bocchieri & Wilpon, 1993). This indicates that the problems involved in pre-processing are still not fully understood, and further research may still improve its quality with respect to the performance of the recogniser. Pre-processing also has interactions with other parts of the recogniser, thus technical developments in other parts may also trigger renewed interest in further optimisation of pre-processing.

Research reported in this chapter covers two sets of tests on pre-processing (Table 3.1), performed during the course of the project. The main concern of the first set of tests was the extent of the impact of correlation removal and dimensional reduction of analysis frames in various systems. For this purpose we used the filterbank parameters, because the correlation between them is known to be severe and thus the impact of removal would be tangible. For this test we used a Dutch speech database REXY (Van Alphen, 1992; also Chapter 2 of this thesis). Both a discrete-density system and a continuous-density system (Wang et al., 1993) were tested.

For the second set of tests, various aspects were improved upon the first one. We chose cepstrum-related coefficients which are increasingly becoming more popular. The TIMIT database was used (see, e.g., Zue et al., 1990 and Chapter 1 of this thesis), in order to have the baseline performance related with an internationally known database. Although in both sets of tests we applied linear transformations for the purpose of correlation removal, in the second set of tests we applied a further refined technique (transformation on the basis of HMM states instead of on a global basis) with the hope to obtain better results. Only a continuous-density system was used in the second set of tests, because of its better modelling accuracy.

In Section 3.2, we will briefly introduce the basic representation of speech obtained from the usual pre-processing, suitable for HMM-based recognition. In this section both filterbank and cepstrum will be introduced. In Section 3.3, general techniques for optimisation by linear transformations will be presented. Then we will discuss the problems, and investigate the impact, of

Table 3.1 Summary of various aspects in the two sets of tests.

aspects	first set tests	second set tests
database	REXY	TIMIT
observation density	discrete, continuous	continuous
analyse feature	filterbank	cepstrum
transformation	global	global, state-specific

correlation removal in speech recognition tests. Tests with the filterbank (Section 3.4), and the cepstrum (Section 3.5), will be discussed. Section 3.6 summarises the insights obtained from our experience with pre-processing.

3.2 Basic speech representation for HMM: pre-processing

In order to analyse and to recognise speech by digital computers, the analogue signal is first converted into digital form. The digitised speech is a time sequence of samples at a constant *sampling frequency*, with each sample represented at a given *resolution* in bits. Not all information in the speech samples is relevant for the purpose of recognition[1]. The usual way to extract the relevant information is to use signal processing techniques to convert the speech samples into *analyse frames* or *basic vectors.* Sometimes the basic vector is further augmented with its time-derivatives (the first and second derivatives are also called Δ and $\Delta\Delta$, respectively). or frequency-derivative (also called *slope*), to provide some more information. The whole vector $\boldsymbol{o}_t = (o_{t1}, o_{t2}, \cdots, o_{tp})^\tau$ (τ denotes transpose) is called an *observation vector*, where p is its *dimensionality*. The whole speech signal is represented as $\boldsymbol{O} = \boldsymbol{o}_1\boldsymbol{o}_2\cdots\boldsymbol{o}_T$, called an *observation sequence*, whereas the time interval between each $\boldsymbol{o}_t$ is called *frame shift*.

The actual speech processing techniques are based on feasible signal processing techniques and on some of our understanding of human speech production (Rose et al., 1996) and perception (O'Shaughnessy, 1987) mechanisms (Jankowski et al. (1995) compared between different pre-processing techniques). Two of the techniques, namely the filterbank and the cepstrum, are used in this study. The specifications of these two processing schemes, chosen for this study based on previous experience (e.g., Rabiner & Juang, 1993), are given in Table 3.2. Some specific issues concerning these two schemes are given in the next two subsections.

Table 3.2 Specifications of signal processing schemes for REXY and TIMIT databases.

database	sampling freq.	resolution	processing	basic dimension	frame shift
REXY	16 kHz	12 bit	filter bank	15	8 *ms*
TIMIT	16 kHz	16 bit	cepstrum	12	8 *ms*

3.2.1 Filterbank

Filterbank processing takes the (logarithmic) energy output of a series of adjacent band-pass filters which mimics the *frequency-selective mechanism* in the human cochlea. The output then represents a coarse spectrum. A conventional choice for the spacing of the centre-frequencies of the filters is

[1]The purpose of speech recognition is only to identify the spoken linguistic content, not other aspects of speech, such as speaking style and speaker identity.

based on the *critical-band* of the human auditory system, approximately along a logarithmic or a *mel* scale (e.g., O'Shaughnessy, 1987). The total frequency range should cover the most informative frequency range of speech, which is typically from 100 to 5,000 Hz.

One way to implement a filterbank is to build all the digital band-pass filters in the time domain (e.g., FIR filters[2], Van Alphen, 1992). Our test with the discrete-density system in this chapter uses this implementation. Another equivalent implementation is to perform an FFT[3] to obtain a fine spectrum and then to cut the portions out in the frequency domain for all the filters. The FFT is performed on a *block* of speech samples within an interval of 20 *ms*, and the blocks for the adjacent frames overlap. This is used for the HTK-based continuous-density system (Young, 1992).

3.2.2 Cepstrum

The cepstrum is defined as an inverse Fourier Transform of a logarithmic spectrum (Rabiner & Schafer, 1978). According to the implementation, there are basically two kinds of cepstrum analysis resulting in slightly different coefficients. One is a mel-frequency cepstrum coefficients (MFCC) based on an FFT filterbank. The other is an LPC-based cepstrum (e.g., Furui, 1989) which uses an extra assumption on the speech signal of being generated by an all-pole system. A mel-frequency cepstrum may also be derived from the LPC-based cepstrum by a bilinear transform (Oppenheim & Johnson, 1972). Both the LPC-based and FFT-based MFCC have been widely used. The LPC parameters may only be appropriate if one merely describes the vowel portions of speech, and if the analysis order is carefully chosen. For our recognition task of words embedded in continuous utterances which contain different kinds of phones, we choose to use the FFT-based MFCC (without extra model assumption), which may be optimal in an overall sense (our pilot study has shown this).

The MFCC coefficient c_l is calculated from a discrete cosine transform of the log filterbank outputs m_j (Davis & Mermelstein, 1980):

$$c_l = \sum_{j=1}^{M} m_j \cos\left(\frac{\pi l}{M}(j-0.5)\right), 1 \le l \le q, \tag{1}$$

where M is the number of the filters and q is the number of MFCCs. The m_j are obtained by applying triangular filters on the FFT spectrum on a mel-scale. The mel frequency is calculated as (O'Shaughnessy, 1987)

$$\text{Mel}(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right),$$

[2]Finite Impulse Response.
[3]Fast Fourier Transform.

3.2.3 Discrete and continuous observation probabilities

One basic operation in an HMM-based recogniser is to calculate the observation probability $b_j(\boldsymbol{o}_t)$ of a vector $\boldsymbol{o}_t$ for a state j of an HMM (Chapter 2). There are generally two kinds of $b_j(\boldsymbol{o}_t)$ which distinguish between a discrete-density system (DDHMM) and a continuous-density system (CDHMM). Since we will use both kinds of HMM, we will first introduce them in somewhat more detail.

DDHMM is often used in applications where system compactness is more important than modelling accuracy. A *codebook* $\{\boldsymbol{w}_k\}, k = 1,2,\cdots K$, is used and it is obtained by a separate vector-quantisation (VQ) procedure (using algorithms of, e.g., LBG or K-means, see, e.g., Gersho & Gray, 1992), which tries to find an adequate *partition* of the whole acoustic space into non-overlapping regions, each represented by one *codeword* $\boldsymbol{w}_k$. The HMM parameters $b_j(\boldsymbol{w}_k), k = 1,2,\cdots K$ are pre-calculated and stored in look-up tables. In training and recognition processes, $b_j(\boldsymbol{o}_t)$ is approximated by $b_j(\boldsymbol{w}_k)$ where $\boldsymbol{w}_k$ is the codeword closest to $\boldsymbol{o}_t$. The codebook size K determines the modelling precision (256 in this study). In a DDHMM system, *VQ-distortion* always exists due to the use of the finite number of codewords to represent the whole acoustic space. When multiple codebooks are used (e.g., one for the basic vector and another for its Δ vector), probabilities from individual codebooks are simply multiplied. Such a multiplication in fact assumes that the codebooks are independent of each other (This will be referred to as the *independence assumption* for codebooks).

CDHMM, on the other hand, takes a *parametrical* way of modelling, i.e., a well known continuous-valued pdf (probability density function) is used, where the pdf is governed by its parameters. The most used pdf's are Gaussian (Young, 1992) and Laplace (Haeb-Umbach & Ney, 1992). In this study we use the *Gaussian mixture density*. Each individual Gaussian component is

$$\mathbf{N}(\mathbf{o};\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^p|\Sigma|}}\exp\left\{-\frac{1}{2}(\mathbf{o}-\mu)^\tau\Sigma^{-1}(\mathbf{o}-\mu)\right\}, \tag{2}$$

where p is the dimensionality of $\boldsymbol{o}$. $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean vector and covariance matrix of the Gaussian pdf, respectively. $|\Sigma|$ is the determinant of $\boldsymbol{\Sigma}$. The whole mixture density is[4]

[4]In HTK (and some other recognisers), this continuous probability *density* value is used directly as a "probability" value. This value can be either larger or smaller than one, and for different compositions of $\boldsymbol{o}$, this value can be very sparsely distributed. However this does not cause a problem in most recognisers in which only the $b_j(\boldsymbol{o}_t)$ with the *same* kind of $\boldsymbol{o}$ (e.g., each being a 12-component MFCC) are compared. If one designs a recogniser in which probabilities of $\boldsymbol{o}$ with, e.g., different dimensionalities should be compared, special treatments must be applied.

$$b_j(\mathbf{o}_t) = \prod_{r=1}^{R}\left[\sum_{m=1}^{M_r} C_{jm}\mathbf{N}(\mathbf{o}_t;\mu_{jm},\Sigma_{jm})\right], \tag{3}$$

where R is the number of *data streams,* (e.g., three streams for the basic cepstrum parameters c, its first time derivative Δc, and second time derivative $\Delta\Delta c$, respectively), and M_r is the number of mixture components in stream r. The components of the vectors c, Δc and $\Delta\Delta c$ may also be concatenated into one large vector and used in one stream. Using such a mixture density with a sufficiently large M_r and choices of the mixture weights C_{jm} ($\Sigma_m C_{jm} = 1$) as in (3), nearly any actual acoustic data distribution can be modelled at a sufficient accuracy. The M_r per state can vary from 1 for simple recognition tasks, to 20 for more complicated modelling requirements. Some examples of Gaussian mixture densities are illustrated in Appendix 3.1.

In a CDHMM system, no observation probabilities are pre-calculated or stored. Instead, in principle, during training and recognition, the probability for each $\boldsymbol{o}_t$ should be calculated explicitly using formula (3). This then has a larger calculation load, but a smaller storage requirement, than a DDHMM system.

There also exists a version with a *semi-continuous-density* (Huang & Jack, 1989; Huang et al., 1990), and it is simulated in HTK as a "tied-mixture" configuration (Woodland & Young, 1993; Young & Woodland, 1993 & 1994). In such a system, in order to reduce the number of model parameters, all the states of all the HMMs in the system share a collection of a number of common Gaussian components, but each state j has its own mixture weights C_{jm}. One could also call these Gaussians codewords, because the way that all the states share them is similar to that in a discrete-density codebook. However, there are a number of differences. The Gaussian codewords overlap in their covering areas whereas the DDHMM codewords do not. In addition, in DDHMM, each time only a single codeword is chosen whereas in CDHMM, *all* (or a fixed reduced number of) the continuous Gaussians can be used for the calculation of each single $\boldsymbol{o}_t$.

3.3 Optimisation with linear transformation

In general there is a statistical correlation between the components of the observation vectors (Goldenthal & Glass, 1993; Niyogi & Zue, 1991) (for an analysis on correlation in filterbank parameters see Wang et al. (1992) and for correlation between cepstrum coefficients see Section 3.5 of this thesis and Ljolje (1994b)). In both a DDHMM and a CDHMM system, it would be better to work with uncorrelated components, but for different reasons (see the next section). In this chapter, component de-correlation is achieved with linear transformations on the observation vectors. We use two transformations, the first one is based on principal component analysis (PCA) (Okamoto, 1969;

Pols, 1977; Ljolje, 1994b), while the second is based on linear discrimination analysis (LDA) (Le Cerf et al., 1992; Haeb-Umbach & Ney, 1992; Haeb-Umbach et al., 1993).

Usually the correlation coefficients $r_{lm},(l,m=1,2,\cdots,p)$ between the individual components o_{tl} and o_{tm} of the observation vectors $\boldsymbol{o}_t$ are used to quantify the correlation,

$$r_{lm}=\frac{s_{lm}}{\sqrt{s_{ll}s_{mm}}}, \tag{4}$$

where the sample covariance matrix $\boldsymbol{S}=\{s_{lm}\}$ is calculated from

$$\mathbf{S}=\frac{1}{M}\sum_{\mathbf{o}_t\in\{\mathbf{o}\}}(\mathbf{o}_t-\bar{\mathbf{o}})(\mathbf{o}_t-\bar{\mathbf{o}})^{\tau};\bar{\mathbf{o}}=\frac{1}{M}\sum_{\mathbf{o}_t\in\{\mathbf{o}\}}\mathbf{o}_t,$$

where M is the number of samples in the data set $\{\boldsymbol{o}\}$.

Both PCA and LDA transformations are given in the form of $\boldsymbol{y}_t=\boldsymbol{V}^{\tau}\boldsymbol{o}_t$ where the transformation matrix $\boldsymbol{V}=(\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_p)$ is composed of eigenvectors $\boldsymbol{v}_i$ of either PCA or LDA system (e.g., Cooley & Lohnes, 1971). After the transformation, $\boldsymbol{y}_t$ is used instead of $\boldsymbol{o}_t$. For PCA the eigenvectors are obtained from $\boldsymbol{SV}=\boldsymbol{V\Lambda}$ with the eigenvalues $\boldsymbol{\Lambda}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_p)$ which are usually arranged $\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_p\geq 0$. With such an arrangement, the set of lower indexed principal components $\{y_{tl}\}$, $l=1,2,\cdots p'<p$ explain the largest part of the total variance in the data. Therefore this provides a way of *data compression* when one only uses some low-indexed components instead of all of them. In this study, this is called a *truncation* process.

For LDA, two matrices, a *between-class* matrix $\boldsymbol{U}$ and a *within-class* matrix $\boldsymbol{W}$ are defined:

$$\mathbf{U}=\sum_{k=1}^{K}R_k(\mu_k-\mu)(\mu_k-\mu)^{\tau};$$
$$\mathbf{W}=\sum_{k=1}^{K}\sum_{\mathbf{x}_l\in k}(\mathbf{o}_l-\mu_k)(\mathbf{o}_l-\mu_k)^{\tau},$$

where R_k is the number of vectors $\boldsymbol{o}$ in class k, $\boldsymbol{\mu}_k$ is the mean in that class, and $\boldsymbol{\mu}$ is the global mean. The classes can for example be defined on the phones, i.e., those $\boldsymbol{o}$ which belong to a phone according to the available hand-labelled training speech data. The eigen-system of LDA is $(\boldsymbol{W}^{-1}\boldsymbol{U})\boldsymbol{V}=\boldsymbol{V\Lambda}$ and the eigenvectors can be arranged in the same way as PCA. When also truncation is performed, the lower indexed components will retain maximal *discrimination* between all the classes in an averaged sense.

It can easily be verified (Appendix 1 of Wang et al., 1992) that a PCA transformation will fully diagonalise the sample covariance matrix $\boldsymbol{S}$ of the transformed vectors, whereas the LDA transformation does not (because LDA is not based on matrix $\boldsymbol{S}$ of the whole data set). In other words, the

components $(y_{t1}, y_{t2}, \cdots y_{tp})$ in $\boldsymbol{y}_t$ from the PCA transformation are completely uncorrelated while those from LDA may still be partially correlated.

The PCA transformation is determined by $\boldsymbol{S}$ which is calculated using a set of vectors $\{\boldsymbol{o}\}$. There can be two ways to define $\{\boldsymbol{o}\}$, resulting in two types of transformation: When $\{\boldsymbol{o}\}$ contains the observation vectors of the whole training set, the transformation is called *global*; whereas when $\{\boldsymbol{o}\}$ only contains $\boldsymbol{o}$ assigned to a state of an HMM, the transformation is called *state-specific*.

3.4 Impact of correlation between filterbank parameters

In this section for filterbank parameters, only a global transformation is used to remove the component correlation. We will systematically investigate the impact of the correlation of filterbank-based observation vectors in various aspects. Some of these aspects will only be clearly described during the investigation because they are specific to the particular ways of de-correlating transformations. A brief summary of the main aspects is as follows:

1. DDHMM vs. CDHMM systems;
2. composed vs. separated vectors (one vs. several codebooks or streams);
3. between-vector correlation (BVC) vs. within-vector correlation (WVC);
4. different pairs of vectors (one pair being the basic and its frequency derivative; another pair being the basic and its time derivative).

First some specifications and implementation issues about component correlation will be discussed in some detail. Then the impact on the recognition scores will be presented.

3.4.1 Specifications of correlation in DDHMM and CDHMM systems

Difference between WVC and BVC: The correlation in the data vectors can be categorised into two types: The correlation between the components within each data vector is called a within-vector correlation (WVC), and the correlation between the basic vectors and the derivatives is called a between-vector correlation (BVC). In a system using only one vector (which may be a large vector concatenated with components of both basic and the derivatives, though), there is only WVC (or the correlation is treated as WVC). In both DDHMM and CDHMM systems where two or more vectors are used, there exist both WVC and BVC. However, they may play different roles in the two kinds of systems.

BVC and VQ-distortion in DDHMM: There exists WVC in the data vectors (or the codewords). However, since a DDHMM system uses a non-parametrical way to model the data, the existing WVC does not violate anything. On the other hand, the BVC does violate the independence

assumption when multiple-independent codebooks are used (because if the vectors are correlated, they are not independent of each other). To avoid (or reduce) the effect of inaccuracy introduced by this violation, it may be better to put all the components in one vector and to use one codebook. However, we must also consider the effect of the VQ-distortion, for the overall performance. It is experimentally shown (Lee, 1989) that the total distortion is smaller in a system using multiple codebooks than in a system with a single codebook, when the total numbers of components of the vectors in the two systems are the same. Therefore, one cannot reduce the two effects simultaneously, and a compromised result will be seen when using either one or several codebooks.

BVC and WVC in CDHMM: In a CDHMM system, there is no VQ-distortion. The BVC should be similar to that in the DDHMM system, except that the discrete type multiple codebooks are now replaced by continuous multiple data streams. In addition, the WVC also plays a role. One has the freedom to choose the type of covariance matrix $\boldsymbol{\Sigma}_j$ for the Gaussian pdf of each state j as either a *full* one in which all the $p(p+1)/2$ values (p being the dimensionality of $\boldsymbol{o}$) are *free parameters*, or a *diagonal* one with only p parameters (all the off-diagonal values are set to zero). Choosing a full $\boldsymbol{\Sigma}_j$ will faithfully model whichever amount of WVC in the data, using a larger number of model parameters. In a statistical system with a finite amount of training data, the number of free parameters should always be kept as small as possible to avoid *under-training*. If, otherwise, a diagonal $\boldsymbol{\Sigma}_j$ is used while serious WVC exists in the data, it would lead to inaccurate modelling. Therefore, if one is able to remove the WVC by a transformation, a diagonal $\boldsymbol{\Sigma}_j$ may also be able to model the data distribution accurately[5].

3.4.2 Implementation issues of correlation removal by the transformation

Removing WVC or WVC plus BVC: When the two vectors in a pair are used in two separate codebooks (or streams), transformation performed on these two vectors separately will only remove the WVC in each vector. When the two vectors are put into one composed vector (by concatenating their components), a transformation on the composed vector will remove both WVC and BVC. This choice provides us with a way to separately investigate the different impact of the WVC and BVC. It is impossible with the current technique to remove merely the BVC without removing also the WVC, though.

Two pairs SP-SL and SP-dSP: One has to decide upon a few different kinds of data vectors to implement the correlation removal. We have chosen

[5]This discussion concerns a single Gaussian component and assumes that the data distribution is also multivariate Gaussian. More general cases that can be modelled with Gaussian mixtures will lead to different discussions (Appendix 3.1). No tests for such cases will be performed in this section for filterbank parameters.

the filterbank spectrum (SP) as the basic vector since the BVC between SP and its frequency-derivative "slope" (SL) is very serious. Due to the way SL is derived from SP ($o_k^{SL} = o_{k+2}^{SP} - o_k^{SP}$), SL is actually totally correlated to SP. This casts two problems. The first problem is on the transformation matrix. It can be verified (Wang et al., 1992) that matrix $\boldsymbol{S}$ of the vector composed of SP and SL is singular, thus the eigenvectors cannot be found. Therefore in both DDHMM and CDHMM systems, the transformations cannot be applied to the composed vector. It follows that removal of BVC cannot be implemented with the pair SP-SL. The second problem is about the covariance matrix $\boldsymbol{\Sigma}_j$ of the HMMs in a CDHMM system with a single Gaussian density (in this section). Because HMMs are trained with the speech data, the singularity in the data (represented by $\boldsymbol{S}$ of the pair SP-SL) would also cause singularity in $\boldsymbol{\Sigma}_j$. This is particularly true when hand-segmentation information is used as in the initialised training where the segmented data and the model have perfect match (resulting in $\boldsymbol{\Sigma}_i = \boldsymbol{S}$). In the later stages of training, where segment borders are relaxed from the labelling information, the $\boldsymbol{\Sigma}_j$ may not be numerically singular, but they would still be close-to-singular, thus very unreliable to use. Actually, even in a system without transformation, SP and SL vectors may not be used in one stream in a CDHMM with one Gaussian, because of the singularity of $\boldsymbol{\Sigma}_j$ (when the Gaussian pdf is not well defined). On the other hand, since a DDHMM system uses a non-parametrical way of modelling, the problem of singularity does not show up. So we can still use the original SP-SL pair in one composed vector in DDHMM. To conclude, by using the pair SP-SL we can only investigate the effect of WVC. We can use another pair, SP and its time-derivative Δ-spectrum (dSP) to investigate BVC. For this pair SP-dSP, the systematic large correlation and the related singularity problems do not exist (Wang et al., 1992).

Truncation after transformation: After any kind of transformation, data compression (truncation) in the vector components is possible by using only the low-indexed components. Truncation is only performed at some selected dimensionalities (number of components), which will be sufficient to show the effect of truncation.

Composite effect of VQ and truncation: For the DDHMM system, some refined procedures are performed to reduce the calculation burden (and to investigate the inaccuracy introduced by the procedures). For each kind of data vector before and after various transformations and truncation, a time-consuming VQ procedure is in principle needed to obtain a new codebook. We call such a VQ procedure an *explicit* procedure in which a new codebook is obtained with all such vectors transformed and truncated from the training set (Figure 3.1). Two simple approaches of obtaining the codebooks other than the explicit VQ procedures are proposed. The first approach is called *transformed* which transforms (and then truncates) the codewords of an original codebook (instead of transforming the data vectors). The second

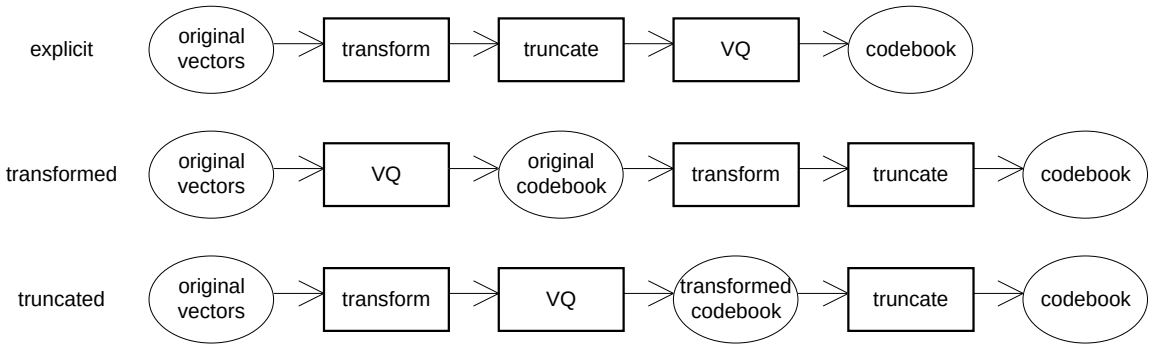


Figure 3.1 Three approaches to obtain transformed and truncated codebooks. Rectangles indicate processes, and ellipses are objects.

approach is called *truncated* which truncates the codewords of a codebook that is based on the full-dimensional transformed vectors. The first requires no VQ procedure in addition to the original VQ, while the second applies only one additional VQ for codebooks of all different dimensionalities. The order to perform the VQ and the transformation procedures is in general irreversible[6]. Therefore the accuracy of the codebooks obtained by either transformed or truncated procedure will in general be different from those obtained by the explicit procedure (see also Adlersberg & Cuperman, 1987). In this study, the actual qualities of the three kinds of codebooks will not be compared directly using any measures on the codebooks themselves. Instead, the overall effect of using different VQ and transformation procedures will be tested by their effect in recognition performance.

3.4.3 Impact of correlation in filterbank parameters on recognition

The impact of correlation and dimensionality of the acoustic observation vectors is tested in recognition runs using the REXY database (Van Alphen, 1992). The training set contains three repetitions of 100 sentences (a 230 word vocabulary) spoken by a single speaker, and the test set contains the fourth repetition of the first 34 sentences (a 110 word vocabulary) from the same speaker. No bigram is used here in order to emphasise the effect of correlation removal with such a relatively simple recognition task. Each word is considered equally probable (within the test set) and has only a single standard pronunciation (thus the perplexity of the task is just 110, see Chapter 1).

---

[6]Because, e.g., in the transformed version, the original codewords are obtained according to the original data distribution, and then it is only the codewords, not all the data vectors, that are transformed. In the explicit version, the vectors transformed and truncated from the original data generally have a different distribution from the original data, and this will give rise to a different codeword distribution. These two codebooks both lose some accuracy due to low dimensionality, but they have undergone different procedures so their resulting accuracy may be different.

The DDHMM system is implemented with the REXY system, while the CDHMM system is based on the HTK toolkit, both using the same speech data. The front-end processing produces 15 logarithmic filterbank outputs (SP), which are further converted to 15 time-derivatives (dSP) and 13 slope values (SL). Frame energy is not included in the observation vectors for the tests in this section. The two systems are kept as comparable as possible in that they use the same inventory of 41 Dutch phones (Van Alphen, 1992), the same pronunciation dictionary, and both use the filterbank parameters. However there is one difference in the transition topology of the HMMs. REXY uses the K.F. Lee model (Lee, 1989; see also Chapter 4) which has an upper branch with a cascade of three selfloops and three lower skipping branches without selfloops. For the HTK CDHMM system we use linear models, each having three selfloops. This difference between the setups of the two systems is not a problem since our main concern is not to compare the performance *between* a CDHMM and a DDHMM system. In the CDHMM system, each state observation probability contains a single Gaussian pdf because of the small amount of training data.

Because the full set of combinations of conditions discussed in the previous subsections would be too large, only selected conditions were tested at selected dimensionalities. The results are listed in the following tables. Some discussions specific to each test are also given. In all the tables, both the word *correct* score counting substitution and deletion errors, and the word *accuracy* score, including the insertion errors as well, are used. For simplicity, a fixed language-match-factor (LMF, which in effect trades off the insertion and deletion error, see Chapter 2) is used in all the tests[7].

Table 3.3 to Table 3.5 show scores of tests on the DDHMM system with SP and SL vectors transformed separately. Table 3.3 shows the effect both of PCA and LDA transformation, and of VQ distortion, at full-dimensionality. The 1-codebook setup uses the large vector concatenated from the *separately* transformed SP and transformed SL vectors. Therefore the WVC and BVC are the same for the 1-codebook and 2-codebook conditions. Because of the larger total VQ distortion with one codebook than with two separate codebooks (Lee, 1989), the scores for 1-codebook condition, both before and after transformations, are lower than the 2-codebook condition. The marginal increase in scores for the 2-codebook setup after transformations may be due to the fact that in the transformed acoustic space for SP and for SL, the codewords find more representative distributions during the VQ procedure. This is not possible for the 1-codebook condition for which the separately

[7]Because the transformation and truncation of the data vectors change the values of the acoustic probability, a fixed LMF cannot be optimal for all the different conditions. It can be seen from the scores in the tables that, e.g., the difference between the correct and accuracy scores is too large for some conditions (due too many insertions). Better care for this problem is taken in Subsection 3.5.5 for tests with the cepstrum.

Table 3.3 Effect of transformation, full dimensionality, DDHMM, with SP and SL vectors. Word percentage "correct" scores count substitution and deletion errors, and "accuracy" scores count also insertion errors.

	original		PCA		LDA	
# codebook	correct	accuracy	correct	accuracy	correct	accuracy
1	70.2	69.9	68.6	68.3	69.6	69.4
2	76.7	70.2	78.0	72.1	78.6	72.1

Table 3.4 Effect of truncation, 2-codebooks, DDHMM.

# component			PCA		LDA	
SP	SL	total	correct	accuracy	correct	accuracy
1	1	2	15.8	5.3	9.0	6.5
2	2	4	50.0	42.6	45.7	39.4
5	5	10	73.7	68.6	69.9	62.7
8	8	16	77.0	71.1	74.2	66.2
15	13	28	78.0	72.1	78.6	72.1

Table 3.5 Scores with three versions of codebooks at a total dimensionality of 10, DDHMM.

		PCA		LDA	
# codebook	version	correct	accuracy	correct	accuracy
1	explicit	61.2	60.9	66.8	66.5
	transformed	52.8	51.9	55.0	54.4
	truncated	63.4	63.0	63.0	62.4
2	explicit	73.3	68.6	69.9	62.7
	transformed	70.8	65.5	66.5	59.6
	truncated	72.7	66.2	69.9	63.4

In Table 3.5, the scores with three versions of codebooks obtained from the transformed vectors actually make the distribution in the 28-dimensional space more difficult for VQ, thus the scores decrease somewhat.

Since the 2-codebook setup is favourable, this is used in tests of Table 3.4, where the effect of truncation is shown. It can be seen that at a total dimensionality of 10, the scores are only slightly lower than at full-dimensionality of 28. In other tests to come, only the 10-dimensional and the full-dimensional conditions will be used.

SEQ chapter \c3.5, the scores with three versions of codebooks obtained from the three procedures are shown. The scores with the *transformed* version are much worse than for the *explicit* version, especially for the 1-codebook setup. The *truncated* version produced reasonable scores. Therefore the *truncated* VQ procedure can be useful for applications where reduction in VQ calculation time is required.

Table 3.6 shows the effect of WVC on the CDHMM between SP and SL vectors, done with PCA transformation only. Knowing the singularity problem with the SP-SL pair on CDHMM, only a 2-stream setup is used. It can be seen that the scores with the diagonal covariance matrix at full-dimensionality increase significantly after the PCA transformation (from 78.3% to 86.7%). This indicates the existence of WVC in the original vectors.

Table 3.6 Effect of WVC removal in PCA-transformed SP and SL vectors. '-' indicate impossible entries. All tests use a 2-stream setup on CDHMM.

# component				original		PCA	
SP	SL	total	cov. matrix	correct	accuracy	correct	accuracy
5	5	10	diagonal	-	-	76.1	70.2
5	5	10	full	-	-	81.4	76.4
15	13	28	diagonal	78.3	68.3	86.7	77.3
15	13	28	full	91.3	87.0	91.9	87.0

Table 3.7 Effect of removing WVC with separate transformation and removing both WVC and BVC with composed transformation, for SP and dSP vectors, on CDHMM. '-' indicate impossible or meaningless entries.

	# component			original-full		PCA-diagonal	
transform	SP	dSP	total	correct	accuracy	correct	accuracy
composed	-	-	10	-	-	85.4	83.5
	-	-	30	94.1	90.7	91.3	88.8
separated-	5	5	10	-	-	82.6	80.1
1-stream	15	15	30	as composed		94.1	91.0
separated-	5	5	10	-	-	as separated-	
2-stream	15	15	30	93.2	91.3	1-stream	

Table 3.8 Scores with DDHMM at full dimensionality, SP and dSP vectors.

original		PCA					
		composed-		separate			
2-stream		1-stream		1-stream		2-stream	
correct	accuracy	correct	accuracy	correct	accuracy	correct	accuracy
72.0	66.7	62.0	61.7	60.4	59.5	75.7	71.3

In the next test we only used diagonal matrices for the transformed vectors. In Table 3.7 the scores are shown for the 10-dimensional and full-dimensional conditions. Since a transformation on the vector composed of SP-dSP pair is possible, the BVC is also removed. The vectors transformed separately from the SP and dSP vectors may in principle be tested under two conditions: whether to put them in two separate vectors, or in one vector by concatenating the components. However this is unnecessary because in a CDHMM system with a single Gaussian, it follows from definitions (3) and (2) that, the observation probabilities calculated in both ways are exactly the same. The scores again show the efficient removal of WVC as was the case with the SP-SL pair. The removal of the BVC by transformation on the composed vector (which also removes the WVC) does not show an increase, but a slight decrease in scores both with respect to those with WVC-removal alone and the original vectors. Such results may be influenced by other factors than the data correlation that also affect the modelling accuracy, in

addition to the small amount of training and testing speech data used. For example, when performing a PCA transformation on the composed vectors which has a larger dimensionality than the separated vectors (30 versus 15 here), those higher-dimensional components in the transformed vectors probably contain very little useful information but mainly noise. When these components are still used, they take up the acoustic space and the model parameters, which may lower the overall discrimination power.

Another interpretation of the score decrease from the original-full to PCA-diagonal conditions in Table 3.6 and Table 3.7 is as follows. The complexity of the actual data distribution may not be close to a Gaussian in both original and transformed acoustic space, therefore it may require a mixture of Gaussians to make an accurate observation probability (see Appendix 3.1). Yet in these tests only one Gaussian is used. Therefore, e.g., for Table 3.6, after transformation, although the PCA-diagonal has a higher score (86.7% correct) than original-diagonal (78.3%), this is lower than the scores of original-full (91.3%) and of PCA-full (91.9%). This indicates that using a full covariance matrix in a single Gaussian may be more accurate than a single diagonal Gaussian, even with transformation, for a data distribution that is not Gaussian[8].

In Table 3.8, on the other hand, the comparison of the scores between the transformations on separate and composed vectors for SP and dSP shows just the opposite result on the DDHMM system as on the CDHMM in Table 3.7. This is clearly an indication that the effect of a smaller VQ-distortion overruled the effect of the noise in the higher-dimensional components. Another important reason for the decrease of scores in Table 3.7 and Table 3.8 after PCA transformation is that the amount of BVC between the original SP and dSP vectors is rather small (Appendix 1 of Wang et al., 1992, and Appendix 3.2 at the end of this chapter). When the removal of such a small BVC does not make much difference in the total correlation, other factors, such as the VQ-distortion, play the dominant role.

It has to be noted that if a bigram is used (perplexity 2.4), the word correct scores with such a small database, especially for the CDHMM system, will reach nearly 100% (shown with informal tests).

3.5 Impact of correlation between cepstrum parameters

Already for some time the cepstrum coefficients have been popular for the pre-processing of the speech recognisers and have led to a more robust signal representation than the filterbank processing. Therefore we will apply the methods discussed above (correlation analysis and removal) to cepstrum (MFCC) coefficients as well. It has also become common practice to use the

[8]Larger number of Gaussians will be used in tests of Section 3.5 with cepstrum.

first time-derivative (Δ) and the second time-derivative (ΔΔ) of the cepstrum as well. (Log) energy of the analysis frames also provides important information. In order to reduce the sensitivity to unwanted conditions such as the recording level, the frame energy is normalised within each sentence utterance (Young et al., 1994). Also the Δ energy and ΔΔ energy are used.

The WVC of the basic MFCC is much smaller[9] than that of the filterbank. This will be shown (indirectly) in Sub-section 3.5.2. We will use a single data stream instead of putting the groups of basic MFCC parameters and their Δ's in multiple streams, to avoid facing the "independence assumption" (see page 31 for similar situation for the codebooks in DDHMM). In such cases, correlation between groups of parameters may not be small, and can cause similar problems in modelling as for the filterbank parameters. One would then hope to remove the correlation in the speech data by using a linear transformation before modelling it with diagonal covariance matrices in a CDHMM system.

In this section we will apply PCA to remove the WVC correlation. In the first subsection we will analyse the correlation of the cepstrum coefficients. Next we will introduce particular methods of linear transformations in both training and recognition algorithms. Then we will show the impact of these transformations on the recognition performance.

3.5.1 Reduction of HMM parameters

We will use 12 MFCC, the log energy, and their Δ and ΔΔ, together making a 39-dimensional vector (the particular way to derive the Δ's from the basic components is given in Appendix 3.2). We will use linear phone models with different lengths ranging from 3 to 10 selfloops (necessary for durational constraints; the details will be discussed in Chapter 5). These two factors make the total number of HMM parameters rather large, which may cause an under-training problem even with the moderate-sized TIMIT database. We will do two *a priori* parameter reductions at design time.

The first reduction is based on an observation that some of the 61 original TIMIT phones occur very rarely. These are mostly different allophones. Therefore we merge some of the similar phones into one single phone. The same has been used by Lee & Hon (1989) and Young & Woodland (1994), also for TIMIT. Our set of merged phones is a little different from theirs because we use the later version of TIMIT (see the documentation on the CD-ROM, 1990). The total number of phones becomes 50 and is shown in Table 3.9. After such merging, each phone appears at least a few hundred times.

---

[9]The MFCC are defined on an orthogonal basis (exact cosine transformation, equation 1) as calculated from filterbank energy outputs of a single frame. Correlation is defined on a (large) number of frames, therefore being data-dependent. Thus the correlation between MFCC parameters is, in general, not zero.

Table 3.9 Comparison table of 61 old and 50 new phone symbols before and after the merging (and renaming) of TIMIT phones. The symbols are printed in ASCII (also called TIMITBET) instead of IPA (International Phonetic Alphabet). The graphemes in the example words corresponding to the phones are printed in bold. The blank entries for "old phones" indicate that no merging occurs for these phones. All the segments with the old phone label are made equivalent to the corresponding new phone, while the old phone symbols are removed from the phone inventory.

new phone	old phone	example	new phone	old phone	example
iy		b**ea**t	ng	eng	si**ng**
ih		b**i**t	ch		**ch**urch
eh		b**e**t	jh		**j**udge
ae		b**a**t	dh		**th**ey
ix		ros**e**s	b		**b**ob
ax	ax-h	th**e**	d		**d**ad
ah		b**u**tt	dx		bu**tt**er
uw	ux	b**oo**t	g		**g**ag
uh		b**oo**k	p		**p**op
ao		ab**ou**t	t		**t**ot
aa		c**o**t	k		**k**ick
ey		b**ai**t	z		**z**oo
ay		b**i**te	zh		mea**s**ure
oy		b**oy**	v		**v**ery
aw		b**ough**	f		**f**ief
ow		b**oa**t	th		**th**ief
l		**l**ed	s		**s**is
el		bott**le**	sh		**sh**oe
r		**r**ed	hh	hv	**h**ay
y		**y**et	cl	pcl, tcl, kcl	(unvoiced closure)
w		**w**et	vcl	bcl, dcl, gcl	(voiced closure)
er	axr	b**ir**d	epi		(epinthetic closure)
m	em	**m**om	q		(ba**t**, glottal stop)
n	nx	**n**on	pau		(pause)
en		butt**on**	ns	h#	(non-speech)

The second parameter reduction is based on the consideration that for each phone, the number of states that have *different* observation pdf's need not be very large. We consider 3 to be sufficient, of which the middle one models the steady part of the phone and the other two model the beginning and end transitions, respectively. The total number of states of each phone is

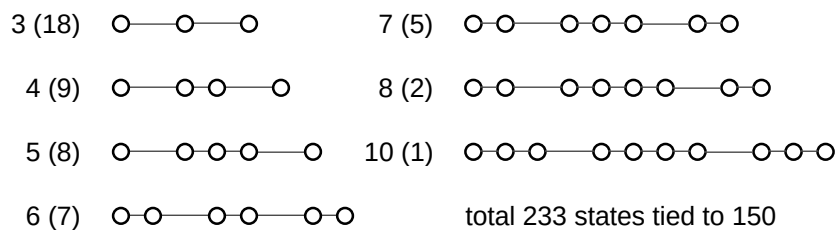


Figure 3.2 For HMMs with different number of states, tying of the states are shown with circles that are put close together. The actual number of such HMMs is given between parentheses.

still determined by the durational constraints (Chapter 5), while some of them will be "tied" together, i.e., they share the same observation parameters[10]. Therefore the duration will be well modelled with a suitable model length and a suitable set of transition parameters, while the acoustic modelling is simplified. The actual sub-sets of HMM states to be tied together are chosen in such a way that the beginning and the end parts have the same length, and these two transition parts are not longer than the middle part. This is schematically shown in Figure 3.2. After such state tying, the total number of states with different pdf in the whole system becomes 150 (as compared to 233 before tying).

3.5.2 Analysis of cepstrum correlation in HMM states

First of all we investigate the correlation between the cepstrum components as represented in the states of a well trained set of HMMs. We calculate the covariance matrices $\boldsymbol{S}_i$ of all the 150 states i, and then normalise them into their matrices $\boldsymbol{R}_i$ of correlation coefficients[11] using (4). In order to get an overview in the correlation between the different components of $\boldsymbol{o}$ in the whole system, we made a plot containing $p \times p$ elements ($p = 39$) (Figure 3.3). Each element at the location (l,m) is a square-wave- like pulse, which has a height proportional to the averaged correlation (either positive or negative) between components l and m over all state i:

$$\bar{r}_{lm} = \frac{1}{N}\sum_{i=1}^{N} r_{lmi},$$

where $N = 150$ is the number of states. The width of the r_{lm} pulse at (l,m) is proportional to the standard deviation of the correlation over all the states:

$$\sigma_{r_{lm}} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(r_{lmi} - \bar{r}_{lm})^2}.$$

For the elements on the main diagonal (i.e., those for r_{ll}), $\sigma_r = 0$ since for all states i $r_{lli} = 1$. In order to make this still visible for this way of plotting, the breadth is set to a minimal value so we see sharp peaks with the maximum height of 1 on the main diagonal.

Besides the main diagonal, we see that the only large correlation occurs systematically between each basic cepstrum parameter c and its corresponding $\Delta\Delta c$. We will show in Appendix 3.2 that the existence of this correlation and the absence of the correlation between c and Δc is due to the particular way of calculating the Δ parameters. Other noticeable irregular

[10]HTK supports a powerful mechanism called *generalised tying*, with which nearly any structure of the HMM system at the same level can be tied.
[11]The special way of calculating $\boldsymbol{S}_i$ in this study will be presented in Subsection 3.5.3.

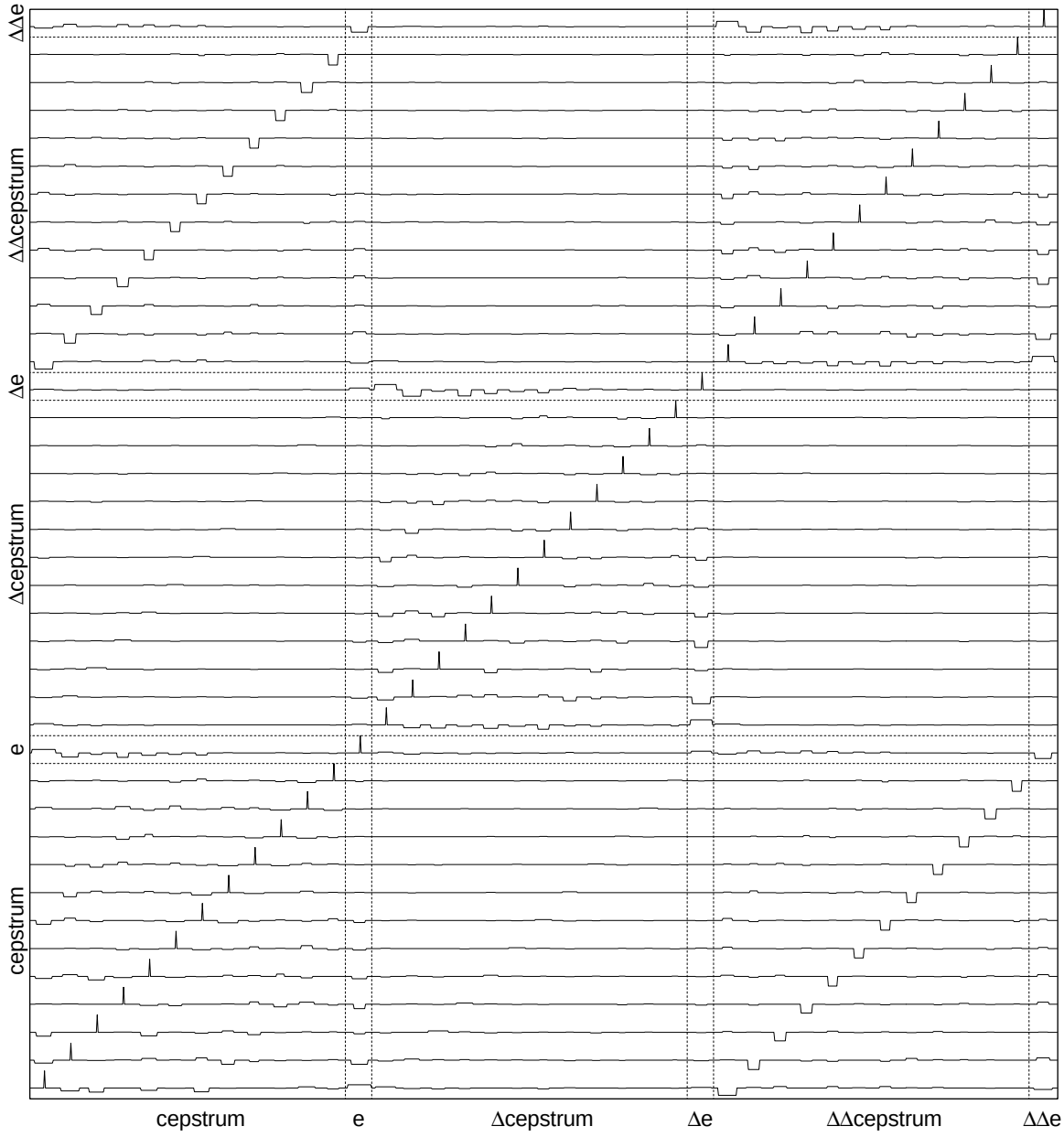


Figure 3.3 Average and standard deviation of the correlation coefficients between the 39 components of the observation vector. The components are arranged from left to right and from bottom to top: 12 cepstra, energy, 12 Δ cepstra, Δ energy, 12 ΔΔ cepstra and ΔΔ energy.

correlations have to do with the way of state-assignment of the $\{\mathbf{o}\}$, and will be discussed in the next subsection.

It has to be noted that what this plot shows is only the normalised covariance that varies over the states, whereas the means of all the states are not shown because the covariance is calculated *around* the mean of each state. In the recognition process, both state mean and state covariance play an important role in providing the discrimination power. In this overall plot one can only see the part of discrimination power provided by the difference of

the state covariance *across* the states. A wide pulse in an element indicates a large spread of the covariance across the states.

Furthermore, it must be noticed that the correlation here is analysed over the 150 HMM states, instead of over the data vectors of the whole training set. If, however, the HMMs are well trained with the data (next subsection), such correlation indirectly shows the situation in the data.

3.5.3 Removing correlation

The recognition results in Section 3.4 with filterbank parameters and in Ljolje (1994b) with a transformation defined globally on all the observation vectors did not show improvement. In this section, we will perform both *global* and *state-specific* transformations. With a linear transformation $\boldsymbol{y}_t = \boldsymbol{V}^{\mathrm{T}}\boldsymbol{o}_t$, the observation vector $\boldsymbol{o}_t$ will be de-correlated into $\boldsymbol{y}_t$, whereas the transformation $\boldsymbol{V}$ is obtained in the same way as in Section 3.3. The only difference is that, for state-specific transformation $\boldsymbol{V}_i$, we have many sample covariance matrices $\boldsymbol{S}_i$, each specific for a state i. Only the subset $\{\boldsymbol{o}\}_i$ of the whole training set $\{\boldsymbol{o}\}$ should be used to calculate $\boldsymbol{S}_i$, where those $\boldsymbol{o}_t$ in $\{\boldsymbol{o}\}_i$ are "assigned" to state i. However, the assignment of $\boldsymbol{o}_t$ to all the states is unknown in the usual training and recognition processes, thus there can be different ways to get an *approximated* assignment.

One state assignment can be obtained using a Viterbi back-tracking with a set of trained models on the training data (Ljolje, 1994b). Then, only those $\boldsymbol{o}_t$ that are assigned to a state i are used to calculate $\boldsymbol{S}_i$. However, in this study, we use a different approach, which is based on the following observation. Since our recognition task is for continuously spoken sentences, fixed state assignments obtained by Viterbi back-tracking on the training data may not be optimal. Actually the final phase of the training process involves a so-called embedded re-estimation, in which even the temporal assignment at the word and phone levels in the training labels are not explicitly used, let alone the state level assignment. This can be further explained with the re-estimation formulae for, e.g., the mean of the Gaussian pdf. The mean, updated with the embedded re-estimation, when state assignment is unknown, is

$$\overline{\mu}_i = \frac{\sum_{t=1}^{T} \gamma_t(i)\boldsymbol{o}_t}{\sum_{t=1}^{T} \gamma_t(i)},$$

where $\gamma_t(i)$ is the state occupation likelihood (a simple special case of Appendix 2.1), and time t goes through all the training samples. $\gamma_t(i)$ for an $\boldsymbol{o}_t$ for a different state $j \neq i$ can be low but generally not zero. On the other hand, when the state assignment is known (e.g., in a Viterbi-training procedure), the mean is just the sample mean

$$\hat{\boldsymbol{\mu}}_i = \frac{1}{M} \sum_{\boldsymbol{o}_t \in \{\boldsymbol{o}\}_i} \boldsymbol{o}_t ,$$

where M is the number of $\boldsymbol{o}_t$ in $\{\boldsymbol{o}\}_i$ that belong to state i. It can be seen that this formula is a special case of the previous one when the likelihood of the "wrong" state occupation is set exactly zero. We argue that the first embedded formula can lead to a better estimate, because this is how the system of HMMs sees which $\boldsymbol{o}_t$ belongs to which state (with a certain likelihood).

In the actual procedure, the first step is to initialise and re-estimate all the parameters of the HMMs with sufficient iterations (we used 5 iterations of embedded training after initialisation training). Each state i of each of such an HMM has a single Gaussian density with a *full* covariance matrix $\boldsymbol{\Sigma}_i$. Although $\boldsymbol{\Sigma}_i$ after the embedded training is not exactly the same as the sample covariance matrix $\boldsymbol{S}_i$, they are close to each other. Then we use this approximated $\tilde{\boldsymbol{S}}_i = \Sigma_i$ to calculate $\boldsymbol{R}_i$ (using (4)). Then, state-specific transformation matrices $\boldsymbol{V}_i$ with the eigenvectors of $\tilde{\boldsymbol{S}}_i$ are made. Thereafter, these $\boldsymbol{V}_i$ are used for further initialisation and re-estimation with transformations (each state has a diagonal variance matrix), starting again with a single Gaussian. During re-estimation the mixture components have been split (with HTK, see Young, 1992) gradually to 2, 4, and 8, respectively. After each splitting step, a single iteration is performed and after the last step, 4 iterations complete the training.

It is based on these $\boldsymbol{R}_i$ that Figure 3.3 is made. It can now be explained that the irregular but noticeable correlations are introduced by such relaxed method of obtaining the $\boldsymbol{R}_i$, which are nevertheless closer to the reality of the recognition process (see for a similar concept in Sun, 1996).

There is a question of how to use $\boldsymbol{V}_i$ to perform the transformation $\boldsymbol{y}_t = \boldsymbol{V}_i{}^{\tau} \boldsymbol{o}_t$, since one does not know to which state $\boldsymbol{o}_t$ belongs. However in practice this is not a problem since for both the training and recognition algorithms with each $\boldsymbol{y}_t$, what one needs is to calculate $b_i(\boldsymbol{y}_t)$, for which the identity of the state i is available.

It has to be noted that after transformation, all $\boldsymbol{S}_i$ become diagonal (and $\boldsymbol{R}_i$ identity matrix). If a plot similar to Figure 3.3 would be made for the diagonalised $\boldsymbol{S}_i$, all the entries except for the main diagonal, as the average over all the states, would be zero.

It may be interesting to discuss whether $b_i(\boldsymbol{y}_t)$ for a state with a $\boldsymbol{\Sigma}_{\boldsymbol{y}}$ diagonalised by the transformation $\boldsymbol{V}_i$ is exactly the same as $b_i(\boldsymbol{o}_t)$ for the same state with the original full $\boldsymbol{\Sigma}_{\boldsymbol{o}}$. This is indeed true if the exact state assignment is used, and a *single* Gaussian is used so that $\boldsymbol{\Sigma}_{\boldsymbol{o}} = \boldsymbol{S}_{\boldsymbol{o}}$. (For this discussion we drop the subscript i for the state index because we confine ourselves to a single state). In fact, since ($\boldsymbol{V}$ is orthogonal and normalised)

$$\mathbf{y}_t = \mathbf{V}^{\tau} \mathbf{o}_t , \mu_{\mathbf{y}} = \mathbf{V}^{\tau} \mu_{\mathbf{o}} , \Sigma_{\mathbf{y}} = \mathbf{V}^{\tau} \Sigma_{\mathbf{o}} \mathbf{V}, |\Sigma_{\mathbf{y}}| = |\Sigma_{\mathbf{o}}|,$$

we have

$$b(\mathbf{y}_t) = \frac{1}{\sqrt{(2\pi)^p \,|\Sigma_{\mathbf{y}}|}} \exp\left\{-\frac{1}{2}(\mathbf{y}_t - \mu_{\mathbf{y}})^\tau \Sigma_{\mathbf{y}}^{-1}(\mathbf{y}_t - \mu_{\mathbf{y}})\right\}$$
$$= \frac{1}{\sqrt{(2\pi)^p \,|\Sigma_{\mathbf{y}}|}} \exp\left\{-\frac{1}{2}(\mathbf{o}_t - \mu_{\mathbf{o}})^\tau \mathbf{V}\mathbf{V}^{-1}\Sigma_{\mathbf{o}}^{-1}(\mathbf{V}^\tau)^{-1}\mathbf{V}^\tau(\mathbf{o}_t - \mu_{\mathbf{o}})\right\}$$
$$= \frac{1}{\sqrt{(2\pi)^p \,|\Sigma_{\mathbf{o}}|}} \exp\left\{-\frac{1}{2}(\mathbf{o}_t - \mu_{\mathbf{o}})^\tau \Sigma_{\mathbf{o}}^{-1}(\mathbf{o}_t - \mu_{\mathbf{o}})\right\} = b(\mathbf{o}_t).$$

However, we gain the luxury of modelling at the same accuracy as with a full $\boldsymbol{\Sigma_o}$, but in fact use only a diagonal $\boldsymbol{\Sigma_y}$. In practice, since we usually use a number of Gaussian elements per state, and due to our special way of obtaining $\boldsymbol{S_o}$, the situation will be more complicated than this simplest one[12].

After transformation, since we need a smaller number of model parameters for the diagonal $\boldsymbol{\Sigma_y}$, the finite amount of training data will allow us to use a larger number of Gaussian elements, therefore we gain a better modelling accuracy.[13] The impact of the transformation on recognition performance will be discussed in Sub-section 3.5.5.

3.5.4 On dimensional reduction of observation vectors

The issue of dimensional reduction in the state-specific transformation in this section is different from that in a global transformation. Since it is unknown in advance to which state an observation vector $\boldsymbol{o}$ belongs, it has to be kept in its original full dimensional form. Each time when it encounters a state, this original $\boldsymbol{o}$ needs to be transformed. This means that the memory requirements are not reduced (e.g., each 39-dimensional input observation vector still occupies 39 locations for floating point values, even if after transformation and truncation only 19 components will be kept). The only savings after truncation would be found in the calculation of the output probability (fewer additions and multiplications in the exponential of the Gaussian pdf's).

It can be seen from the upper panel of Figure 3.4, that the statistics of the parameters of a set of HMMs trained *without* transformation show a similar pattern as those with transformation (lower panel), although somewhat less steep. This is due to the fact that the cepstrum parameters are obtained from

---

[12]In general, the actual *data* distribution will not be Gaussian, therefore it is insufficient to model the data with a single Gaussian (see an illustration in Appendix 3.1). However, if a single Gaussian is used (e.g., for small recognition tasks), the above formula still holds, i.e., the *inaccuracy* in modelling with $b(\boldsymbol{o}_t)$ is completely transferred to $b(\boldsymbol{y}_t)$.

[13]For 39-dimensional vectors, the full covariance matrix of a single Gaussian has $(1+39)\times 39/2 = 780$ parameters, while a pdf with 8 Gaussian elements each with a diagonal variance matrix needs only a total of $8\times 39 + 8 = 320$ parameters (including the mixture weights). Numbers of parameters in other parts of the model remain the same.

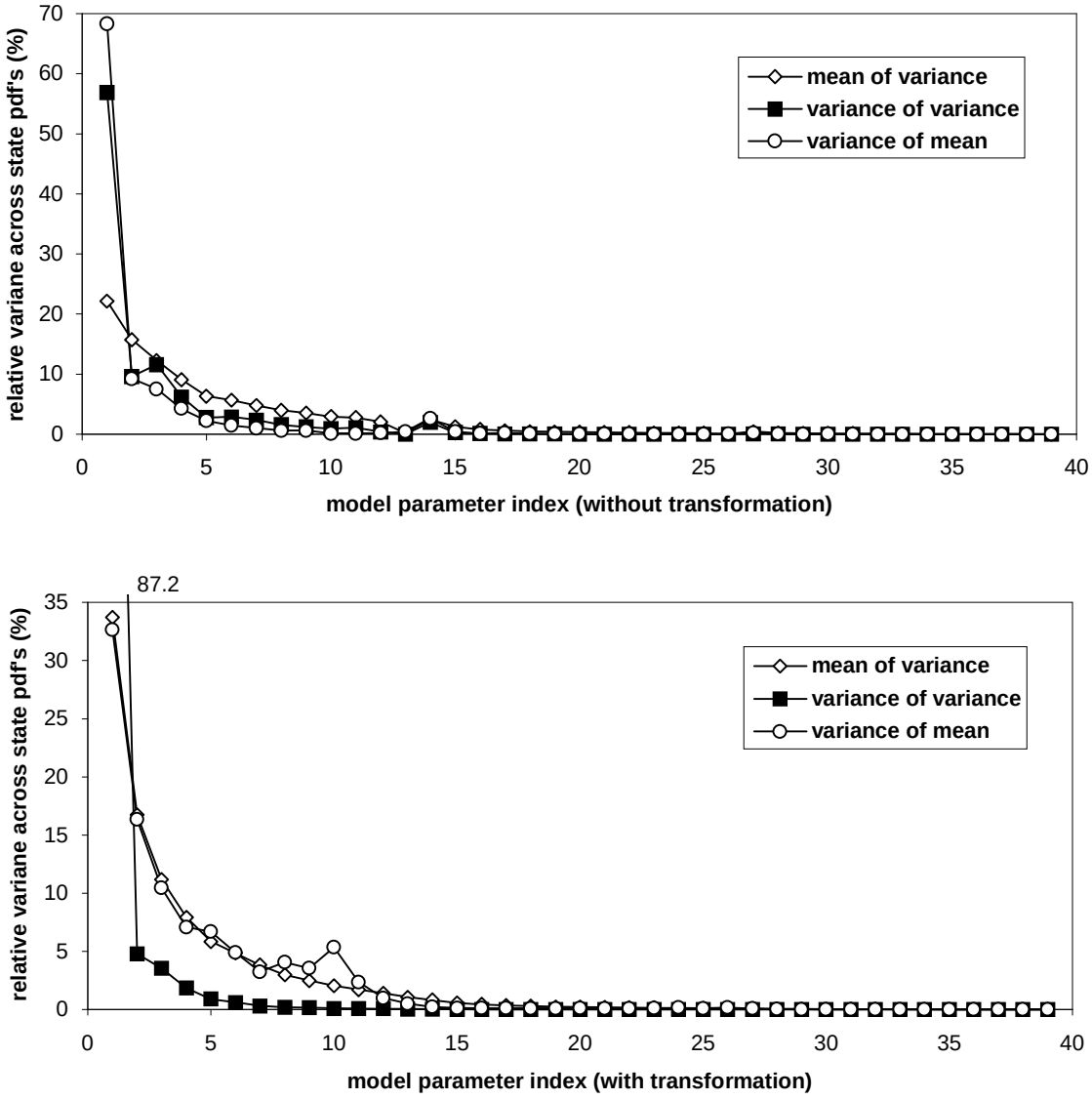


Figure 3.4 Variance distributions among the 39 model parameters without (upper panel) and with (lower panel) transformation, calculated across all the 150 state pdf's of all the models. Each curve is normalised to itself and displayed in percentages.

a cosine transformation from the filterbank energies (see formula (1) in Section 3.2); therefore they already have a similar distribution as the principal components. Such a phenomenon may reduce the difference in recognition performance with and without transformation (next subsection).

In order to investigate the relative importance of the components of the data vectors, we look at the statistics of the *parameters* of the HMMs in the recogniser trained using the state-specific transformation, instead of the statistics of the *data vectors*. The former should be a good representative of the latter, if the HMMs are well trained with the data. Furthermore, it is the parameters of the HMMs that provide the discrimination power in the recognition process. In an extreme case, if a particular parameter had exactly

the same value across all the HMMs, this parameter would have no use and could be eliminated from all the HMMs.

Since each state has a different mean vector, we would like to see if the small values of the variances in the high-indexed components for each state also cause that the variances *across* different states (and models) are negligibly small as well. This is shown in the lower panel of Figure 3.4 as *Variances of the state means* across all the 150 states of the system, which are calculated for all the 39 dimensions using the set of HMMs trained with state-specific transformation. The mean of each state is calculated as the weighted sum of the Gaussian means using the mixture weights. Another two measures on the distribution among the components are the *variance of the state variances* and an *mean of state variances*. It can be observed that, in components indexed higher than 15, various kinds of variances across the states are all rather small[14] but not zero. It will be shown in the next subsection whether these small values have significant contributions to recognition performance or not.

3.5.5 Impact of cepstral correlation removal on recognition

In the test of this section we used all the 3,696 TIMIT `sx` and `si` utterances (see Chapter 1) in the training set for training and the 1,344 `sx` and `si` utterances in the test set for testing. For testing purposes two (deterministic) word-pair grammars[15] of the regular type have been extracted, one from the test set only (2,373 different words, perplexity 1.32, see Chapter 2) and another from both training and test sets (6,100 different words, perplexity 1.48). Since the out-of-vocabulary (OOV) rate (proportion of words that appear in the testing set but not in the training set) is very high (51%), a grammar with only the training set would cause a very low recognition score, although that may be an "unbiased" one. The high OOV of TIMIT is due to the design of the training and the test sets which guarantees that no sentence texts overlap across the two sets (thus there is also little word-overlap). This design also guarantees a full use of all the material. Other partition of the two sets may reduce the OOV somehow, but then the usable part of the material will also be reduced if one still wants the non-overlap condition.

The pre-processing results in 12 basic MFCCs plus the normalised frame energy. Their Δ and $\Delta\Delta$ are augmented in various test conditions (in most

[14]After transformation, although the general trend of the *variance of means* is also decreasing, it shows irregularities in its pattern. The variance itself shows a strictly decreasing pattern because the eigenvectors of all the states are arranged according to decreasing variance. The state means, however, are generally rather *independent* from the state variances. Therefore, after HMM training, they may have values which give rise to an irregular variance distribution among the components.

[15]HTK (version 1.4) also supports a bigram grammar but its internal data structure will cause memory problems on our Silicon Graphics workstations for a vocabulary size above 5,000.

conditions they are both used). The training procedure resulted in three sets of HMMs for different data vectors, i.e., the original, the globally transformed, and the state-specifically transformed, respectively.

For each word, the allowed sequence of phones (pronunciation) is basically linear (extracted from the TIMIT lexicon) plus an optional pause at the end of each word, and additional closure phones before the burst parts of plosives when they do not immediately follow a silence. However, in the actual pronunciation of each sentence, there exist severe deviations of the phones from such a norm sequence, because of, e.g., substitution by allophones or deletion of phones. The use of a single pronunciation will certainly cause all the scores to be lower than if one had multiple pronunciations for all the words. But such a multiple pronunciation is not available for TIMIT (in forms of either explicit listing or phonological rules). Nevertheless, our goal is only to compare the relative performance under different transformations and amounts of dimensional reduction, instead of to reach a maximal performance. Therefore, a single pronunciation is acceptable here.

The language match factor (LMF) is chosen for each condition in such a way that the word accuracy is optimised, i.e., the total error rates including both insertion and deletion are minimised[16]. We tested on two conditions of dimensional reduction for both the global and the state-specific transformations, at 2/3 of the full dimensionality (26) and at 1/2 (19), respectively. The word recognition scores are shown in Table 3.10.

We also performed phone recognition to provide a more direct comparison of the scores with those published in the literature, because most authors use the TIMIT database only for phone recognition tests (Lee & Hon, 1989; Ljolje 1994a; Lamel & Gauvain, 1993b; Young & Woodland, 1994). Phone recognition relies almost exclusively on the acoustic modelling since there is very little contribution from the language modelling as in the case of word recognition. Therefore it will generally show a more direct impact of the pre-processing quality. In our phone recognition tests, the same three sets of HMMs as in the corresponding conditions for word recognition are used, therefore no extra training was needed. The "grammar" used allows all between-phone transitions with bigram probabilities estimated from the training set alone (there is no "OOV" problem since all between-phone transitions occurring in the test set also occur in the training set). The scores are shown in Table 3.11. Our phone scores are comparable with the best reported results using context-free (monophone) models (Table 3.12). For scoring purposes only, the 50 phones are further folded down to 39 symbols, exactly the same as used in Lee & Hon (1989) and Young & Woodland (1994).

[16]Because actual values of the acoustic probabilities for different conditions are different due to different transformations and truncations, a separate optimisation of LMF for each condition is necessary. However, the optimisation was only performed upto a certain limit since it would be impractical to test all possible values of the LMF for each condition.

Table 3.10 Word recognition scores with MFCC for different dimensionality and the two word-pair grammars ("test-set" and "both-set"). "Original" refers to the case without transformation, whereas "global" and "specific" refer to the two ways of transformation. The original at the dimensionality of 26 is achieved using only the basic and the first derivative parameters. In all the test conditions for this table, 8 Gaussian mixture elements per state are used with a diagonal covariance matrix. "Correct" scores include only the deletion and substitution errors, whereas "accuracy" scores also include insertion errors.

Dimensionality		39		26		19	
Word scores (%)		correct	accuracy	correct	accuracy	correct	accuracy
Test-set Word-Pair	original	91.72	90.71	90.22	89.10	-	-
	global	91.63	90.59	92.25	90.98	90.78	89.18
	specific	91.44	90.18	90.92	89.62	89.96	88.54
Both-set Word-Pair	original	84.17	82.51	81.12	79.44	-	-
	global	83.26	81.64	82.66	80.54	80.72	78.43
	specific	83.79	81.90	81.62	79.69	79.46	77.58

Table 3.11 Same as Table 3.10, but for phone recognition scores with a phone bigram grammar.

Dimensionality		39		26		19	
Phone scores (%)		correct	accuracy	correct	accuracy	correct	accuracy
Train-set Phone Bigram	original	70.74	66.02	69.08	64.14	-	-
	global	69.46	64.66	68.20	62.74	66.66	60.94
	specific	72.07	66.85	68.46	62.51	65.77	58.92

Some general comparisons between the scores of different test conditions

The phone scores in the tables are obtained by comparing the recognised phone sequences with the exact phone labels provided by TIMIT. For a database without phone labelling, one usually can only use phone labels generated from the norm pronunciation of each lexical word. In order to make our scores comparable with such situations, we also generated such norm labels based on the TIMIT lexicon. Then the same phone recognition results were scored with such norm labels, for two example conditions of "original" and "specific" at the full dimensionality of 39 (not shown in tables). The accuracy scores were then reduced for the two conditions from 66.02% down to 55.98% and from 66.85% down to 56.72%, respectively.

al comparisons between the scores of different test conditions in Table 3.10, 3.11 and 3.13 are summarised as follows.

1. **Transformation has little effect.** From the scores in Table 3.10 and 3.11 (diagonal matrices) shows that neither kind of transformation changes the scores significantly. This indicates that the correlation in MFCC is not very severe. Furthermore, using a large number (of 8) Gaussians (as is often used in the state-of-the-art ASR systems), the modelling accuracy does not change much with a linear transformation (Appendix 3.1).
2. **Data distribution is not Gaussian.** The fact that the word accuracy score in Table 3.13 with 8 diagonal-Gaussians (90.71%) is higher than with

one diagonal-Gaussian (80.35%) indicates that the actual data distribution in the MFCC space (for TIMIT) is more complicated than just a single Gaussian, being either correlated or not. The fact that the score with two diagonal-Gaussians (86.18%) is already higher than with one full-Gaussian (82.62%) further indicates that the data distribution is better modelled with a two-cluster shape than with a single cluster with a shape of an oblique hyper-ellipse (Appendix 3.1).

Table 3.12 Phone recognition scores for some systems and our own using TIMIT. Entries with '-' were not performed. The acoustic model used in each system can be either context-dependent or context-free, and the language model can be either (phone) bigram or trigram.

acoustic model	context-free		context-dependent		language
phone scores (%)	correct	accuracy	correct	accuracy	model
Lee & Hon (1989)	64.07	53.28	73.80	66.08	bigram
Lamel & Gauvain (1993b)	-	-	77.5	72.9	bigram
Lamel & Gauvain (1993b)	-	-	78.3	73.4	trigram
Ljolje (1994a)	-	-	74.8	69.4	trigram
Young & Woodland (1994)	~71	66	76.7	72.3	bigram
our best (state-specific,1995)	72.07	66.85	-	-	bigram

Table 3.13 Word and phone recognition scores at full dimensionality (39) of MFCC without transformation, using 1 or 2 Gaussian mixture elements per state with a diagonal covariance matrix, and using 1 mixture element with a full covariance matrix. Scores for 8-mixture diagonal cases from previous tables are shown again for comparison.

Grammar	test-set word pair		both-set word pair		train-set phone bigram	
Scores (%)	correct	accuracy	correct	accuracy	correct	accuracy
1-mix diag.	82.88	80.35	71.17	67.63	63.30	57.95
2-mix diag.	87.85	86.18	77.73	75.27	65.99	60.77
8-mix diag.	91.72	90.71	84.17	82.51	70.74	66.02
1-mix full	84.43	82.62	75.11	72.60	70.01	65.15

3. **Different impact between phone and word scores.** When we appreciate[17] the small differences in scores between various transformations, we can draw a different conclusion about the impact of the transformation on word recognition and phone recognition. For word recognition (Table 3.10), both the global and the state-specific transformation reduce the score slightly (from 90.71% down to 90.59% and 90.18%, respectively). For phone recognition (Table 3.11), the situation is different: The state-specific transformation improves the score (from 66.02% to 66.85%) while the global transformation decreases the score (down to 64.66%). A possible interpretation for this difference in behaviour between phone and word recognition is as follows. In phone recognition, the optimal modelling with the state-specific transformation is easily propagated to the phone level, thus the score improves. In word recognition, on the other hand, the strong constraints provided by the language models (LM) may cause the locally optimised phones (with state-specific transformation) to be more difficult to concatenate into the correct words, than would be the case without transformation, thus degrading the performance. The global transformation degrades the modelling accuracy at the local state or phone levels, thus the scores go down for both phone and word recognition. The decrease in scores is smaller for

[17]If we perform statistical tests (χ^2), these changes in scores all fall within a 95% confidence interval given the number of items tested (instances of words or phones). However, in speech recognition literature reporting the original work on recognisers, the confidence interval is rarely looked at while any (small) improvement in scores is reported.

words than for phones, because the strong LM constraints in word recognition compensate this score decrease.

4. **Different degree of under-training for phones and for words.** As discussed earlier, using 8 diagonal-Gaussians will provide a better modelling scheme for our specific amount of training data than using one full-Gaussian (the latter modelling is rarely seen in the literature). However, the difference in word accuracy scores (Table 3.13) is more tangible for word (90.71% vs. 82.62%) than for phone (only 66.02% vs. 65.15%). Part of the speech information *associated* with the higher level structures is not well integrated into the (monophone) HMMs with one full-Gaussian, for which more severe under-training exists, than for the HMMs with 8 diagonal-Gaussians. Such information is required especially for the more complicated task of word recognition, therefore any lack of it reduces the word score. Such an under-training does not reveal itself in phone recognition because the task does not require much higher level information. Therefore, *a measure of the quality of the acoustic HMMs obtained from a phone recognition test is in general not directly transferable to word recognition.*

5. **Transformation provides possible dimensionality reduction.** At 2/3 of the original dimensionality, 26, the performance does not degrade significantly (e.g., word accuracy scores in Table 3.10, from 82.51% for 39-dimensions down to 80.54% for 26). Actually, we also performed a test with only the basic and the Δ, but without the $\Delta\Delta$. At this same dimensionality of 26, the (globally) transformed system performs slightly better (80.54%) than the original one (79.44%). Therefore transformations can be useful for applications where dimensionality reduction is desired. (Phone scores show just the opposite behaviour for the 26-dimension case: e.g., the score with the state-specific transformed system is 62.51% and the original is 64.14%. The only interpretation is that the $\Delta\Delta$ is not needed for phone recognition).

3.6 Conclusion

In this chapter the pre-processing part of the recogniser is optimised via different kinds of linear transformations on the observation vectors. Two kinds of observation vectors are involved: filterbank and cepstrum (MFCC), and two different recognisers are used: a DDHMM and a CDHMM. The overall conclusions are:

1. The MFCC seems to perform better than the filterbank (although no direct comparison was made);
2. The "global" transformation is neither beneficial for filterbank nor for MFCC;
3. For word recognition using MFCC, using "original" vectors without transformation is the best;

4. For phone recognition using MFCC, using "state-specific" transformation leads to best scores.

In addition, it is known that a CDHMM system generally performs better than a DDHMM system. All these concrete conclusions about the system setups and transformations will be useful for the later chapters, as a guideline for a best baseline performance chosen for each specific task. In general, correlation removal provides a method of dimensionality reduction with slight decrease in performance, useful for applications requiring the system to be compact. The best baseline system for this study will be the MFCCs without truncation. Furthermore, for situations close to phone recognition, the state-specific transform will be the best. For word recognition, the original MFCC is simply the best. These results are all achieved with 8 Gaussians per state.

In addition to the concrete (technical) specifications for the best baseline performance as obtained in this chapter, we can also draw conclusions about general practice in this chapter. As stated in Chapter 2, various system components in a recogniser have interactions, which makes the attempt to improve (or optimise) the performance a difficult task. Therefore we tried to have some (indirect) measure other than the overall performance scores of the system, which provides us with guidelines to improve the system. In this chapter, the correlation between the components in the observation vectors has been chosen as such a measure. In fact, the *hypothesis* was that reduction of such a correlation can improve the system performance.

The detailed technical development in this chapter shows that this hypothesis is not always true, and is different for the various system setups. It is only after the development and tests on different conditions that this can be verified, and detailed answers can be given for specific situations. The different experiences for the various setups have been presented in the discussions embedded in the sections of this chapter. Some general points will be presented here as follows.

First of all, one has to know the reason why the existence of correlation is unwanted. This is analysed with the help of dividing the correlation into two types, i.e., within-vector (WVC) and between-vector (BVC) in Section 3.4 for filterbank parameters. For a DDHMM system, only BVC is harmful when multiple codebooks are used. For CDHMM system, WVC as well is unwanted if one uses a simplified Gaussian pdf with a diagonal covariance matrix. Some kinds of correlation removal are impossible with linear transformations due to technical limits. Also taken into account is the joint effect with VQ distortion, that plays a role in a DDHMM system.

For the CDHMM system in Section 3.5 using MFCC, a more detailed analysis of correlation is presented, and the attempt there actually aims at making the best baseline system for this study. Correlation removal is applied on the composed vectors with the MFCC and the Δ's, and an additional state-specific transformation is performed. A general outcome is

that no transformation changes the recognition scores substantially, when 8 Gaussians per state are used. For complicated data distributions (e.g., in Appendix 3.1), the benefit of the correlation-removal-based approach is limited. Using a larger number of Gaussians in a mixture density is a better solution than transformations.

It is also shown that the actual amount of correlation is dependent on the kind of pre-processing parameters used in a recogniser. Therefore, removal of correlation in different kinds of parameters influences the recognition performance to different extents. The filterbank parameters have a larger correlation, but the performance with them is generally worse. The cepstrum parameters give better recognition performance. However, the correlation is rather small between them (indirectly shown in Pols (1977)), giving rise to very tiny changes in performances with or without the de-correlation transformation. Actually, the only significant correlation is between the basic and ΔΔ cepstrum parameters. The use of these additional ΔΔ parameters gives some improvement, but not very much (differences between the entries for 39-dimensional and 26-dimensional for "original" in Tables 3.10 and 3.11). This means that the total useful information provided by the ΔΔ parameters alone is relatively small. Therefore, whether this part of the information is well modelled (with transformation) or less well modelled (without transformation) will lead to little difference in the overall performance, for which modelling of other parts of the information plays a more substantial role.

Appendix 3.1 Gaussian-mixture distributions

In this appendix we illustrate different data distributions with 2-dimensional Gaussian pdf's (Figure 3.5), and discuss the possibility to model them accurately in each situation. In the legend for each of the six panels, ρ is used to quantify the correlation between the two dimensions (x and y), as in the covariance matrix

$$\Sigma = \begin{pmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{pmatrix}.$$

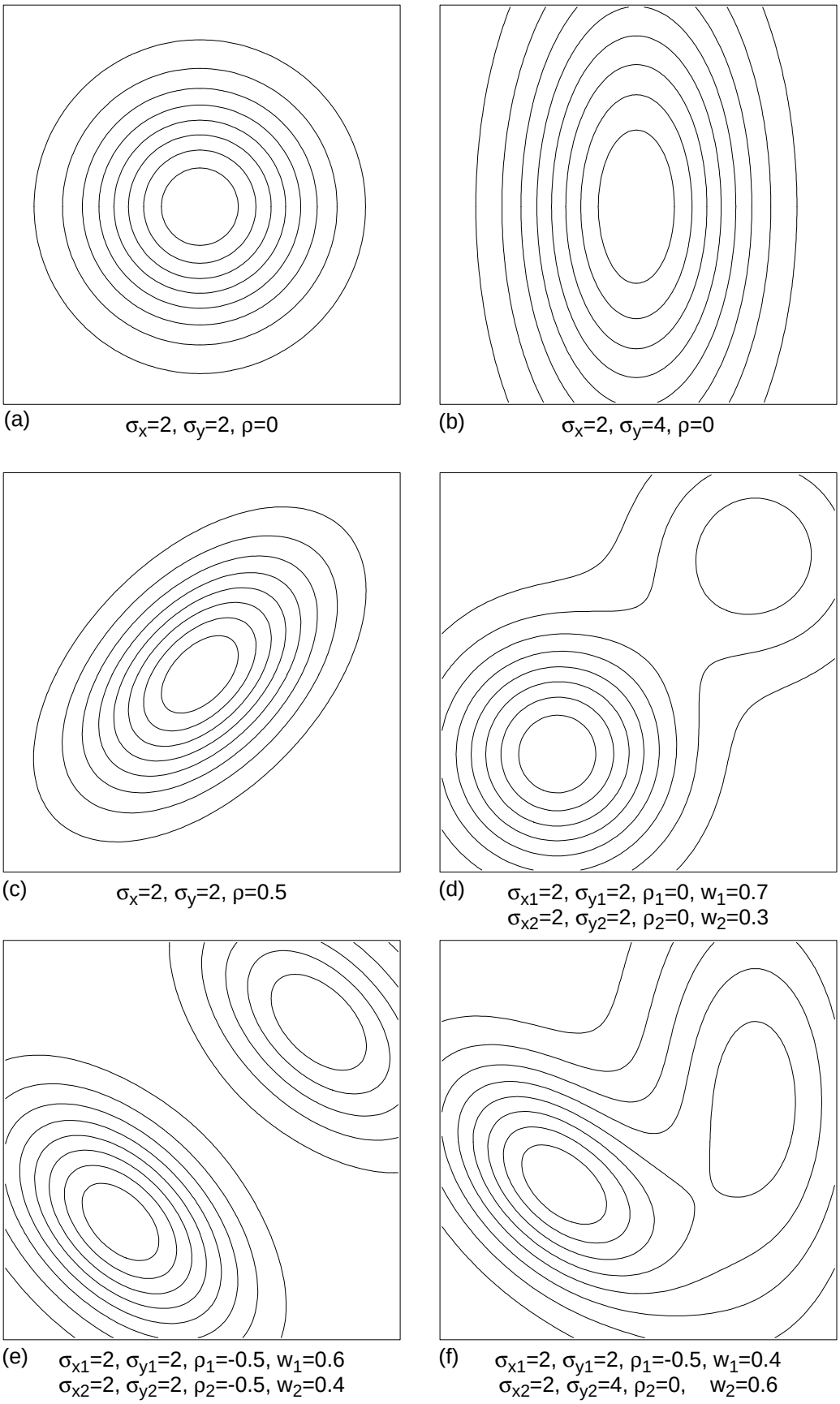


Figure 3.5 Grey-scale illustration of various 2-dimensional Gaussian pdf's (see text). In panels (d-f), "*w*" are the mixture weights for the Gaussians, and the "1" and "2" in the subscripts of ρ, σ and *w* refer to the first (bottom-left) and the second Gaussian.

When $\rho = 0$, Σ is diagonal, otherwise, it is "full". The means μ for all Gaussians are omitted from the legend, since they only determine the relative positions of the Gaussian centres. A general point to start with is that a Gaussian pdf with correlation ($\rho \neq 0$) can model an ellipse-like cluster *oblique* to the axes of the space.

In Figure 3.5, both (a) and (b) show a single Gaussian-like distribution without correlation between x and y; (b) has a larger variance in y than in x. For both these situations a single Gaussian with a diagonal covariance matrix (a "diagonal-Gaussian" for short) will be sufficient. The data in (c) contains correlation between x and y thus requires a single full-Gaussian. If, however, a linear transformation is applied to (c), the new axes will be along the two main axes of the ellipse, thus a single diagonal-Gaussian in the transformed space will model the data accurately. Both (d) and (e) show data with two clusters of high density. (d) can be well modelled with a mixture of two diagonal-Gaussians. (e) can be accurately modelled either with two full-Gaussians in the original space, or with two diagonal-Gaussians in the transformed space. This is because the axes of the two ellipses are parallel, which make it possible for the axes of the transformed space to be along these axes. For this kind of data distribution, linear transformation indeed improves the modelling accuracy with simplified models. The last situation (f) is the most general, for which no transformation can be found that "rotates" the axes to such directions that all Gaussians used can be diagonal.

It must be noted that, in general, a data distribution with a shape of an ellipse with non-equal axes can be approached with two Gaussians located close to each other. For such cases, even if the axes of the ellipse do not parallel the coordinates, two diagonal-Gaussians will model the whole ellipse well, without transformation. In the original space for (f), either two full-Gaussians (one for each oblique ellipse) or four diagonal-Gaussians will be

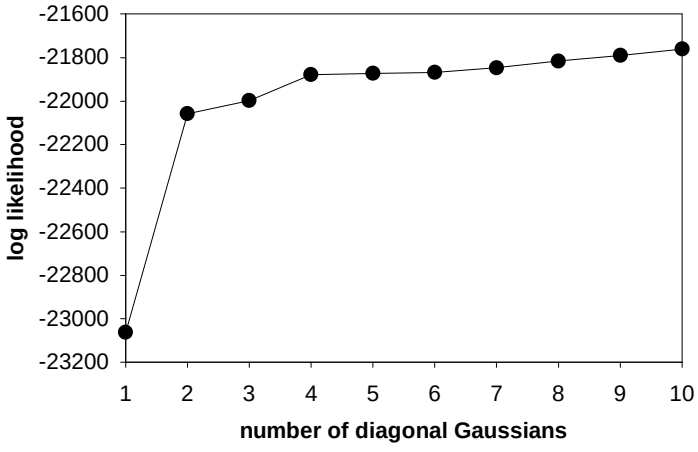


Figure 3.6 Log likelihood of the 5,000 data points given the mixture density with various number of Gaussians (see text).

sufficient. If the space for (f) is transformed (e.g., by PCA), since the new axes are not along any axes of the ellipses, still two full-Gaussians or four diagonal-Gaussians will be needed for a similar accuracy in modelling. This explains why transformation does not help for general data distributions.

The situation of approaching each (oblique) ellipse-like Gaussian by two diagonal-Gaussians is further illustrated with 5,000 simulated data points, randomly distributed according to the pdf in Figure 3.5 (f). Various numbers of diagonal-Gaussians are used to approach the total distribution. A maximum-likelihood method is used to iteratively estimate the parameters of the mixture density of the diagonal-Gaussians. The (log) likelihood of "observing" the data points indeed grows with the number of Gaussians (Figure 3.6). This illustrates the well-known fact that larger number of Gaussians in the mixture is always better. However, after 4 diagonal Gaussians (used to approximate the 2 full-Gaussians), the growth in likelihood becomes slower.

Appendix 3.2 Correlation due to delta parameter calculation

In this appendix we do not look at the intrinsic correlation between *different* components within a basic vector, and leave that as it is. We only look at the correlation between the basic parameters and its Δ and $\Delta\Delta$ parameters with the *same* index, due to the particular way these Δ are calculated from the basic parameters. Therefore in all the formulae (except for the last numerical example) we do not use the index for the elements, but only a time index. Furthermore, since the discussion here is totally independent of the actual kind of signal processing, the result will be general for both the filterbank and the cepstral parameters and any other parameters (although we use the notation c that is usually used for cepstrum).

In all the tests in this thesis, using the continuous density recogniser built on HTK, the Δc are calculated from

$$\Delta c_t = \frac{1}{K}\sum_{\tau=1}^{2} \tau \cdot (c_{t+\tau} - c_{t-\tau}), \tag{5}$$

where $K = \sum_{\tau=-2}^{2} \tau^2 = 2\sum_{\tau=1}^{2} \tau^2 = 10$ for this case, to make Δc a regression coefficient. It can be seen that 4 terms of c appear in Δc, but none of them are at the same time t as c. Therefore, the covariance between a c and its corresponding Δc does not contain an element of a covariance of c to itself. The covariance between c_t and another realisation $c_{t+\tau}$ at time $t+\tau$ reads

$$s(\tau) = \frac{1}{T-\tau}\sum_{t=1}^{T-\tau} (c_t - \mu_{c_t})(c_{t+\tau} - \mu_{c_{t+\tau}}),$$

and may not be zero, especially for small τ. However, due to the symmetric form of (5) for τ, $s(\tau)$ will equal $s(-\tau)$ if we omit the effect at edges of the data set (T is large), so that all μ_c are considered the same:

$$\begin{aligned} s(-\tau) &= \frac{1}{T-\tau}\sum_{t=\tau+1}^{T}(c_t-\mu_c)(c_{t-\tau}-\mu_c) = \frac{1}{T-\tau}\sum_{t'=1}^{T-\tau}(c_{t'+\tau}-\mu_c)(c_{t'}-\mu_c) \\ &= \frac{1}{T-\tau}\sum_{t'=1}^{T-\tau}(c_{t'}-\mu_c)(c_{t'+\tau}-\mu_c) = s(\tau), \end{aligned} \tag{6}$$

and $s(\tau)-s(-\tau)=0$. Therefore, the correlation between c and Δc tends to zero. Since the ΔΔ parameters are calculated in the same way from Δ parameters

$$\Delta\Delta c_t = \frac{1}{K}\sum_{\tau=1}^{2}\tau\cdot(\Delta c_{t+\tau}-\Delta c_{t-\tau}),$$

due to the same symmetric form of (5), it also holds that the covariance between Δ and ΔΔ tends to zero.

However, the covariance between the basic and ΔΔ may not be small or zero. In fact,

$$\begin{aligned} \Delta\Delta c_t &= \frac{1}{K^2}\sum_{\tau=1}^{2}\tau\sum_{\tau'=1}^{2}\tau'\cdot\left(c_{t+\tau+\tau'}-c_{t+\tau-\tau'}-c_{t-\tau+\tau'}+c_{t-\tau-\tau'}\right) \\ &= \frac{1}{K^2}\left(4c_{t+4}+4c_{t+3}+c_{t+2}-4c_{t+1}-10c_t-4c_{t-1}+c_{t-2}+4c_{t-3}+4c_{t-4}\right). \end{aligned} \tag{7}$$

We can represent s between $\Delta\Delta c_t$ and c_t in terms of s between c_t at different time points, using (7). Since $T >> \tau$, we approximate $T-\tau \cong T$ in using relation (6), and omit the slight difference between the μ_c involved, for simplicity:

$$\begin{aligned} s_{c_t,\Delta\Delta c_t}(0) &\cong \frac{1}{K^2}\frac{1}{T}\sum_{t=1}^{T}(\Delta\Delta c_t-\mu_{\Delta\Delta c_t})(c_t-\mu_{c_t}) \\ &\cong \frac{1}{K^2}\left[8s(4)+8s(3)+2s(2)-8s(1)-10s(0)\right], \end{aligned} \tag{8}$$

where all the $s(\tau)$, $\tau=0,\ldots,4$ without subscripts refer to covariance between realisations of c_t at t and $c_{t+\tau}$ at $t+\tau$.

From (7) and (8) we observe that $\Delta\Delta c_t$ contains c_t, c_{t+1} and c_{t-1} terms, which are all negative, and that the total numbers of negative and positive terms in $s_{c_t,\Delta\Delta c_t}(0)$ are both 18. Furthermore, the (positive) terms $s(\tau)$ for larger τ generally have smaller absolute values (see Figure 3.7). Therefore $s_{c_t,\Delta\Delta c_t}(0)$ should have a negative value.

To further check this numerically, we use values of these s from the training set of TIMIT (Figure 3.7) and a further normalisation to the correlation coefficients

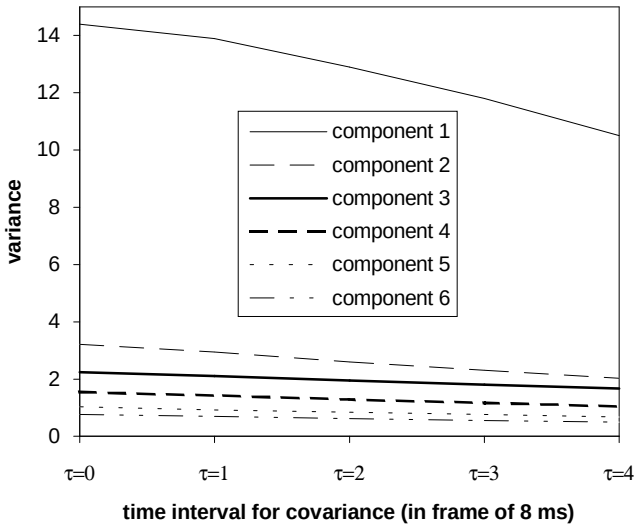


Figure 3.7 Covariance between the first 6 MFCC components and their realisations at various time intervals τ from 0 to 4, calculated from the whole TIMIT training set (3696 sentences). For all the components, the covariance shows strict decreasing behaviour as a function of τ.

$$r_{c_t \Delta\Delta c_t}(0) = \frac{s_{c_t \Delta\Delta c_t}(0)}{\sqrt{s_{c_t c_t}(0) s_{\Delta\Delta c_t \Delta\Delta c_t}(0)}},$$

we obtain, e.g., for the first MFCC $r_{c_t \Delta\Delta c_t}(0)_1 = -0.636$ and for the last MFCC $r_{c_t \Delta\Delta c_t}(0)_{13} = -0.531$, respectively (Note that the subscripts 1 and 13 after the brackets are the indices within the MFCC vector). Such a correlation may certainly not be ignored, and this also explains the existence of the systematic (negative) correlation between c and $\Delta\Delta c$ shown in Figure 3.3 for HMMs trained with MFCC.

In general, if the calculation of an order $(m+1)$ coefficient $\Delta^{m+1}c$ from the order m coefficients $\Delta^m c_t$ is of the symmetric form, and $\Delta^m c_t$ at t is not included in $\Delta^{m+1}c$, as in (5), there is no covariance between the two, but there *is* covariance between $\Delta^{m+2}c$ and $\Delta^m c_t$. In (Ljolje, 1994b) $\Delta\Delta c$ apparently has not been calculated from Δc this way because in his data there is covariance between them. Personal communication between the author and Ljolje (1995) about his way of calculating $\Delta\Delta c$ did not resolve this discrepancy.

4

WHOLE-MODEL DURATIONAL PROBABILITY DENSITY FUNCTIONS*

Abstract

One of the common approaches to duration modelling in HMM is to add explicit durational probability functions (pdf) to each single state, hoping to overcome the improper state durational pdf of standard HMM which is geometrical. It is discussed in this chapter that attention should be paid to the pdf of a whole HMM containing multiple states, instead of the pdf of each single state. We will describe how the closed-form of the whole-model pdf for general left-to-right HMM, which has no explicit state durational pdf, can be obtained. Closed-forms of the whole-model pdf show a rich capacity in modelling various durational distributions of practical phonetic segments. Some HMM topologies are given as examples to explain why a linear topology will be chosen. Relations between the model parameters and the modelled durational statistics are presented for linear models (relations for models with skip transitions are given in an appendix). Finally, a discussion is given on the problem of the divergence between the fact that the whole-model pdf is only governed by the transition probabilities of the HMM, and the fact that the observation probability plays an (even more important) role in determining the transition behaviour. An acoustics-related durational pdf is proposed.

*This chapter is a revised and extended version of the first part of Wang (1994).

4.1 Introduction

It is well-known that the durational probability density function (pdf) of a single state i of an HMM is

$$P_i(d) = a_{ii}^{d-1}(1-a_{ii}), \qquad d \geq 1. \tag{1}$$

$P_i(d)$ denotes the probability of staying in state i for exactly d time steps, and a_{ii} is the selfloop probability of state i. This is a geometrical distribution. It gets to its maximal value at the minimal duration $d = 1$, and decays exponentially as d increases. Such durational behaviour is regarded as improper in modelling the duration of actual speech segments. Based on the consideration of this improper behaviour at the state level, one started to modify the HMM states. A common idea concerning the durational modelling in HMM is to replace the pdf of (1) with some well-chosen pdf close to the durational distribution of speech segments. HMMs with such an explicitly added state durational pdf are called hidden semi-Markov models (HSMM) (e.g., Guédon, 1992), because the transition properties are no longer governed by a Markov process. The selfloops are removed: $a_{ii} = 0, (i = 1, 2, \cdots, n)$.

Various state durational pdf's have been chosen for HSMM, such as the Poisson distribution (Russell & Moore, 1985), the Gamma distribution (Levinson, 1986), and the Gaussian distribution (Hochberg & Silverman, 1993) for recognition purposes and a Log-Normal distribution (Crystal & House, 1988a) for a pure modelling purpose. All of these distributions have a common feature that they contain a single peak not located at the minimal duration, which makes them suitable for modelling the duration of speech segments. For example, a Gamma distribution has the form

$$P_i(d) = \frac{\eta_i^{\nu_i}}{\Gamma(\nu_i)} d^{\nu_i - 1} e^{-\eta_i d},$$

where η_i and ν_i are two parameters for state i and Γ is the usual Gamma function. Each of such explicit pdf's is governed by its parameters which ought to be estimated from the training data so that the actual pdf matches the durational distribution of the segment under concern. Parameter estimation algorithms have been developed for HSMM. The HSMM improves the recognition performance, at least for simple isolated-word recognition tasks (e.g., Russell & Moore, 1985). On the other hand, HSMM increases the system complexity of a recogniser. Both the memory space and calculation costs increase with a factor from 2 to a few hundred, as compared to that of HMM (Mitchell et al., 1995).

An obvious missing point in the argument behind all the aforementioned approaches is a question as at *which level* the durational behaviour should be matched to the durational distribution of the speech, i.e., phone segments or states. In most practices of speech recognition using HMM, it is not the single

state, but an HMM with multiple states, that models the speech segments. Therefore it would be important to investigate the durational behaviour of a whole multiple-state HMM. Furthermore, in continuous speech recognition with phones as sub-word units, duration modelling at the state level will be less important because of many higher-level structures in the recogniser.

Viewpoints different from the pure state-level explicit durational pdf have in fact already been investigated. Hochberg & Silverman (1993) used (Gaussian) explicit state durational pdf but an additional constraint is cast on the durational variance of the whole model. Guédon (1992) compared HMM, HSMM and a model without explicit state durational pdf, but each state (called *macro-state*) is composed of a number of sub-states in a cascade, whose observation probabilities are tied together. The durational pdf of such a macro-state resembles the pdf of a Gamma distribution.

What is left to be investigated is to use simply *one* macro-state, which is in fact an HMM with a number of states without the explicit durational pdf, to model the speech segments, while constraining the whole-model durational pdf to that of the statistics obtained from speech data. It is unnecessary to tie the observation probabilities of different states because now we have only one such macro-state. It is the main topic of this chapter to investigate the durational behaviour of the whole HMM in terms of the closed-form durational pdf (Section 4.2). In Section 4.3 and Appendix 4.1, an acoustics-related durational pdf, which is supposed to include the influence of the acoustic observation probability of the HMM, will be proposed.

4.2 Forms of durational pdf of general HMM

The most extreme transition topology (connections between the states) of an HMM is *ergodic,* in which each state transits to all states. In speech recognition, however, *left-to-right* is by far the most commonly used topology. In this topology, the states are indexed from 1 (begin) to n (end), and only transitions from state i to j are allowed where $j \geq i$. A transition back to a state itself is called a *selfloop*, whereas a feedback loop involving more than one state is not allowed. A transition going to a state beyond the next one is called a *skip*. There can be a number of *linear paths* in a model. Each linear path contains a single cascade of states, and goes from state 1 to state n. In this study we call a model with only one path *linear*.

The durational behaviour is characterised by a durational pdf. In this section we derive the closed form whole-model durational pdf for a general left-to-right HMM. First a few methods of calculating the pdf will be given. Then the general properties of the pdf will be analysed with the help of some examples of useful topologies.

4.2.1 Obtaining the durational pdf of the whole model

The simplest way to get a *numerical* form of the durational pdf is to make use of a property of the Markov chain (Lloyd, 1980): The probability of going from state 1 to state n of a Markov chain in exactly d time steps is an entry in a product matrix of the transition matrix A, namely

$$P_n(d) = \hat{a}_{1n}, \{\hat{a}\} = \hat{A} = A^d, \tag{2}$$

where A^d denotes A to the power of d. This makes one point on the pdf. The whole pdf can be calculated for all d values under concern.

In the following, however, we derive the analytical (or closed) form of the pdf, in order to get some *insight* into the durational behaviour of the HMM. First we will look at the durational pdf of a single cascade of n selfloops. Then we analyse pdf's of some example left-to-right HMMs.

4.2.1.1 Pdf's of linear cascades

We start with $n = 2$ in the cascade. Since the total duration d in two states is a random variable being the sum of the durations d_1 and d_2 in the two cascaded selfloops, each being an independent random variable, the total pdf is the convolution of the two geometrical pdf's $P_1(d)$ and $P_2(d)$ (see, e.g., Papoulis, 1990), each given by (1). This principle is easily extended to a linear cascade of $n > 2$ selfloops, and the whole pdf is

$$P_n(d) = \left[* \right]_{i=1}^{n} P_i(d), d \geq n,$$

where $*$ denotes convolution in d. A special case where all the selfloops are mutually different $a_i \neq a_j, (i \neq j)$, results in a simple closed form, which is a weighted sum of the individual geometrical terms[1]:

$$P_n(d) = \left[\prod_{i=1}^{n} (1 - a_i) \right] \left[\sum_{i=1}^{n} \left(\prod_{j \neq i} \frac{1}{a_i - a_j} \right) \quad a_i^{d-1} \right], d \geq n. \tag{3}$$

However, when some of a_i are equal, the closed form gets more complicated. A direct convolution of a number of different selfloops would require a troublesome book-keeping procedure since each partial result of convolution should be further convoluted with all the remaining terms. In the following we use z-transform to help the calculation. Assume the general case with K subsets each having n_k equal a_k, and the total number of selfloops is $n = n_1 + n_2 + \cdots + n_K$. Since each selfloop a_k has a constant factor $(1 - a_k)$, being the probability of going out of the state, we can write the pdf of the cascade as

[1]For discussions about linear cascades of selfloops, it is sufficient to write a_{ii} as a_i.

$$P_n(d) = \left[\prod_{k=1}^{K}(1-a_k)^{n_k}\right]\hat{P}_n(d), \tag{4}$$

where

$$\begin{aligned}\hat{P}_n(d) &= \underbrace{a_1^{d-1} * a_1^{d-1} * \cdots * a_1^{d-1}}_{n_1} * \underbrace{a_2^{d-1} * a_2^{d-1} * \cdots * a_2^{d-1}}_{n_2} * \cdots * \underbrace{a_K^{d-1} * a_K^{d-1} * \cdots * a_K^{d-1}}_{n_K} \\ &= \left[\underset{k=1}{\overset{K}{*}}\right]\left[\underset{s=1}{\overset{n_k}{*}}\right] a_k^{d-1}.\end{aligned} \tag{5}$$

The z-transform of $\hat{P}_n(d)$ is very simple:

$$\hat{P}_n(z) = \prod_{k=1}^{K}\frac{1}{(z-a_k)^{n_k}}. \tag{6}$$

It is known (Wang, 1993) that each subset of n_k equal selfloops has a negative-binomial pdf. Using z-transformation properties (linearity and shift, see, for instance, Rabiner & Schafer, 1978) and some induction, we have the z-transform of the general negative-binomial terms:

$$\binom{d-1}{i-1} a_k^{d-i} v(d-i) \leftrightarrow \frac{1}{(z-a_k)^i}, i = 1,2,\cdots,n_k. \tag{7}$$

Here v is a step function ($v(d)$ is 1 for $d \geq 0$ and 0 otherwise), $\leftrightarrow$ denotes z-transformation, and the binomial function is written out

$$\binom{d-1}{i-1} = \frac{1}{(i-1)!}(d-1)(d-2)\cdots(d-i+1),$$

where $(i-1)$ is called the *order* of the binomial. Due to the coexistence of the K subsets of equal a_k, the pdf $\hat{P}_n(d)$ in (5) of the cascade may contain in general also lower-order binomial terms for each subset. When each such term is z-transformed using (7), the whole pdf is

$$\hat{P}_n(d) = \sum_{k=1}^{K}\sum_{i=1}^{n_k} C_k^i \binom{d-1}{i-1} a_k^{d-i} v(d-i) \leftrightarrow \sum_{k=1}^{K}\sum_{i=1}^{n_k}\frac{C_k^i}{(z-a_k)^i}. \tag{8}$$

When we equate the right-hand side of (6) to that of (8),

$$\prod_{k=1}^{K}\frac{1}{(z-a_k)^{n_k}} = \sum_{k=1}^{K}\sum_{i=1}^{n_k}\frac{C_k^i}{(z-a_k)^i},$$

the problem is converted to finding the n coefficients C_k^i in the summed form by equalling the terms with the same orders of z to those in the product form.

To find the C_k^i we can use a classical procedure *partial fraction decomposition*. In the following example, a *Mathematica*®[2] function *Apart* is

[2]Software of Wolfram Research, Inc., mainly for symbolic manipulations.

used to obtain C_k^i of the closed form pdf. Assume 6 selfloops, in 3 subsets with 3, 2, and 1 selfloops, and $a_1 \neq a_2 \neq a_3$. The z-transform of $\hat{P}_6(d)$ is

$$\hat{P}_6(z) = \frac{1}{(z-a_1)^3(z-a_2)^2(z-a_3)},$$

The $\hat{P}_6(d)$ with the coefficients given in terms of the a's is[3]

$$\begin{aligned}\hat{P}_6(d) = {} & \frac{1}{(a_1-a_2)^2(a_1-a_3)}\binom{d-1}{2}a_1^{d-3} + \frac{-3a_1+a_2+2a_3}{(a_1-a_2)^3(a_1-a_3)^2}\binom{d-1}{1}a_1^{d-2} \\ & + \frac{6a_1^2-4a_1a_2+a_2^2-8a_1a_3+2a_2a_3+3a_3^2}{(a_1-a_2)^4(a_1-a_3)^3}a_1^{d-1} + \frac{1}{(a_2-a_1)^3(a_2-a_3)}\binom{d-1}{1}a_2^{d-2} \\ & + \frac{a_1-4a_2+3a_3}{(a_2-a_1)^4(a_2-a_3)^2}a_2^{d-1} + \frac{1}{(a_3-a_1)^3(a_3-a_2)^2}a_3^{d-1}, \quad d \geq 6.\end{aligned}$$

The pdf is then (using (4))

$$P_6(d) = (1-a_1)^3(1-a_2)^2(1-a_3)\hat{P}_6(d).$$

It can be seen that, except for the highest order binomial term in each subset, the coefficients for even such a simple cascade are complicated (the coefficient for the lowest order binomial contains 6 terms in its numerator). For another example with a cascade of a total of 10 selfloops in 4 subsets with 4, 3, 2 and 1 equal selfloops, respectively, the most complicated coefficient contains 54 terms in its numerator, as found with *Mathematica* in one hour of time for symbolic manipulations. Reading such a closed-form pdf would not be very insightful nor pleasant. We end up here with the following knowledge: For a linear path, each subset of n_k equal selfloops (the locations of these selfloops are irrelevant) with probability a_k will generally give rise to n_k negative-binomial terms with orders $0, 1, \cdots, n_k - 1$, respectively (order 0 being geometrical). The pdf of the whole cascade is the sum of the n binomial terms for all the subsets of selfloops, then multiplied by the constant factors of (4):

$$P_n(d) = \left[\prod_{k=1}^{K}(1-a_k)^{n_k}\right]\sum_{k=1}^{K}\sum_{i=1}^{n_k}C_k^i\binom{d-1}{i-1}a_k^{d-i}v(d-i).$$

For the special case where all the a_i in the cascade are the same, the closed form reduces to just one binomial term of order n (Wang, 1993).

4.2.1.2 Analysis of whole-model pdf's of left-to-right HMMs

The pdf of any left-to-right HMM can be obtained by considering each *linear path* separately as above, and summing them together with the constant factor in (4) as the weight for that path. Each path has a *scope* $d \geq d_0$ within

[3]At any d<6, some individual terms may not be zero, but the sum of all terms is zero.

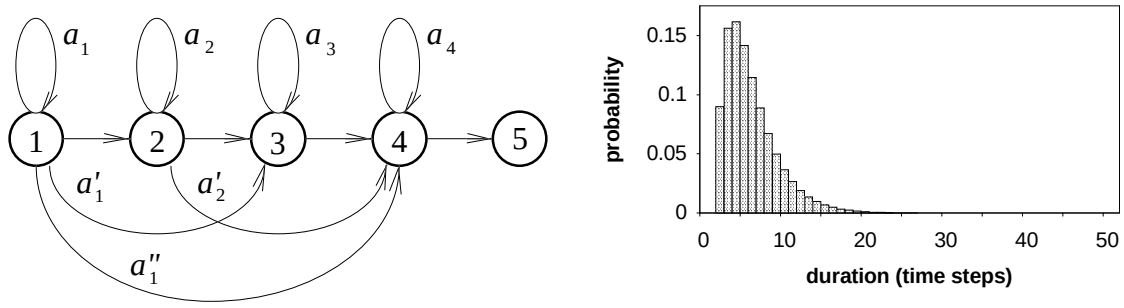


Figure 4.1. Left: an HMM with its transition probabilities shown. It has 5 states and 4 paths. These paths given in their state-indices are (1→2→3→4→5), (1→2→4→5), (1→3→4→5) and (1→4→5) respectively. Right: its durational pdf with specific values assigned to all the transition probabilities, as a function of any constant time steps.

which the pdf is non-zero, where d_0 is the minimal number of time steps needed to go through the path (the *minimal duration*). If some states in a path have no selfloop, the pdf for this path will not contain binomial terms for those states, and their contributions are simply a constant. In the following, we analyse a pdf of an example HMM, for which the transition topology is shown in Figure 4.1, and the selfloop probabilities are all different.

The complete pdf of this HMM consists of 4 parts, each concerning a linear path. Although the geometrical terms with the same a can be put together, they may have different scopes of d, as indicated on the right side of the contributing part of the separate path, in the following formula:

$$P_4(d) = (1-a_1-a_1'-a_1'')(1-a_2-a_2')(1-a_3)(1-a_4)$$

$$[\frac{a_1^{d-1}}{(a_1-a_2)(a_1-a_3)(a_1-a_4)} + \frac{a_2^{d-1}}{(a_2-a_1)(a_2-a_3)(a_2-a_4)}$$

$$+\frac{a_3^{d-1}}{(a_3-a_1)(a_3-a_2)(a_3-a_4)} + \frac{a_4^{d-1}}{(a_4-a_1)(a_4-a_2)(a_4-a_3)}] \qquad (d \geq 4)$$

$$+a_1'(1-a_3)(1-a_4)[\frac{a_1^{d-1}}{(a_1-a_3)(a_1-a_4)}$$

$$+\frac{a_3^{d-1}}{(a_3-a_1)(a_3-a_4)} + \frac{a_4^{d-1}}{(a_4-a_1)(a_4-a_3)}] \qquad (d \geq 3)$$

$$+(1-a_1-a_1'-a_1'')a_2'(1-a_4)[\frac{a_1^{d-1}}{(a_1-a_2)(a_1-a_4)}$$

$$+\frac{a_2^{d-1}}{(a_2-a_1)(a_2-a_4)} + \frac{a_4^{d-1}}{(a_4-a_1)(a_4-a_2)}] \qquad (d \geq 3)$$

$$+a_1''(1-a_4)[\frac{a_1^{d-1}}{a_1-a_4} + \frac{a_4^{d-1}}{a_4-a_1}]. \qquad (d \geq 2)$$

This looks complicated. But if we give some numerical values to each a, e.g.,

a_1	a_1'	a_1''	a_2	a_2'	a_3	a_4
0.1	0.2	0.3	0.4	0.5	0.6	0.7

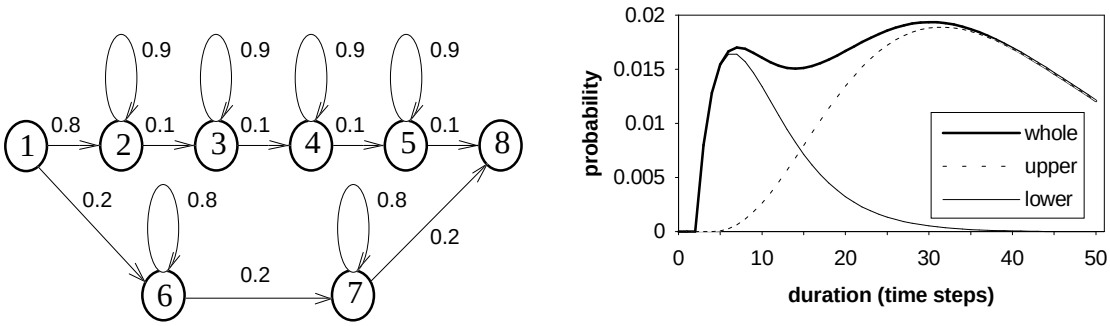


Figure 4.2. Left: an HMM with 2 parallel paths each containing selfloops. Right: the durational pdf of the upper and lower paths and that of the whole model.

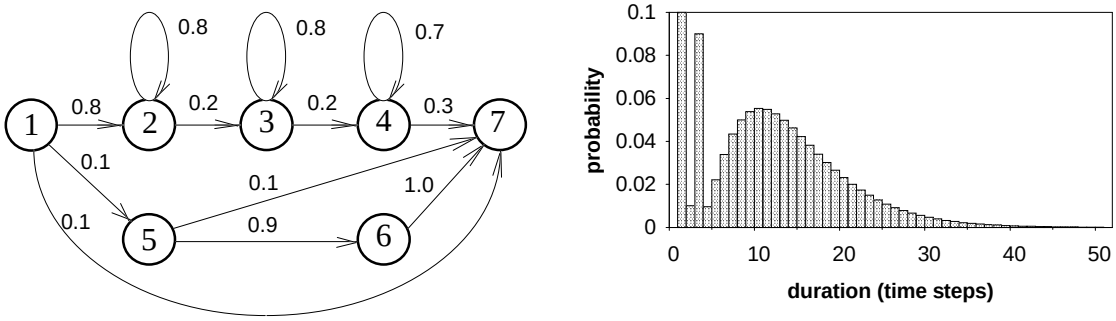


Figure 4.3. Left: the Kai-Fu Lee model with transition probabilities shown. The 3 lower paths given in their state-indices are (1→7), (1→5→7) and (1→5→6→7), respectively. Right: its durational pdf. The values at duration 1, 2 and 3 are contributions from the 3 lower paths respectively, all without selfloops.

then the total pdf, after being organised for different scopes of d, is simple:

$$P_4(d) = \begin{cases} 0, & d < 2; \\ 0.09, & d = 2; \\ 0.156, & d = 3; \\ 0.21(0.1)^{d-1} - 0.4(0.4)^{d-1} - 0.96(0.6)^{d-1} + 1.15(0.7)^{d-1}, & d \geq 4, \end{cases}$$

and this is plotted in the right panel of Figure 4.1.

Next, we show two examples of models with parallel paths, both with numerical transition probabilities. The first model (Figure 4.2) consists of two paths[4], each contributing a single peak and the pdf of the whole model shows two peaks. The second model (Figure 4.3) is the "K.-F. Lee" model (Lee, 1989), with four paths. Each of the 3 lower paths contributes only a single point on the pdf (the first 3 points) since they do not contain any selfloops.

So far, as observed in the above examples, the pdf's of linear models have always a single peak. For models with skipping transitions, the shape of the pdf depends on the lengths of all the linear paths. When the lengths n of all

[4]The durational pdf's in this study are plotted in different styles for the sake of clarity. However it must be noted that even if it is plotted as a continuous line drawing, the pdf values are only defined at discrete time steps (they should actually be called *pmf*: probability mass functions. The values in our pdf's are not "densities", but simply the probabilities).

linear paths do not differ much (in Figure 4.1 this difference is one), the total pdf may have one peak, whereas if there are only very short paths and very long paths, it is possible that the total pdf has more than one peak. Skipping transitions provide a possibility for shorter minimal duration than n of a path. The above is the general durational behaviour of left-to-right HMMs.

As will be seen in Chapter 5, the real data of speech segments always show single-peak behaviour. Furthermore, using parallel paths would probably cause incorrect modelling, because the parameters in different paths would be trained with the *same* data while different paths should model *different* processes. Therefore, we do not use any parallel paths in our study.

4.2.2 Relations between HMM parameters and durational statistics

In Chapter 5 we will use durational mean μ and variance σ^2 of the speech segments to constrain the model parameters. Therefore we will need relations between HMM parameters and these durational statistics. The geometrical pdf of each state i has a durational μ_i and σ_i^2 as (e.g., Papoulis, 1990)

$$\mu_i = \frac{1}{1-a_i}; \qquad \sigma_i^2 = \frac{a_i}{(1-a_i)^2}.$$

The μ_m and σ_m^2 of the whole linear model are sums of those of individual states (because the total duration d in a model is a random variable being the sum of the independent variables d_i of all the states):

$$\mu_m = \sum_{i=1}^{n} \frac{1}{1-a_i}; \qquad \sigma_m^2 = \sum_{i=1}^{n} \frac{a_i}{(1-a_i)^2}. \tag{9}$$

These relations between (μ_m, σ_m^2) and $(n, \{a_i\})$ are very simple. They can be readily used in the training procedure of Chapter 5 to constrain (μ_m, σ_m^2) to the data statistics (μ, σ^2). In addition to constraining (μ_m, σ_m^2), if one also wants to obtain a fitted minimal duration, one can make use of models with skipping transitions. However, either the relations between (μ_m, σ_m^2) and $(n, \{a_{ij}\})$ are too complicated for such models, thus difficult to use in the training procedure, or it cannot guarantee a single peak pdf (Appendix 4.1). Therefore, for all the aforementioned reasons, we **have chosen to use linear models in the remaining part of this thesis**.

To illustrate the relations between (μ_m, σ_m^2) and $(n, \{a_i\})$, we simplify the selfloop probabilities to be equal: $a_i = a$. Relations (9) are thus simplified as

$$\mu_m = \frac{n}{1-a}; \qquad \sigma_m^2 = \frac{na}{(1-a)^2}, \tag{10}$$

as shown in the upper panels of Figure 4.4. From (10), when either n or a is fixed, both μ_m and σ_m^2 increase monotonically with the other variable. On the other hand, if we eliminate a from (10) to make it implicit and rewrite

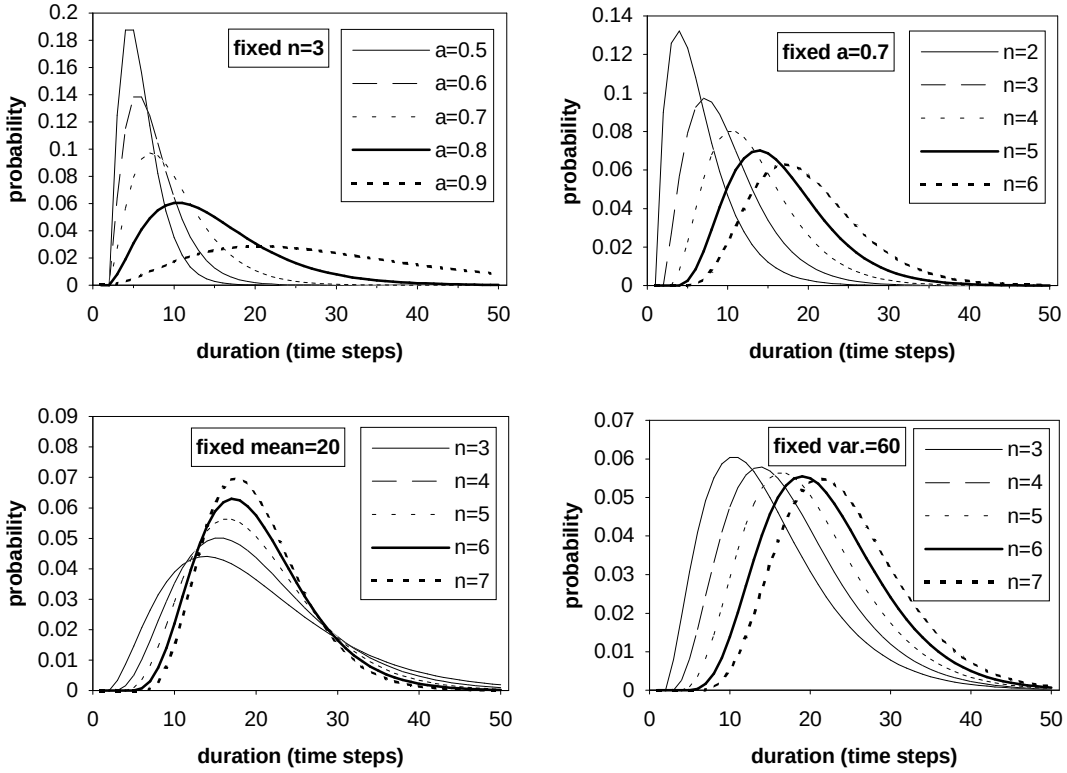


Figure 4.4. Durational pdf's for linear models with equal selfloops. Different panels show cases when either n, a, mean or variance is fixed while other variables are varying.

$$\sigma_m^2 = \frac{\mu_m}{n}(\mu_m - n); \qquad \mu_m = \frac{n}{2} + \sqrt{\sigma_m^2 n + \frac{n^2}{4}},$$

then it can be seen that σ_m^2 decreases as n increases when μ_m is fixed, and μ_m increases with n when σ_m^2 is fixed (the lower panels of Figure 4.4).

It can be seen Figure 4.4 that even the linear models with equal a show a rich modelling capacity for various durational distributions. In the more general case of linear models with different $\{a_i\}$ (Chapter 5), even wider range of durational distributions can be modelled.

4.3 Acoustics-related whole-model durational pdf

The whole-model durational pdf's discussed so far contain explicitly only the A-parameters. Therefore, according to it, merely the transition probabilities govern the durational behaviour. This contradicts the observation that, in both training and recognition procedures, the stochastic decision of a transition to the next state is also governed by the observation probability via acoustic vectors. In other words, only when

$$a_{ii} b_i(\mathbf{o}_t) < a_{ij} b_j(\mathbf{o}_t)$$

is the transition $i \to j$ more favourable than staying in state i. Since the b terms have, in general, a much larger dynamic range of values than the a terms (Juang, & Rabiner, 1992), the total transition behaviour is actually governed more strongly by the observation probability than by the transition probability itself. We anticipate, therefore, that a durational pdf which characterises the true durational behaviour of an HMM should at least include both A and B parameters in some way[5]. We call such a pdf an *acoustics-related durational-pdf* (ARD-pdf). The durational pdf discussed up to Section 4.2 will simply be referred to as a durational pdf or D-pdf. Detailed discussion of the ARD-pdf is given in Appendix 4.2. Due to the time limit of the project, the ARD-pdf is only proposed in this chapter, but not developed further. In later chapters, only D-pdf will be used for duration modelling.

Note that the attempt to define an ARD-pdf is only one way to model the influence of acoustic observation, whereas other possible ways may treat such influences without explicitly using a durational measure (Chapter 7).

4.4 Conclusion

In this chapter, we first discussed the reasons why we decided not to study the commonly used HSMM. The main reasons are: (1) It undesirably increases the system complexity; (2) it has been systematically studied in the literature; (3) it only concerns the duration at the state level. Then we discussed the durational pdf of the standard HMM with a general left-to-right topology, at the whole-model level. Both simple numerical calculation and analytical forms of the pdf are presented.

It is revealed that each linear path of the left-to-right model has a suitable binomial-like form to model the segmental duration. A decision is made to use only linear HMMs in this study. Relations between the HMM parameters and the durational mean and variance are presented for the purpose of durational fitting in Chapter 5. Finally, an acoustics-related durational pdf is proposed based on the consideration that the observation probabilities play an important role in governing the between-state transitions.

Appendix 4.1 Durational mean and variance of HMM with skips

In this appendix we derive the relations between the model parameters $(n, \{a_{ij}\})$ and the HMM-modelled durational statistics (μ_m, σ_m^2). The first section presents analytical relations for HMMs with a specific kind of skip

[5]In some operational recognisers, the significance of A is completely eliminated by setting all the selfloop probabilities to a single constant value.

transitions. The second section derives such relations in numerical form, for HMMs with more general kinds of skips.

A4.1.1 Analytical relations for specific skips

In our recogniser, the data statistics (μ, σ^2) for each phone will require a different suitable n for each linear HMM (in our case $n \geq 3$, see Chapter 5). Two skips are added to each linear model with $n > 3$ in a simple way (each of the first two states $i \leq 2$ starts a skip transition with the probability $a_{i,i+n-2}$ to state $i+n-2$), to make the minimal duration $d_{\min} = 3 \leq n$ for all the phone HMMs (Figure 4.5; n is only the number of selfloops).

We will use the *probability generating function* (p.g.f., Lloyd. 1980) as a tool to derive the relations between $(n, \{a_{ij}\})$ and (μ_m, σ_m^2). In Section 4.2, since the purpose was to get the closed-form pdf, p.g.f. is not used since the p.g.f. should still be converted into pdf. Now we really need some properties of p.g.f. to represent (μ_m, σ_m^2) in terms of $(n, \{a_{ij}\})$, for models with skips.

The p.g.f. for any pdf $P(d)$ is defined as

$$g(s) = \sum_{d=0}^{\infty} P(d) s^d,$$

where s is a formal variable. The total pdf $P(d)$ of this model is the sum of the pdf $P_k(d)$ of the 3 linear paths. Therefore the total p.g.f. $g(s)$ is also the sum of all $g_k(s)$:

$$g(s) = \sum_{d=0}^{\infty} \sum_{k=1}^{3} P_k(d) s^d = \sum_{k=1}^{3} \sum_{d=0}^{\infty} P_k(d) s^d = \sum_{k=1}^{3} g_k(s).$$

Since $P_k(d)$ is a *convolution* of all the selfloops in path k, the *product* property of p.g.f. gives $g_k(s) = \prod_i g_{ki}(s)$. Since $P_{ki}(d) = a_i^{d-1} a_{k,out\_i}, (d \geq 1)$, where a_{k,out_i} is the probability of going out of state i along path k, we have

$$g_{ki}(s) = \sum_{d=0}^{\infty} P_{ki}(d) s^d = a_{k,out\_i} \sum_{d=1}^{\infty} a_i^{d-1} s^d = a_{k,out\_i} s \sum_{d=0}^{\infty} (a_i s)^d = a_{k,out\_i} \frac{s}{1 - a_i s}.$$

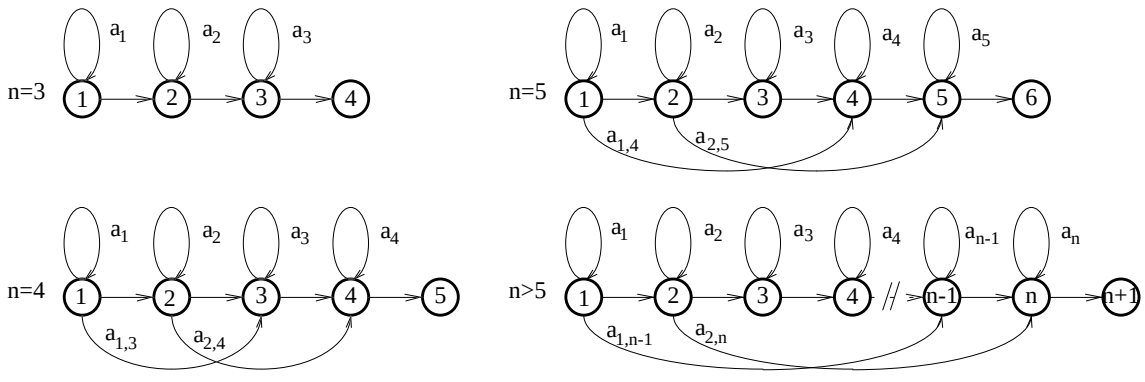


Figure 4.5 Different HMMs with n selfloops. For each HMM with n>3, there are two skips.

Considering the different compositions of $\{a_{k,out\_i}\}_i$ in each path k, the explicit forms of the $g_k(s)$ are, respectively

$$g_1(s) = (1-a_1-a_{1,n-1})(1-a_2-a_{2,n})\frac{\prod_{i=3}^{n}(1-a_i)}{\prod_{i=1}^{n}(1-a_i s)}s^n;$$

$$g_2(s) = a_{1,n-1}(1-a_{n-1})(1-a_n)\frac{s^3}{(1-a_1 s)(1-a_{n-1}s)(1-a_n s)};$$

$$g_3(s) = (1-a_1-a_{1,n-1})a_{2,n}(1-a_n)\frac{s^3}{(1-a_1 s)(1-a_2 s)(1-a_n s)}.$$

Another property of p.g.f. gives the expressions of model statistics (μ_m, σ_m^2):

$$\mu_m = g'(1); \qquad \sigma_m^2 = g''(1) + g'(1) - \{g'(1)\}^2,$$

where the primes above $g(s)$ denote the derivatives with respect to s, and $g'(1)$ denotes $g'(s)|_{s=1}$. We use $\log g(s)$ to ease calculation. The resulting $g_k'(s)$ and $g_k''(s)$ for, e.g., the upper path $k=1$ are[6]

$$g_1'(s) = \prod_{i=1}^{2}(1-a_i-a_{i,i+n-2})\frac{\prod_{i=3}^{n}(1-a_i)}{\prod_{i=1}^{n}(1-a_i s)}\left[\frac{n}{s}+\sum_{i=1}^{n}\frac{a_i}{1-a_i s}\right]s^n;$$

$$g_1''(s) = \prod_{i=1}^{2}(1-a_i-a_{i,i+n-2})\frac{\prod_{i=3}^{n}(1-a_i)}{\prod_{i=1}^{n}(1-a_i s)}\left[\left(\frac{n}{s}+\sum_{i=1}^{n}\frac{a_i}{1-a_i s}\right)^2-\frac{n}{s^2}+\sum_{i=1}^{n}\frac{a_i^2}{(1-a_i s)^2}\right]s^n.$$

We omit rest of the algebraic manipulations here[7] and give only (μ_m, σ_m^2). Due to the crossed terms introduced by the square in $\{g_k'(1)\}^2$, which cannot be further simplified, the final expressions are rather complicated:

$$\mu_m = \sum_{k=1}^{3} C_k D_k;$$

$$\sigma_m^2 = \sum_{k=1}^{3} C_k (D_k^2 + E_k) - \left(\sum_{k=1}^{3} C_k D_k\right)^2,$$

where the coefficients are

$$C_1 = \prod_{i=1}^{2}\frac{1-a_i-a_{i,i+n-2}}{1-a_i}, \quad C_2 = \frac{a_{1,n-1}}{1-a_1}, \quad C_3 = \frac{(1-a_1-a_{1,n-1})a_{2,n}}{(1-a_1)(1-a_2)};$$

$$D_1 = \sum_{i=1}^{n}\frac{1}{1-a_i}, \quad D_2 = \sum_{i=1,n-1,n}\frac{1}{1-a_i}, \quad D_3 = \sum_{i=1,2,n}\frac{1}{1-a_i};$$

---

[6]The selfloop probabilities a_i are specified by one subscript, whereas the skip transition probabilities $a_{i,i+n-2}$ are specified by two subscripts.

[7]It is shown in Subsection 4.2.2 that whether some a_i are equal or not will lead to different closed-form pdf's. However, expressions of μ_m and σ_m^2, instead of pdf, are of concern here. $d = \sum_{i=1}^{n} d_i$ always holds irrespective of the compositions of $\{a_i\}$, and the expressions of μ_m and σ_m^2, do not contain factors $1/(a_i - a_j)$ that may lead to evaluation problems for those $a_i = a_j$. Therefore μ_m and σ_m^2 in this appendix are general for any compositions of $\{a_i\}$.

$$E_1 = \sum_{i=1}^{n} \frac{a_i}{(1-a_i)^2}, \qquad E_2 = \sum_{i=1,n-1,n} \frac{a_i}{(1-a_i)^2}, \qquad E_3 = \sum_{i=1,2,n} \frac{a_i}{(1-a_i)^2}.$$

It can be seen that when all $a_{i,i+n-2} = 0$ (no skips), $C_1 = 1$ and $C_2 = C_3 = 0$, thus both μ_m and σ_m^2 reduce to formula (9) for linear models in Subsection 4.2.2.

Since the final relations for μ_m and σ_m^2 only contain $\{a\}$, they hold for any values of $\{a\}$. Therefore, in principle, such relations can be used in the constrained training of Chapter 5, although these relations are much more complicated than the relations for linear models. The drawback of such models with only two specific skips is that they may give multiple-peak pdf, especially for long models. (The two skipping linear paths which each contains only two selfloops and may show peaks at short durations, and the long upper linear path may show another peak at a long duration).

A4.1.2 Numerical relations for general skips

More number of skips would have to be added to the above two-skip model to prevent multiple-peak pdf's. An extreme case of a left-to-right model (Figure 4.6) with all possible skips going from all the n states to all states on their right, can guarantee a single-peak pdf since all the linear paths only differ in lengths by one. However, it is hard to derive analytical relations in the same way as above. We present here a partial numerical solution for such models.

For a given set of selfloop probability values $\{a_i\}, i = 1,\cdots,n$ of, e.g., a trained HMM, the pdf of the HMM can always be written as

$$P(d) = \sum_{i=1}^{n} c_i a_i^{d-1}, \qquad d \geq n. \tag{11}$$

The coefficients c_i can be obtained, e.g., by evaluating $P(d)$ at n points of d (where each d should not be smaller than n since then (11) may not hold due to the skipping paths), to get a set of n equations, with c_i as variables and $\{a_i\}$ as coefficients. The n numerical values of $P(d)$, treated as constants here, can be obtained by, e.g., multiplications of the transition matrix as in (2). The solutions to this set of equations are then $c_i = c_i(a_1,\cdots,a_n), i = 1,\cdots,n$.

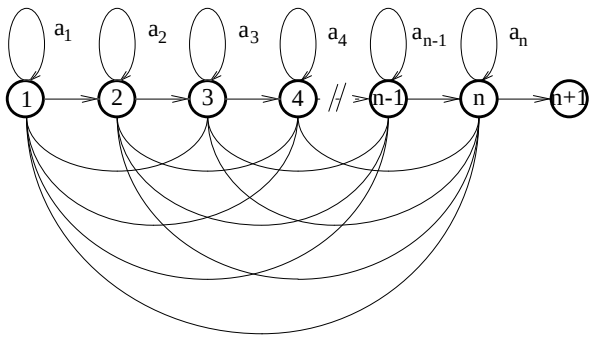


Figure 4.6. A general left-to-right HMM with n selfloops and all possible skips.

For each of the n values of $d < n$, although (11) may not hold, a numerical value of $P(d)$ can still be obtained. Then the p.g.f. of the whole $P(d)$ is

$$g(s) = \sum_{d=0}^{n-1} P(d)s^d + \sum_{d=n}^{\infty} P(d)s^d$$
$$= P(0) + P(1)s + P(2)s^2 + \cdots + P(n-1)s^{n-1} + \sum_{i=1}^{n} c_i \sum_{d=n}^{\infty} a_i^{d-1} s^d,$$

where we know $P(0) = 0$. The inner summation of the last term can be written out as

$$\sum_{d=n}^{\infty} a_i^{d-1} s^d = \frac{1}{a_i}\left[-1 - a_i s - a_i^2 s^2 - \cdots - a_i^{n-1} s^{n-1} + \frac{1}{1 - a_i s}\right].$$

Then the p.g.f. has a form

$$g(s) = K_0 + K_1 s + K_2 s^2 + \cdots + K_{n-1} s^{n-1} + \sum_{i=1}^{n} \frac{c_i}{a_i} \frac{1}{1 - a_i s},$$

with

$$K_0 = -\sum_{i=1}^{n} \frac{1}{a_i}, \qquad K_1 = P(1) - n,$$
$$K_2 = P(2) - \sum_{i=1}^{n} a_i, \quad \cdots, \quad K_{n-1} = P(n-1) - \sum_{i=1}^{n} a_i^{n-2}.$$

Taking $g(s)$ only as a function of s and all a, c, K and P as constants, the next step to obtain $g'(s)$ and $g''(s)$ is straightforward. Then expressions of μ_m and σ_m^2 can be obtained, which will be omitted here because they are more complicated than the μ_m and σ_m^2 in the previous section. Furthermore, a more important difference from the previous section is as follows.

In all the steps of the above procedures, $P(d)$, $d = 1, \cdots 2n-1$, c, K and the last μ_m and σ_m^2, can take one of the two ways of $\{a_i\}$:

1. A numerical set of values of $\{a_i\}$;
2. Symbolic forms of $\{a_i\}$ (then also all skipping probabilities $\{a_{ij}\}$ via $P(d)$).

The expressions of μ_m and σ_m^2 of the second (fully symbolic) type will be much more complicated than the first (partially numerical) type. In the constrained ML-equations of Chapter 5, however, all $\{a_i\}$ and $\{a_{ij}\}$ are required to be in their symbolic forms, since each update of them will give rise to new durational statistics. So only the second fully symbolic μ_m and σ_m^2 may in principle be used, however their extremely complicated form would make the rest of the ML-training procedure impossible. The first partially numerical expressions of μ_m and σ_m^2, although simpler in form, only hold following a fixed-point evaluations of current values of $\{a_i\}$. So they can only be used to calculate the current μ_m and σ_m^2 values, but not be used in the ML-equations.

The only way to use these expressions of μ_m and σ_m^2 is thus an iterative process, in which each step takes the current values of $\{a_i\}$, and some different ways of modification of $\{a_i\}$ are needed.

Appendix 4.2 Acoustics-related durational pdf

In this appendix, first the starting point for defining the acoustics-related durational pdf (ARD-pdf) will be presented. This will lead to a better insight into the problem, and lead to an interpretation of the ARD-pdf. Then the problem of considering the ARD-pdf between the state level and the whole-model level is discussed. Note that only the *problems* with the ARD-pdf are presented here, while no *solutions* are worked out yet. This is why in later chapters of this study, the ARD-pdf will not be used.

A4.2.1 Relation between durational pdf and the likelihood

We first derive a relation between the D-pdf of a whole model and the likelihood $P(\boldsymbol{O}|\lambda)$ that the model λ generates $\boldsymbol{O}$ (Wang, 1995). $P(\boldsymbol{O}|\lambda)$ is a quantity available during the training and recognition process. Then we propose a relation between the ARD-pdf and $P(\boldsymbol{O}|\lambda)$.

The D-pdf is by definition the sum of the probabilities of all the state sequences of length d:

$$P(d) = \sum_S P(S(d)|\lambda).$$

Usually, when the HMM λ is given and d is fixed, $S(d)$ is countable and the summation over S is specified. For each $S(d)$, $P(S(d)|\lambda)$ is simply the product of those transition probabilities given $S(d)$. After each transition, the HMM also generates some acoustic vector $\boldsymbol{o}_t$. The probability that the given $\boldsymbol{O} = \boldsymbol{o}_1\boldsymbol{o}_2\cdots\boldsymbol{o}_d$ is generated by λ with all possible S is the likelihood $P(\boldsymbol{O}|\lambda)$. The summation over all S can be efficiently achieved using the usual forward probability $\alpha_j(t) = P(\boldsymbol{o}_1\boldsymbol{o}_2\cdots\boldsymbol{o}_t, s_t = j|\lambda)$, which is recursively calculated

$$\alpha_j(t) = \left[\sum_{i=1}^{n} \alpha_i(t-1)a_{ij}\right] b_j(\mathbf{o}_t).$$

We define here a similar partial-observation probability $\tilde{\alpha}_j(t)$ which can also be calculated recursively. The difference is that at each time step of recursion, a further summation is performed:

$$\tilde{\alpha}_j(t) = \sum_{\mathbf{o}_t} \left[\sum_{i=1}^{n} \tilde{\alpha}_i(t-1)a_{ij}\right] b_j(\mathbf{o}_t).$$

The summation of $\boldsymbol{o}_t$ is defined to be over all possible values of $\boldsymbol{o}_t$, and it applies only to the b terms. The b terms sum to unity by definition, and then

$$\tilde{\alpha}_j(t) = \sum_{i=1}^{n} \tilde{\alpha}_i(t-1)a_{ij}.$$

It can be seen that at the end of recursion $t = d$, $\tilde{\alpha}_j(t)$ has been summed over all the between-state transition probabilities, effectively all the $S(d)$. Therefore $P(d) = \tilde{\alpha}_{s_d}(d)$ where s_d is the last state visited at d.

From the above procedure, we can give the meaning of $\tilde{\alpha}_j(t)$ as $\tilde{\alpha}_j(t) = P(\forall(\boldsymbol{o}_1\boldsymbol{o}_2\cdots\boldsymbol{o}_t), s_t = j|\lambda)$. An important question is how $\forall(\boldsymbol{o}_1\boldsymbol{o}_2\cdots\boldsymbol{o}_t)$ should be further specified, to be discussed as follows. In the recursive calculation of $\tilde{\alpha}_j(t)$, in effect, each distinct $S(d)$ has been gone through. At each state along each of these $S(d)$, all possible values of $\boldsymbol{o}_t$ are taken. Therefore $\forall(\boldsymbol{o}_1\boldsymbol{o}_2\cdots\boldsymbol{o}_t)$ should not be interpreted as just a single time sequence $\forall\boldsymbol{o}_1\forall\boldsymbol{o}_2\cdots\forall\boldsymbol{o}_t$ in which each $\boldsymbol{o}_t$ takes all its possible values. It should be interpreted as the collection of all the different $S(d)$, and further at each state j, taking all the $\boldsymbol{o}$ values possibly evaluated by $b_j(\boldsymbol{o})$. The number of distinct $S(d)$ for a general HMM with n states is n^d, and for a linear HMM is $(d-1)!/((n-1)!(d-n)!)$. Furthermore, when we take, e.g., a discrete density HMM (DDHMM), an explicit calculation of the summation at each state over $\boldsymbol{o}_t$ should be to take each codeword value exactly once.

We give an example of all the different state sequences $S(4)$ with duration 4, of a linear HMM with $n = 3$ selfloops, in the following table:

$S_1 = s_1s_1s_2s_3$
$S_2 = s_1s_2s_2s_3$
$S_3 = s_1s_2s_3s_3$

There are three such state sequences in total. After transition into each state j of each S, all possible values $\boldsymbol{o}_t$ should be evaluated by $b_j(\boldsymbol{o}_t)$. If a DDHMM is assumed and the codebook size is m, the total number of $\forall(\mathbf{o}_1\mathbf{o}_2\mathbf{o}_3\mathbf{o}_4)$ is thus $3m^4$. The total number of sequences in $\forall\mathbf{o}_1\forall\mathbf{o}_2\forall\mathbf{o}_3\forall\mathbf{o}_4$ in this case is only m^4. Each $\forall\boldsymbol{o}_t$ is only associated with time, since no state assignment is known.

Under the above interpretation of $\forall(\boldsymbol{o}_1\boldsymbol{o}_2\cdots\boldsymbol{o}_t)$, together with the comparison between $\tilde{\alpha}_j(t)$ and $\alpha_j(t)$, the likelihood $P(\boldsymbol{O}|\lambda)$ is related with the D-pdf $P(d)$ in a very specific way. Actually, many different $\boldsymbol{o}_1\boldsymbol{o}_2\cdots\boldsymbol{o}_t$ are needed for a numerical calculation, which would never be available in practice. In order to get the numerical value of $P(d)$ from $P(\boldsymbol{O}|\lambda)$, one has to simulate all the data $\boldsymbol{o}_1\boldsymbol{o}_2\cdots\boldsymbol{o}_t$ for each $S(d)$, and at each state of $S(d)$, each codeword should be taken once. For a continuous-density HMM (CDHMM), further simulation is needed for numerical integration at each state.

For the purpose of actually getting the D-pdf $P(d)$, the above summation of $\boldsymbol{o}_t$ at each state is not necessary since the result is simply one. However

this concept can be a starting point for searching for the ARD-pdf. The idea is to sum using $P(\boldsymbol{O}|\lambda)$ of only the available $\boldsymbol{O}$ in a data set, instead of the $\forall(\boldsymbol{o}_1\boldsymbol{o}_2\cdots\boldsymbol{o}_t)$ as specified above. This way it is assumed that the influence of the acoustic observation from a given data set is included by using only the set. The resulting ARD-pdf, denoted $\tilde{P}(d)$, is supposed to be a better measure of duration than $P(d)$, with respect to the faithful inclusion of only the available data set (e.g., training). In practice, however, the number of actual $\boldsymbol{O}$ is much smaller than all the theoretically possible values as listed above. Therefore, careful *normalisation* of the probability values would be needed.

For both DDHMM and CDHMM, the above discussion about the way of specifying $\forall(\boldsymbol{o}_1\boldsymbol{o}_2\cdots\boldsymbol{o}_t)$ is essential in designing the correct normalisation procedure[8].

A4.2.2 Relation between single-state and whole-model levels

The argument that the observation probabilities play a more important role than the transition probability, was discussed at the state level (Section 4.3, transition between two states). How well this argument also fits into the whole-model level, is a difficult question to answer. Yet the whole model is more important than the states in modelling the speech for recognition. The transition between two HMMs would also be more heavily controlled by the observation probabilities of the acoustic vectors of the two segments, than the whole-model D-pdf of the two HMMs, because the dynamic range of the B-parameters is much larger than the A-parameters. We do not analyse further the exact difference between the two sets of parameters in governing the transition.

Due to the time limit in the project, the ARD-pdf is only developed to this stage. A model that can be used to actually calculate pdf values from HMMs and training data has not yet been worked out. When, in the future, such a model will have been worked out, its use will be similar to that of the D-pdf as used in Chapter 5. Although we do not have a theoretical comparison between the whole-model D-pdf and the whole-model ARD-pdf, whether ARD-pdf is a better measure of duration than the D-pdf can be tested with a recogniser. Two sets of HMMs can be trained using the durationally constrained training procedure of Chapter 5, while the constraints used will be based on either ARD-pdf or D-pdf, respectively. Then the recognition scores with both sets of HMMs can be compared.

---

[8]The author tried a few examples of numerical methods for normalisation, of which the results are omitted here, because the procedures are complicated, yet not essential for the problem. An important step in these normalisation procedures is a partitioning of the actual acoustic space.

5

CONSTRAINING THE MODELLED DURATION WITH DATA STATISTICS*

Abstract

In this chapter we use the knowledge obtained in the previous chapter on the durational behaviour of the standard HMM, to modify the durational behaviour. Given that the form of the durational pdf of linear HMMs is suitable and rich enough to model the segmental duration, the actual values of the HMM parameters may not be perfect because of the ignorance of the durational knowledge in the standard Baum-Welch training procedure. An improved training procedure is developed which uses the lower-order statistics of the phone duration from the training data as extra constraints for the optimisation of the HMM parameters. Comparisons are given between the HMMs trained with and without the durational constraints, both on the fit of their durational pdf on the data histograms, and on the performance of the HMMs in speech recognition and automatic segmentation. Different impacts on these two tasks are shown.

* This chapter is a revised version of the second part of Wang (1994).

5.1 Introduction

In the literature, there are very few attempts to look at the *whole model* durational behaviour (representing, e.g., a phone), let alone attempts to modify it. The usual approach for HSMM (hidden semi-Markov model) only deals with the durational behaviour at the state level. One of the exceptions is found in a recent work of Hochberg & Silverman (1993) on constraining the whole model durational variance, although it was an HSMM (Levinson, 1986) with a Gaussian state explicit durational pdf that was constrained. Other explicit pdf's more suitable than the Gaussian one were not used due to technical difficulties. Wakita & Tsuboka (1994) constrained the state duration with the syllable duration. We see from these approaches that the paradigm of constraining may be a good way to include extra information while retaining the existing optimisation. Our knowledge obtained in the previous chapter indicates the potential suitability of the *standard* linear HMM in duration modelling. Based on this knowledge, we would like to try to modify the HMM parameters based on the whole-model durational statistics, and to see if such modification improves the duration modelling and the performance of speech recognition and segmentation.

We will use the two lower-order durational statistics μ and σ^2 of the phone segments to constrain the modelled durational statistics μ_m and σ_m^2. The relations between (μ_m, σ_m^2) and the HMM parameters $(n, \{a_{ii}\})$ for a linear model, for the general case of arbitrary $\{a_{ii}\}$ are (Subsection 4.2.2)

$$\mu_m = \sum_{i=1}^{n} \frac{1}{1-a_{ii}}; \qquad \sigma_m^2 = \sum_{i=1}^{n} \frac{a_{ii}}{(1-a_{ii})^2}, \tag{1}$$

and for the special case with equal selfloop $a_{ii} = a$ are simply

$$\mu_m = \frac{n}{1-a}; \qquad \sigma_m^2 = \frac{na}{(1-a)^2}. \tag{2}$$

The discussion in Chapter 4 about the D-pdf answered the question about the *form* (topology) of linear HMMs, which is suitable. The actual way to obtain the *values* of the parameters such that the modelled (μ_m, σ_m^2) lead to a good fit to the durational statistics (μ, σ^2) will be discussed in this chapter.

In this chapter, first we will briefly outline the standard Baum-Welch training procedure and explain why the durational behaviour of the HMM thus trained may not be optimal in terms of duration modelling. Next, we will compare a few different training paradigms in which the relations between the model parameters and the measure of segmental duration are different. A relatively lengthy description of an improved training procedure will be presented, mainly in three appendices at the end of this chapter. In fact, the procedure is a modification of the Baum-Welch maximum-likelihood (ML) training with duration measures as extra constraints. Finally, we will

present the results of the durationally constrained training of HMM both in its quality of duration modelling and in its performance in speech recognition and segmentation. A discussion on the approach concludes the chapter.

5.2 Durational behaviour of HMM from standard Baum-Welch training

From the previous chapter, we know that the form of the durational pdf of the standard HMM, being a weighted sum of binomials (governed by *A*-parameters), provides a possibility for modelling the segmental duration. Now we will check if the values of the *A*-parameters obtained from the standard Baum-Welch ML training procedure will lead to an optimal fit.

We present here a simpler formalism (for linear HMMs) than the general case (e.g., Kamp, 1991). The only constraint used in the optimisation is the unity of probabilities, which for *A*-parameters of linear HMM is reduced to[1]

$$a_{ii} + a_{i,i+1} = 1, \qquad i = 1,2,\cdots n. \tag{3}$$

Using this to eliminate $a_{i,i+1}$ from the part of the auxiliary likelihood function Q (see Appendix 2.1) concerning the transition probabilities (for any fixed i), then we have

$$Q_{\bar{a}_{ij}} = \sum_r \sum_t \sum_i \left[\gamma^r_{t-1}(i,i) \log \bar{a}_{ii} + \gamma^r_{t-1}(i,i+1) \log(1-\bar{a}_{ii}) \right], \tag{4}$$

where the "counts" γ are obtained from the previous iteration, and $\bar{a}_{ii}$ are the new values of selfloop probabilities after the current iteration. To get the critical point we stipulate $\partial Q / \partial \bar{a}_{ii} = 0$, which yields

$$\frac{D(i,i)}{\bar{a}_{ii}} - \frac{D(i,i+1)}{1-\bar{a}_{ii}} = 0, \qquad i = 1,2,\cdots,n,$$

where we used the notation

$$D(i,j) = \sum_r \sum_t \gamma^r_{t-1}(i,j)$$

for simplicity, which sums γ over time t and observation sequences r (see Appendix 2.1). The solution is

$$\bar{a}_{ii} = \frac{D(i,i)}{D(i,i) + D(i,i+1)}, \qquad i = 1,2,\cdots,n. \tag{5}$$

This is the re-estimation formula for $\bar{a}_{ii}$ at each iteration. It is given in a nice closed form. The new values $\bar{a}_{ii}$ are directly evaluated using the D values

[1]For our linear models, n represents the number of the selfloops in the cascade, while state n+1 is the last exit state without selfloop.

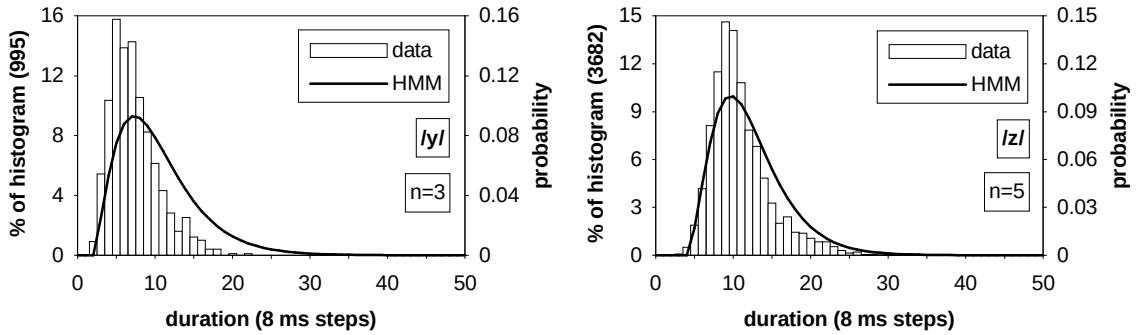


Figure 5.1. The durational pdf's of two example phones /y/ and /z/ from the TIMIT database: Data histograms (data: blank vertical bars) are estimated from the number of instances given between the brackets. Pdf of standard (stand.: continuous curve) HMM is scaled to the histogram. Histograms are plotted against the left axis, and pdf's are plotted against the right axis, of each panel. The chosen model length n (see later sections) is also indicated.

which are all constants after the previous iteration. There are re-estimation formulae for all other HMM parameters.

It can be seen that in the standard Baum-Welch training of HMM, no model-level durational constraints are used. Therefore there may be a divergence between the phone durational distribution in a data set and the modelled durational pdf of the models that have been well trained with the same set of data (or simply between (μ, σ^2) and (μ_m, σ_m^2)). This can be seen in Figure 5.1. for two example phones. Although the lengths of the linear models are chosen properly for the durational statistics (see Subsection 5.3.2), the divergence is evident, especially for the semi-vowel /y/. These data clearly show that the usual ML criterion on the *acoustic* observations will allow for a large freedom for the HMM parameters to have quite different *durational* fit (see Appendix 5.1 for relations between the freedom and model parameters). In the next sections, we will still train the parameter values with the ML criterion, but further constrained by durational statistics, for our purpose of duration modelling.

5.3 Constrained training of HMM embedded in ML procedure

From this point on in this chapter we use "HMM" to refer to a standard (linear) HMM trained with usual Baum-Welch procedures, and "DCHMM" to refer to the HMM trained with durationally constrained procedures.

5.3.1 Paradigm of the constrained training

The whole parameter set of the standard HMM to be estimated is $\lambda = (A, B)$ where A is the transition probability and B is the observation probability. The auxiliary function Q can be decomposed into two terms for A and B

Table 5.1. Different parameter constraining in training of HMM. The second, third and last columns show constraints for A, B and the state durational (dur.) term, respectively.

	constraints for A	constraints for B	constraints for state dur. term
HMM	unity	unity	
HSMM	unity	unity	unity
DCHSMM	unity	unity	unity, σ^2
DCHMM	unity, (μ, σ^2)	unity	

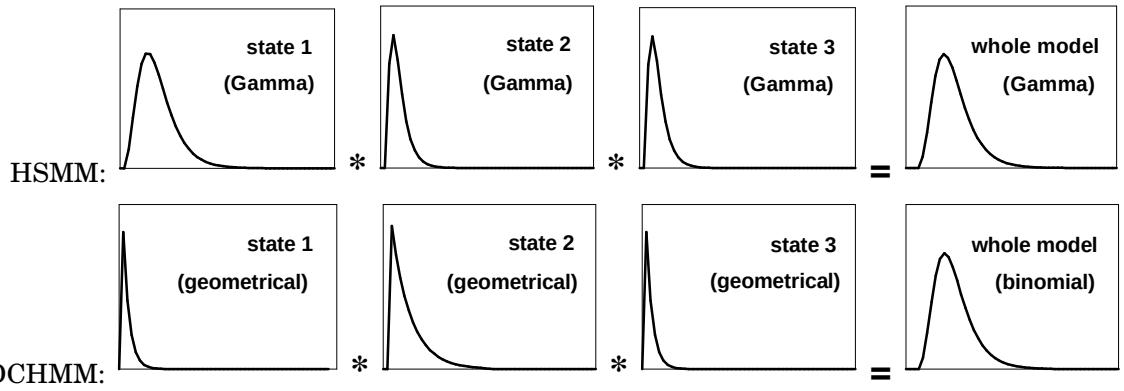


Figure 5.2 Convolutions (*) of state pdf's give whole-model pdf's, for HSMM and DCHMM.

respectively. Each term is maximised separately with a unity constraint, given by the properties of the probability measures (first row in Table 5.1), which, e.g., for A of linear models is given in (3).

For HSMM (e.g., Levinson, 1986), the additional state-duration term (e.g., the Gamma pdf) also contributes to the auxiliary function, and should also be maximised using its own constraint (also being unity, second row of Table 5.1). The approach of Hochberg & Silverman (1993) uses HSMM (with Gaussian state pdf) additionally constrained by the phone durational statistics σ^2 (we call them durationally constrained HSMM or DCHSMM, the third row). In our approach (DCHMM), since the modelled duration (μ_m, σ_m^2) is given by the A-parameters of HMM alone (relations (1)), the phone durational statistics (μ, σ^2) is used to constrain the A-parameters (the fourth row).

An HSMM has suitable pdf *forms* at both state and model levels; however, the pdf at model level is usually not durationally constrained. A DCHMM has an unsuitable geometrical pdf at each state, but the model-level pdf both has a suitable form, and is durationally constrained (illustrated in Figure 5.2).

5.3.2 Embedded training with extra durational constraints

The whole procedure for incorporating the knowledge about segmental duration (μ, σ^2) into the DCHMM consists of the following two steps.

Step one: choose the most suitable length *n* for each phone in the system. If we used relation (2) to force $a_{ii} = a$, both n and a would only be allowed to be a fixed value:

$$n = \frac{\mu^2}{\mu + \sigma^2}, \qquad a = \frac{\sigma^2}{\mu + \sigma^2}.$$

We know that n must be an integer, as it represents the number of selfloops. However the n solved here based on the data (μ, σ^2) might not coincidentally be an integer. Therefore we must allow a_{ii} to be different. On the basis of the relations between n, a, μ_m and σ_m^2 in (1) for different a_{ii} and for a given pair (μ, σ^2) to be fitted by (μ_m, σ_m^2), the choice of the length n of the linear model is confined within a range (Appendix 5.1), instead of being set to a fixed value. This range is further shrunk when we use a particular way of numerical search (Appendix 5.3). The range is represented by three quantities $\tilde{n}_{\min}$, $n_{\max}^L$ and $n_{\max}^U$(defined in Appendices 5.1 and 5.3), for different situations:

$$\begin{aligned}
\tilde{n}_{\min} &= \frac{\mu(\mu - 1) + \sigma^2}{\mu - 1 + \sigma^2}; \\
n_{\max}^L &= \mu + 1 - \sqrt{2\sigma^2 + 1}; \\
n_{\max}^U &= \mu + \frac{1}{2} - \sqrt{\sigma^2 + \frac{1}{4}}.
\end{aligned}$$

With the given data (μ, σ^2) for a phone, if $\tilde{n}_{\min} < 3$, it is mostly true that $n_{\max}^L < 3$ and $n_{\max}^U > 3$. For such a situation, we choose $n = 3$, i.e., the upper limit $n_{\max}^U$ is used. For other situations where $\tilde{n}_{\min} > 3$, it is mostly possible to choose n within the limit: $\tilde{n}_{\min} \le n < n_{\max}^L$. If for some phones neither of the above two situations is true for the given data (μ, σ^2), we can only fit the HMM modelled (μ_m, σ_m^2) to some (slightly) modified values $(\hat{\mu}, \hat{\sigma}^2)$, which is nevertheless better than without durational fitting at all. The above three limiting values are calculated again based on $(\hat{\mu}, \hat{\sigma}^2)$ and a new suitable n is chosen in the same way. For some phones several iterations are needed before reaching a final n. In all cases, the smallest possible integer n is chosen within the particular range, for the simplicity of the system. The whole above procedure and explanations for choosing n are detailed in Appendix 5.1 and 5.3.

Step two: Baum-Welch training of DCHMM with extra constraints. Similar to the way presented in Section 5.2, the $a_{i,i+1}$ are eliminated from the formulae for linear models. The set of equations obtained using the Lagrange multipliers (e.g., Wang et al., 1979) for constrained optimisation concerning the A parameters, but now with extra constraints cast by (μ, σ^2), are

$$\begin{cases}
\dfrac{D(i,i)}{\tilde{a}_{ii}} - \dfrac{D(i,i+1)}{1 - \tilde{a}_{ii}} + \theta_1 \dfrac{1}{(1 - \tilde{a}_{ii})^2} + \theta_2 \dfrac{1 + \tilde{a}_{ii}}{(1 - \tilde{a}_{ii})^3} = 0, i = 1, 2, \cdots n; \\
\displaystyle\sum_{i=1}^{n} \frac{1}{1 - \tilde{a}_{ii}} = \mu; \\
\displaystyle\sum_{i=1}^{n} \frac{\tilde{a}_{ii}}{(1 - \tilde{a}_{ii})^2} = \sigma^2,
\end{cases} \tag{6}$$

where $\tilde{a}$ are new values to be sought after the current iteration, θ_1 and θ_2 are the two multipliers, and $D(i,j)$ are the same as in Section 5.2. The difference from the standard Baum-Welch procedure for HMM is that, after each iteration, instead of putting $D(i,j)$ into the re-estimation formula to get the new parameter values directly as in (5), we now have to put $D(i,j)$ into a set of $n+2$ non-linear equations, for DCHMM. The updating of other parameters of DCHMM at each iteration remains the same as HMM.

It turned out that this set of non-linear equations about $\tilde{a}$ cannot be solved analytically to give formulae for calculating the new A values from old ones, as (5) in Section 5.2. We have chosen to use a Newton-Raphson (Press et al., 1989) iteration procedure to find numerical solutions (Appendix 5.2) with some initial points chosen on the basis of data constraints (μ, σ^2) (Appendix 5.3). The following set of $2n$ inequalities further constrain the iteration procedure to find only values of $\tilde{a}$ meaningful as a probability[2]:

$$\begin{aligned} &\tilde{a}_{ii} > 0, \\ &\tilde{a}_{ii} < 1, \end{aligned} \qquad i = 1,2,\cdots n. \tag{7}$$

We put the detailed procedures to establish the set of equations including (6) and the constraints (7), and to solve them, in Appendix 5.2. In general, the procedure searches for improvement of solutions starting from the points of the current iteration, based on the local derivatives of the set of equations.

5.4 Results

In this section, results on the improvement by durationally constrained training, both in modelling of segmental duration, and in performance of automatic recognition and segmentation, will be presented.

5.4.1 Results of durational pdf fitting

Before the recognition and segmentation tests, the effect of the durational constraint on the modelled durational pdf of the HMM is checked. In this subsection, we used the training set (3,696 utterances) of the TIMIT database (see Zue et al., 1990, and Appendix 2.1) to collect the durational statistics.

The durational histograms for all the 50 reduced TIMIT phones (see Chapter 3) are estimated from the training set. The statistics pair (μ, σ^2) is calculated and the allowed range of n for each phone is calculated based on (μ, σ^2), and the suitable lengths n for all the phones are chosen (Appendix 5.1). These n range from 3 to 10 for the whole system. Table 5.2 shows

[2]Since closed-form solutions to the constrained optimisation equations exist for the case of standard Baum-Welch procedure, there is no need for a numerical search which might lead to meaningless solutions, thus there is no need for an explicit 'meaningful' constraint since it is implicit from the closed form solution.

Table 5.2. The first column from the left shows the symbols of 8 example phones. The second column lists the durational μ and σ directly calculated from the hand-segmented label files of the TIMIT training set. The third column shows those necessarily modified target values. The fourth column shows the allowed range for choosing a suitable n and the actual chosen n for each phone. The last two columns show the durational μ_m and σ_m calculated from the models trained without (HMM) and with (DCHMM) the durational constraints, respectively. The two shaded rows for /ih/ and /q/ show the situation when DCHMMs are not durationally fitted. All the μ and σ values are presented in time steps (of 8 *ms*).

	original data		modified		after data modification				HMM		DCHMM	
	μ	σ	$\hat{\mu}$	$\hat{\sigma}$	$\tilde{n}_{\min}$	$n^L_{\max}$	$n^U_{\max}$	n	μ_m	σ_m	μ_m	σ_m
aw	20.09	6.40			7.07	11.98	14.17	8	20.07	5.99	20.09	6.40
b	2.18	0.89	3.50	0.85	2.94	2.94	3.02	3	3.64	0.92	3.50	0.85
ih	9.80	3.65			4.50	5.54	6.61	5	8.58	3.60	8.63	3.64
pau	23.44	15.63			2.89	2.31	8.30	3	21.72	13.33	23.44	15.53
q	7.75	3.76		4.13	2.91	2.82	4.08	3	6.32	2.66	6.31	2.65
sh	14.77	3.88			7.59	10.20	11.36	8	16.71	4.51	14.78	3.88
y	6.81	3.10		3.41	2.93	2.88	3.86	3	10.38	5.15	6.81	3.41
z	10.47	3.89			4.64	5.88	7.05	5	12.20	4.65	10.47	3.96

examples of the situations of the modelled durational statistics calculated from the HMMs and the DCHMMs, for 8 phones. For easy references to an accompanying figure to come, σ instead of σ^2 is used in the table.

Because of the numerical values of (μ, σ^2) of all the 50 phones from the original data, it is impossible for some phones to have a suitable n. Therefore for these phones, slightly modified values $(\hat{\mu}, \hat{\sigma}^2)$ had to be used as the target of durational fitting. First, we decide to have all $n \geq 3$ (Appendix 5.3). Then, a simple modification was made to increase those $\mu < 3$ to $\hat{\mu} > 3$. The remaining modification was only applied to σ^2, based on an observation that a slightly different $\hat{\sigma}^2$ will have a smaller effect on the precision of fitting than would a different $\hat{\mu}$. Since n is fixed, σ^2 can only be modified within a range (Appendix 5.3). With all these modifications, for 8 out of the 50 phone HMMs it was still impossible to fit their durational pdf's, because non-linear equations (6) do not always have solutions (2 of them, /ih/ and /q/, are shown in Table 5.2). For these phone HMMs, the parameters were updated after the iteration using the formulae of the conventional HMM as in (5).

The following details can be seen from Table 5.2:

1. For those phones (e.g., /b/) with $\mu < 3$, a modified $\hat{\mu} = 3.5$ was used and σ was modified accordingly to $\hat{\sigma}$, in order to be able to choose an $n \geq 3$;
2. For /q/ and /y/, since no suitable $n < n^L_{\max}$ can be found based on the original data μ and σ, the modified (increased) $\hat{\sigma}$ were used;
3. For phones /b/, /pau/ (pause), /q/ (glottal stop) and /y/, which require an $n = 3$, the upper limit $n^U_{\max}$ was used (true for all phones with $n = 3$);
4. For those phones that require $n > 3$, some have a small range $(\tilde{n}_{\min}, n^L_{\max})$ for choosing n (e.g., /z/), while others have a larger one (e.g., /aw/);
5. Both the modelled μ_m and σ_m of the HMMs (trained *without* durational constraints) show various degree of deviations from the data (μ, σ^2) or

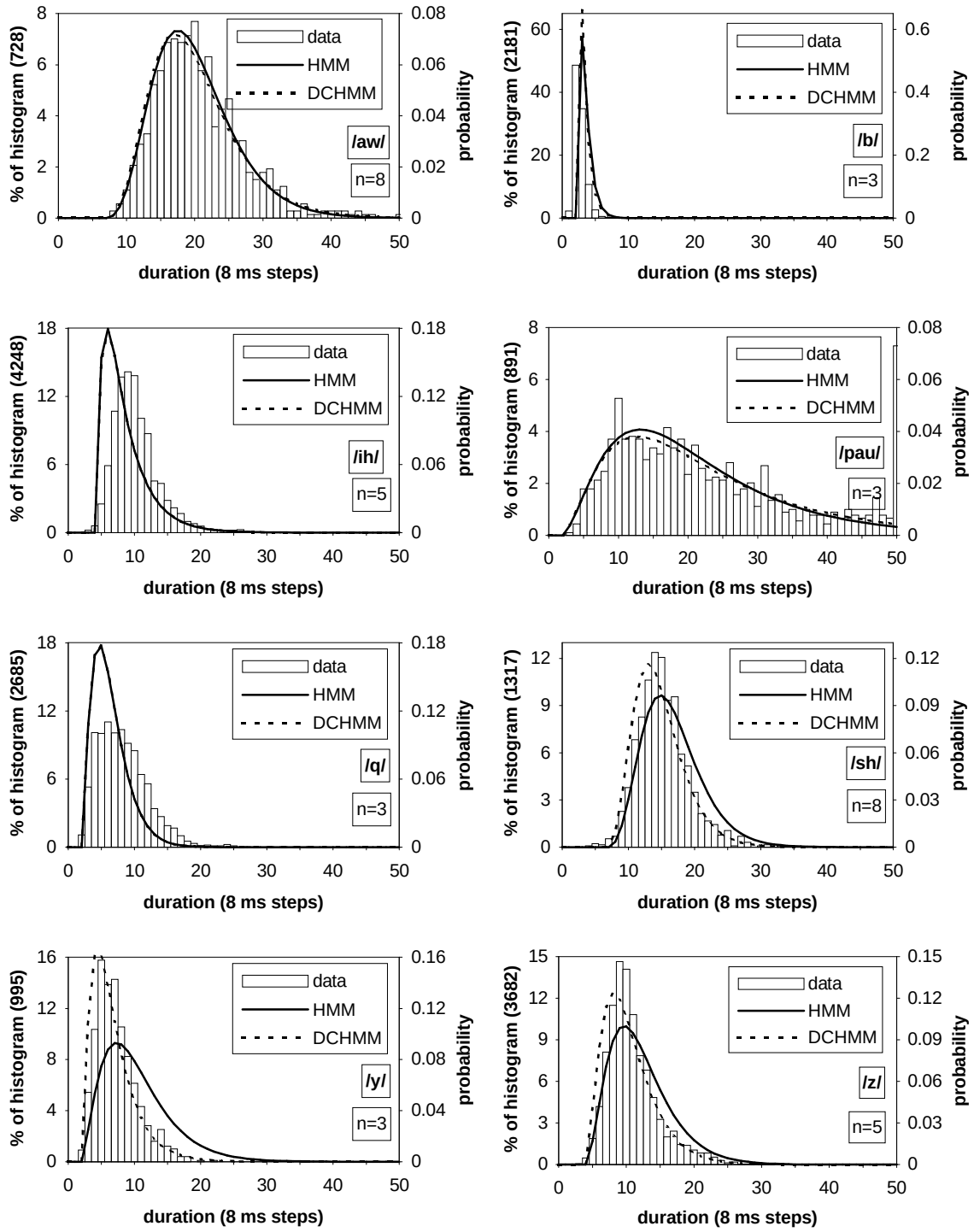


Figure 5.3. Examples of phone durational pdf's from the TIMIT training set: Data histograms (data: vertical bars) are obtained from the number of instances given between the brackets. Pdf's of HMM (continuous curve) and DCHMM (dashed curve) are scaled to the histogram. Histograms are plotted against the left scale, and pdf's are plotted against the right scale, of each panel. The chosen optimal model length n are also shown.

$(\hat{\mu}, \hat{\sigma}^2)$, the worst being the semi-vowel /y/ (see also Figure 5.3), indicating that the fitting is advantageous;

6. The modelled (μ_m, σ_m^2) of all the 6 fitted DCHMMs (trained *with* durational constraints) show good agreement with data (μ, σ^2) or $(\hat{\mu}, \hat{\sigma}^2)$.

Note that we have only used two lower-order statistics μ and σ^2 to fit the durational pdf[3]. It is shown in Figure 5.3 that for 6 out of the 8 phones, which found solutions in the numerical search, the fitting of their DCHMM pdf's is better than the HMM pdf's (and for /sh/, /y/ and /z/, much better).

Figure 5.3 shows that both the pdf's of the HMMs and DCHMMs fit well to the *shape* of the data histograms, indicating that the *form* (weighted sum of binomial/geometrical terms, see Chapter 4) of the pdf of linear models is suitable. It is evident then that the HSMM is unnecessary, because at the level of the whole model, a simple linear HMM can model the duration well. Among these HMMs, however, some (e.g., for /y/) have a very bad match with the durational histograms, even with the well chosen n. Furthermore, it seems true that those fitted DCHMMs always fit the data histograms better as compared with the HMMs (for those phones whose durational fitting is already good without the constraints, the improvement is of course small).

The improvement in durational fitting is an indication of the improvement of the quality of the DCHMM as compared with the HMM, at least with respect to the accuracy of whole-model duration modelling.

5.4.2 Results of recognition and automatic segmentation

The ultimate goal of duration modelling is to see if a more accurate modelling improves the *performance* of speech recognition and segmentation. Technically, during the durationally constrained Baum-Welch training, the constrained values of the A-parameters obtained from each iteration will be used in the next iteration, in which both the A- and the B-parameters will be affected by the durational constraints. (In this way B-parameters are *indirectly* affected). The new values of A and B of the DCHMM together may have a *different* general behaviour (not only concerning the transitions) from that of HMM parameters. Therefore, in recognition and segmentation, the performance may be different (hopefully improved with durational constraints). In this section, this effect will be checked experimentally.

In our experiments of both recognition and segmentation, the training set (3,696 utterances) of the TIMIT database was used for training, and its test set (1,344 utterances) for testing. TIMIT has been used for various related research topics, e.g., speaker identification (Lamel & Gauvain, 1993a) and phone recognition (Lamel & Gauvain, 1993b).

Our recogniser uses 50 context-independent phone models with a linear transitional topology (Chapters 3 and 4). 12 MFCC coefficients plus their

---

[3]Better approximation can be achieved by including, e.g., a constraint on the third moment $E(d-\mu)^3 = \sum_i [a_i(1+a_i)/(1-a_i)^3]$, which is a measure of the *skewness* of the pdf.

Table 5.3. Scores of recognition and segmentation using HMM and DCHMM, for speaker-independent tests on the TIMIT database. The HMMs have either a fixed length n=3, or a varying n suitable for duration modelling. For recognition at both word and phone levels, both correct and accuracy (between brackets) scores are shown. For segmentation, the percentages are given on the correctly matched phone segment borders within the threshold of 10 or 20 *ms* in both directions, as compared with the hand labels.

		HMM (n=3)	HMM (suitable n)	DCHMM
recognition	word	86.62% (85.23%)	84.17% (82.51%)	84.31% (82.60%)
	phone	70.73% (65.68%)	70.74% (66.02%)	70.77% (65.99%)
segmentation	10 *ms*	56.2%	60.0%	60.5%
	20 *ms*	84.6%	86.9%	87.1%

energy, and their Δ and $\Delta\Delta$ are used in one stream for the HMM. The frame-shift is 8 *ms*. (This is why in the pdf plots in this chapter the time scale is 8 *ms* while those in Chapter 4, the time scale was unspecified). The observation probability of each state has a weighted mixture of 8 Gaussian densities, each with a diagonal covariance matrix. The pre-processing without any linear transformation is used (the "original" in Chapter 3). The model lengths n chosen for all the 50 phones as described in the previous subsection are kept during the whole test. Then two setups of training are performed, for HMM and DCHMM, respectively. The same set of HMMs (with their parameters) after the *initialisation training* are used for both setups.

The language model used for recognition is a *regular* type of word-pair grammar estimated from both training and testing sets, the same as in Subsection 3.5.5. The word pronunciation dictionary is based on the lexicon.

The recognition scores are shown in Table 5.3. It can be seen that the scores of both word and phone recognition are only slightly improved from HMM to DCHMM cases, both of which use HMMs with suitable lengths n. Another test was performed using a set of HMMs, each with a fixed length n=3. The word recognition scores from this set of HMMs are higher than with the duration modelling in this chapter. Some explanations for this unfortunate situation are as follows.

The model length n for a phone is chosen based on (μ, σ^2). This n (minimal duration), together with the trained transition probabilities, will fit (μ, σ^2) well in a global way, but some phone instances with shorter duration $d < n$ may be outlayers of the fitted durational pdf. When such phone instances do occur in the test set, they are not allowed in the recognition process. Adding skip transitions may be a solution to satisfy both minimal duration and the fit to (μ, σ^2); however, the whole procedure using the current approach becomes unmanageably complicated. First, because there is no manageable procedure for choosing the model length n based on (μ, σ^2), as was the case for linear models in Appendix 5.1. Second, because the relations between the parameters of models with skips and the modelled durational statistics are too complicated (Appendix 4.1) to be used in the constrained training in this chapter. The technique presented in this chapter is only feasible for linear

models, however the linear models cannot satisfy the minimal duration. Some procedures that compromise between the complexity and fitting accuracy for models with skips may be topics for future studies along this line.

Another technical reason for the unsatisfactory scores is, as seen from the previous subsection, that for some phones, modified durational statistics had to be used as the target of durational fitting. Even with this relaxation, 8 HMMs out of the 50 did not get their durationally fitted parameters. Given such a quality of the durational fitting, it is difficult to improve the recognition performance.

We have anticipated that a more accurate modelling of phone duration may lead to a better performance of automatic phone segmentation, since in such a task the duration of the phones may have a more direct relation with the performance due to the smaller number of language model constraints as compared to word recognition. In the literature, various techniques have been applied to automatic segmentation (Cosi et al., 1991; Brugnara et al., 1993; Ljolje & Riley, 1991; Pauws et al., 1994; Vorstermans et al., 1995 & 1996).

Performing segmentation in our system is simply a recognition process with the "language model" set as an exact linear sequence of the phones in each sentence (the forced Viterbi procedure). The difference from a phone recognition process is that the identities of the phones in the sequence are known *a priori* and only the timing information is missing. The same three sets of HMMs were used. The scores (Table 5.3) show that both using a varying suitable model length, and applying a further constrained training (DCHMM), indeed improved the segmentation performance. The total improvement is significant with 95% confidence (Chollet, 1995).

5.5 Conclusions

In this chapter, a technique is implemented to train linear HMMs without explicit state durational pdf's by including additional constraints on the phone durations. The durational statistics of the phone used are context-independent, and the HMMs are monophones. The choice of linear model topology is based on the simple relations between the phone durational statistics and the model transition probabilities. The constrained training is embedded in the Baum-Welch algorithm to retain the optimality on the acoustic observations as well as achieving durational fit. We realised that, prior to training, the length of each linear phone model has to be well chosen, also based on durational statistics. Numerical solutions for the constrained training are obtained for the majority of the phone models.

The purpose of the durationally constrained training in this chapter is to obtain insight into whether such a way of incorporating context-independent durational knowledge into the HMM system can improve the system performance. The phone durational pdf's of the HMMs successfully trained

with durational constraints show a much better fit with the phone durational histograms than HMMs trained without durational constraints. This better quality in duration modelling at the context-independent phone level is reflected in the improved performance in automatic phone segmentation. The first step in the procedure of choosing a variable model length suitable for each phone improved segmentation, and the second step of the constrained Baum-Welch training improved it further. However, this improved duration modelling is not propagated to the level of recognition performance. For phone recognition, the scores increased only a very small amount at both steps of the procedure. For word recognition, the first step of using suitable model lengths even degraded the performance as compared with the scores using a fixed length 3 for all phone models.

The results in this chapter showed a possibility for incorporating extra knowledge about phone duration into the *existing structure* of the standard HMMs. The inclusion of the extra knowledge is also implemented within the standard mathematical treatment for the HMM training procedure. This makes the whole approach somehow compact and easy to compare with the standard HMM. However, an apparent disadvantage of the current approach, also seen from the unsatisfactory results in score improvements, is the extra "burden" on the HMM parameters. Because the *number* of model parameters (model structure) is not increased in the step of constrained training, the same set of parameters has to take care of the additional durational behaviour at the phone level. The HMMs trained without extra durational constraints may already have used all the parameters in an optimal way for acoustic modelling. When the *values* of these parameters are optimised in both durational aspects and all the other aspects, the already optimised acoustic modelling may somehow be sacrificed. Problems like this can only be solved by introducing additional parameters into the HMMs or the recogniser (Chapter 7). Nevertheless, this problem is only identified after we implemented the technique in this chapter. Without having done this, one might expect better results from such an efficient use of the parameters.

Further detailed limitations cast by the current approach, which can inspire future studies, are as follows. Context-independent durational measures and modelling may only capture a small portion of the duration-related information, as presented in a multi-speaker continuous speech database. The insufficient context-independent duration modelling in this chapter might be the major limiting factor in improving the performance. Contextual factors on durational distributions should be incorporated using entirely different techniques (e.g., Chapter 7). The choice of linear models makes the current approach feasible; however, it does not satisfy the minimal duration of the phones, thus it also hinders the improvement.

Appendix 5.1 Choice of the model length

Based on (μ,σ^2) from speech data for a phone, only *some* combinations of $(n,\{a_{ii}\})$ can fit (μ,σ^2). The selfloop probabilities $\{a_{ii}\}$ will be sought in the training step, however the value n, which allows for a fit by $(n,\{a_{ii}\})$, has to be chosen in advance. In this appendix we leave $\{a_{ii}\}$ as variable, and find the limits for possible values of n. In the discussion, n is relaxed to any real value and in the end, an integer must be taken as number of selfloops.

For convenience of thinking in the following discussion, we use a monotonic (thus one-to-one) transformation

$$u_i = \frac{1}{1-a_{ii}}, \tag{8}$$

with $u_i > 1$. The general relation in (1) is then written as

$$\sum_{i=1}^{n} u_i = \mu, \sum_{i=1}^{n}\left(u_i - \frac{1}{2}\right)^2 = \sigma^2 + \frac{n}{4}. \tag{9}$$

It can be seen from this set of equations that for a given n, the set of values $\{u_i\}$ that satisfy both μ and σ^2 lie on the n-dimensional hyper-circle defined by the intersection between a hyper-plane (the first equation in (9)) and a hyper-sphere (the second equation) centred at $\{½,½,\cdots ½\}$, in the space $\Re^n$ for $\{u_i\}$. In the following we will use this space to imagine the relations between the variables.

The *radius* of the sphere is controlled by n and σ^2, whereas the *intercepts* of the plane are defined by μ. For given (μ,σ^2), different values of n give rise to different situations for the intersection circle. Among all the situations, we identify three typical ones corresponding to three limit values for n.

The first situation, corresponding to the smallest possible $n_{\min}$, is signalled when the sphere is *tangent* with the plane at a single point, namely when all u (or equivalently a) are equal. By setting $u_i = u$ and eliminating u from (9), it is easy to verify that

$$n_{\min} = \frac{\mu^2}{\sigma^2 + \mu}. \tag{10}$$

The second situation applies when n increases (thus the sphere larger) so that the intersection becomes a hyper-circle, but this increase is bounded by the condition required for $u_i > 1$ (see (8)) when a lower bound of the maximum $n_{\max}^L$ is defined[4]. In this situation, all the points on the circle are solutions of (9). When n further increases beyond $n_{\max}^L$, some points on the hyper-circle will cause some $u_i < 1$, thus they are no more solutions. The solutions left are n disjoint pieces of arcs on the circle.

[4]We use superscripts L and U for lower and upper bounds, respectively.

The third and extreme situation when there still are solutions, is when these n arcs shrink into n isolated points. This corresponds to the largest $n = n^U_{\max}$ that can provide a solution.

We first find $n^U_{\max}$. The n points on the largest possible circle form an n-dimensional equilateral polygon. The co-ordinates u_i of $\boldsymbol{u}$ for each endpoint of this polygon have a pattern that $(n-1)$ components are equal to 1 while one $u = \mu - (n-1)$. These $\boldsymbol{u}$ are $\{\mu-(n-1),1,1,\cdots,1\}$ and $\{1,\mu-(n-1),1,\cdots,1\}$, etc. Putting any such point into the second equation of (9) we then get

$$(n-1)\left(1-\frac{1}{2}\right)^2+\left[\mu-(n-1)-\frac{1}{2}\right]^2=\sigma^2+\frac{n}{4}.$$

With given (μ,σ^2), we solve for the largest possible n (for the largest sphere)

$$n^U_{\max}=\mu+\frac{1}{2}-\sqrt{\sigma^2+\frac{1}{4}}.$$

From here on we get $n^L_{\max}$ for a complete solution hyper-circle. One point on such a hyper-circle is tangent with the middle point on one of the edges of the polygon, defined by the two end points of the edge:

$$\mathbf{u}=\left\{\frac{\mu-(n-1)+1}{2},\frac{\mu-(n-1)+1}{2},1,1,\cdots,1\right\}.$$

Putting this point into (9):

$$2\left(\frac{\mu-n+2}{2}-\frac{1}{2}\right)^2+(n-2)\left(1-\frac{1}{2}\right)^2=\sigma^2+\frac{n}{4}.$$

Again, with a given pair (μ,σ^2) this gives us the 'safest' largest n with which all the points on the intersection circle are solutions for (9):

$$n^L_{\max}=\mu+1-\sqrt{2\sigma^2+1}.$$

From above we observe that an ill-behaved hyper-circle given by $n^L_{\max} < n < n^U_{\max}$[5] can bring a numerical search from a solution to a non-solution point along the hyper-circle. Therefore in order to prevent numerical problems caused by this reason, one should preferably choose $n^L_{\max}$, rather than $n^U_{\max}$, when (μ,σ^2) allows.

[5]It can be proven that $n^L_{\max} < n^U_{\max}$: They both are monotonically decreasing functions of σ^2, both evaluate μ at 0, and zero-cross at $\mu(\mu/2+1) < \mu(\mu+1)$, respectively.

Appendix 5.2 Non-linear ML equations and their solution

Generally, the Newton-Raphson method (e.g., Press et al., 1989) searches for numerical solutions for N variables y_i given in N non-linear equations

$$f_i(y_1, y_2, \cdots, y_N) = 0, \qquad i = 1,2,\cdots,N. \tag{11}$$

This is achieved by using the current values of f_i and their partial derivatives to form a set of linear equations about the local increments δy_i:

$$\sum_{j=1}^{N} \frac{\partial f_i}{\partial y_j} \delta y_j = -f_{i,} \qquad i = 1,2,\cdots,N. \tag{12}$$

The solution for δy_i can be obtained by any standard method for linear equation systems, such as LU-decomposition. Then y_i values are updated as

$$y_i^{new} = y_i^{current} + \delta y_i, \qquad i = 1,2,\cdots,N.$$

The iteration starts at some chosen initial point and ends when some convergence threshold is reached.

Our particular set of non-linear equations come from the Baum-Welch ML (maximum-likelihood) parameter estimation procedure. We consider the part of the *auxiliary function* (e.g., Kamp, 1991) concerning the A parameters for the linear DCHMM with extra durational constraints (in terms of (μ, σ^2)) in ML. Any fixed state i only transits into two states i and $i+1$. Using the unity constraint $\tilde{a}_{i,i+1} = 1 - \tilde{a}_{ii}$, the auxiliary function is

$$F = \sum_r \sum_t \sum_i \left[\gamma_{t-1}^r(i,i) \log \tilde{a}_{ii} + \gamma_{t-1}^r(i,i+1) \log(1-\tilde{a}_{ii})\right] + \\ + \theta_1 \left(\sum_i \frac{1}{1-\tilde{a}_{ii}} - \mu \right) + \theta_2 \left(\sum_i \frac{\tilde{a}_{ii}}{(1-\tilde{a}_{ii})^2} - \sigma^2 \right),$$

where $\tilde{a}_{ii}$ are the new values of selfloop probabilities after the current iteration, θ_1, θ_2 are two Lagrange multipliers. Further constraints for the numerical search to be confined within the meaningful region (in the a-space) may be written as $2n$ negative functions

$$g_k = \begin{cases} -\tilde{a}_{kk} < 0, & k = 1,\cdots,n; \\ \tilde{a}_{k-n,k-n} - 1 < 0, & k = n+1,\cdots,2n. \end{cases}$$

Introducing some positive relaxation functions (e.g., Wang et al., 1979)

$$s_k = x_k^2 + \varepsilon, k = 1,\cdots,2n,$$

where $\varepsilon > 0$ is a small constant that guarantees $s_k > 0$. This brings the constraints into equation form:

$$\varphi_k = s_k + g_k = \begin{cases} x_k^2 + \varepsilon - \tilde{a}_{kk}, & k = 1,\cdots,n; \\ x_k^2 + \varepsilon + \tilde{a}_{k-n,k-n} - 1, & k = n+1,\cdots,2n. \end{cases}$$

Now the new auxiliary function including all the constraints becomes

$$\Phi = F + \sum_{k=1}^{2n} \lambda_k \varphi_k .$$

To get the critical point for the ML of Φ, we take the partial derivatives with respect to the $N = 5n+2$ variables y_i, namely n of $\tilde{a}_{ii}$, 2 of θ, $2n$ of x_k and $2n$ of λ_k, respectively, and let them be zero, resulting in a total of $5n+2$ non-linear equations $f_i = 0$. The specific forms of these different f's are:

$$\begin{aligned} f_i &= \frac{1}{\tilde{a}_{ii}} D(i,i) - \frac{1}{1-\tilde{a}_{ii}} D(i,i+1) + \theta_1 \frac{1}{(1-\tilde{a}_{ii})^2} + \\ &+ \theta_2 \frac{1+\tilde{a}_{ii}}{(1-\tilde{a}_{ii})^3} - \lambda_i + \lambda_{n+i}, i = 1,2,\cdots n; \\ f_{n+1} &= \sum_{i=1}^{n} \frac{1}{1-\tilde{a}_{ii}} - \mu; \\ f_{n+2} &= \sum_{i=1}^{n} \frac{\tilde{a}_{ii}}{(1-\tilde{a}_{ii})^2} - \sigma^2; \end{aligned}$$

$$\begin{aligned} f_{n+2+k} &= 2\lambda_k x_k, && k = 1,2,\cdots,2n; \\ f_{3n+2+k} &= \begin{cases} x_k^2 + \varepsilon - \tilde{a}_{kk}, & k = 1,2,\cdots,n; \\ x_k^2 + \varepsilon + \tilde{a}_{k-n,k-n} - 1, & k = n+1,\cdots,2n. \end{cases} \end{aligned}$$

For the first n functions f_i we used a notation

$$D(i,j) = \sum_r \sum_t \gamma_{t-1}^r (i,j)$$

for simplicity. To solve these equations we use the *linear* equations (12) about the increments δy. For clarity, we write (12) in matrix form

$$\mathbf{C}\delta\mathbf{Y} = -\mathbf{f}.$$

Here $\delta\boldsymbol{Y} = (\delta y_1, \delta y_2, \cdots, \delta y_{5n+2})^\tau$ (τ denotes transpose), $\boldsymbol{f} = (f_1, f_1, \cdots, f_{5n+2})^\tau$, and $\boldsymbol{C}$ is a (symmetrical) matrix formed by taking further partial derivatives $\partial f_i / \partial y_j$, $(i, j = 1,2,\cdots,5n+2)$,

$$\mathbf{C}=\begin{pmatrix} \mathbf{U} & \mathbf{V} & \mathbf{W} & . & . & \mathbf{X} & \mathbf{Z} \\ \mathbf{V}^{\tau} & . & . & . & . & . & . \\ \mathbf{W}^{\tau} & . & . & . & . & . & . \\ . & . & . & \mathbf{P} & . & \mathbf{R} & . \\ . & . & . & . & \mathbf{Q} & . & \mathbf{S} \\ \mathbf{X} & . & . & \mathbf{R} & . & . & . \\ \mathbf{Z} & . & . & . & \mathbf{S} & . & . \end{pmatrix},$$

where all the dots denote zero sub-matrices. The non-zero sub-matrices are specified, respectively, as

$$\mathbf{U}_{(n\times n)}=\operatorname{diag}\left\{-\frac{1}{\tilde{a}_{ii}^{2}}D(i,i)-\frac{1}{(1-\tilde{a}_{ii})^{2}}D(i,i+1)+\frac{2\theta_1}{(1-\tilde{a}_{ii})^{3}}+2\theta_2\frac{2-\tilde{a}_{ii}}{(1-\tilde{a}_{ii})^{4}}\right\}_{i=1,2,\cdots,n};$$

$$\mathbf{V}_{(n\times 1)}=\left\{\frac{1}{(1-\tilde{a}_{11})^{2}},\ \frac{1}{(1-\tilde{a}_{22})^{2}},\ \cdots,\ \frac{1}{(1-\tilde{a}_{nn})^{2}}\right\}^{\tau};$$

$$\mathbf{W}_{(n\times 1)}=\left\{\frac{1+\tilde{a}_{11}}{(1-\tilde{a}_{11})^{3}},\ \frac{1+\tilde{a}_{22}}{(1-\tilde{a}_{22})^{3}},\ \cdots,\ \frac{1+\tilde{a}_{nn}}{(1-\tilde{a}_{nn})^{3}}\right\}^{\tau};$$

$$\mathbf{X}_{(n\times n)}=-\mathbf{I}_n;\qquad \mathbf{Z}_{(n\times n)}=\mathbf{I}_n;$$
$$\mathbf{P}_{(n\times n)}=\operatorname{diag}\{2\lambda_k\}_{k=1,2,\cdots,n};\quad \mathbf{Q}_{(n\times n)}=\operatorname{diag}\{2\lambda_k\}_{k=n+1,\cdots,2n};$$
$$\mathbf{R}_{(n\times n)}=\operatorname{diag}\{2x_k\}_{k=1,2,\cdots,n};\quad \mathbf{S}_{(n\times n)}=\operatorname{diag}\{2x_k\}_{k=n+1,\cdots,2n},$$

where $\boldsymbol{I}_n$ is an $n\times n$ identity matrix, and the subscripts between brackets of the sub-matrices denote their dimensions.

Appendix 5.3 Initial points for numerical search

The numerical solutions of the equations in Appendix 5.2 require initial values of the variables. These initial values should also be chosen based on some relations between the variables, so that they themselves do not conflict. There are different ways of choosing the initial values. In this appendix we discuss a specific choice, which has been tested in the training procedure of this chapter.

A5.3.1 Initial points

For convenience of analysis we still use u as in (8). When $n=2$, the space is reduced to a plane[6] and the solution intersection for (9) given a pair (μ,σ^2) is reduced to the intersection between a 2-dimensional circle and a straight line, resulting in at most 2 points. This is logical since 2 equations for 2 variables

[6]We do not consider the case for $n = 1$ because it gives only a geometrical durational pdf.

will leave no freedom for relaxed solutions. It is easy to obtain the analytical solution of (9) for $n = 2$:

$$u_1 = \frac{1}{2}\left(\mu - \sqrt{2\sigma^2 + 2\mu - \mu^2}\right), u_2 = \frac{1}{2}\left(\mu + \sqrt{2\sigma^2 + 2\mu - \mu^2}\right)$$

Although a fixed solution can satisfy the durational constraints (9) (Appendix 5.1), there is little chance that this is, by coincidence, also the solution for the whole ML equations (6). This implies that in practice, if for some HMM the smallest integer n possible is really 2, one should take some value of $n > 2$ in order to let the searching procedure find solutions for the entire (6). In the following discussion, we will assume $n \geq 3$. From $u_i > 1$ and (9) it follows that we should have $\mu > n$. Then if for some phones $\mu < 3$, we have to use some $\hat{\mu} > 3$ instead as the target value for the fitting procedure (In this appendix we assume $\mu > 3$).

For the numerical search not being trapped into some bad point (e.g., local optima), we need to give some number of initial points and start searching from all these points. From (9) it is clear that permuting the components of $\boldsymbol{u}$ makes no difference for the durational constraints, but it does make a difference for the first equation in (6) which includes also the distribution of acoustic observations. When only one component in $\boldsymbol{u}$ is different while all other components are the same, we get only n initial points by permuting the components. We consider n points as insufficient and "design" some more points as follows. We take all the $n-2$ components of $\boldsymbol{u}$ to be equal, and another one to have a small difference $\delta > 0$:

$$u_3 = u_4 = \cdots = u_n = u_2 + \delta, \tag{13}$$

Then we find the last component on the intersection circle given these $n-1$ values. Putting this into the first equation of (9) we get

$$(n-2)u_n + (u_n - \delta) + u_1 = \mu.$$

From this we solve

$$u_n = \frac{\mu - u_1 + \delta}{n-1}. \tag{14}$$

Putting this and the u_{n-1} from (13) into the second equation of (9) we then get

$$(n-2)\left(\frac{\mu - u_1 + \delta}{n-1} - \frac{1}{2}\right)^2 + \left(\frac{\mu - u_1 + \delta}{n-1} - \delta - \frac{1}{2}\right)^2 + \left(u_1 - \frac{1}{2}\right)^2 = \sigma^2 + \frac{n}{4}.$$

Solving for u_1 and taking arbitrarily the higher value for convenience, we get:

$$u_1 = \frac{1}{n}\left[\mu + \sqrt{(\sigma^2 n + \mu n - \mu^2)(n-1) - \delta^2 n(n-2)}\right] \tag{15}$$

The above $\boldsymbol{u} = \{u_1, u_2, u_3, \cdots, u_n\}$ is only one initial point. Since 2 components have different values while all the others are the same, permuting these components will give us $n(n-1)$ initial points (The other $n(n-1)$ points obtained by taking a negative sign before the square root are not used).

The remaining problem is how to choose the value of δ. The condition is to guarantee all $u_i > 1$. We only have to guarantee the smallest component u_2 and other u will then also be guaranteed. Using (13) and (14) this gives

$$u_2 = u_n - \delta = \frac{\mu - u_1 + \delta}{n-1} - \delta > 1.$$

To find $\delta_{\max}$ of δ we combine this with (15) to eliminate u_1 and it follows

$$\left[\delta_{\max} n(n-2) + n(n-1) + (1-n)\mu\right]^2 = (\sigma^2 n + \mu n - \mu^2)(n-1) - \delta_{\max}^2 n(n-2).$$

Solving this and taking the smaller (safer) value, this gives

$$\delta_{\max} = \frac{1}{n-1}\left[(\mu - n) - \sqrt{\frac{\sigma^2(n-1) - (\mu - n)(\mu - 1)}{n-2}}\right]. \tag{16}$$

In practice we take some value $\delta < \delta_{\max}$ for getting the initial values $\{u_i\}$.

A5.3.2 Modified limit for n *and range for modifying variance*

The $\delta_{\max}$ is only meaningful if the argument of the square root in (16) is non-negative, and this casts another lower limit on n for a given σ^2:

$$\tilde{n}_{\min} = \frac{\mu(\mu-1) + \sigma^2}{\sigma^2 + \mu - 1}.$$

Comparing this with (10) we have $\tilde{n}_{\min} > n_{\min}$, which means that in practice we have to take an $n > \tilde{n}_{\min}$ in choosing n. (Recall that $n_{\min}$ refers to the case with equal u. This means then that in order to be able to use the initial points chosen this way, it is no longer allowed to have equal selfloop probabilities).

On the higher border of n, the data statistics pair (μ, σ^2) of some phones may not allow any $n < n_{\max}^L$ nor even $n < n_{\max}^U$. Therefore a reasonable compromise is to relax on the fitting of one of the two statistics, and preferably on σ^2 (the accuracy of it in modelling is less important than μ). Then we need to know the possible range within which σ^2 is allowed to vary, based on the given μ and a chosen n. Also using the constraints that the argument of the square root of $\delta_{\max}$ should be non-negative, but now with a fixed value n, we get the lower limit:

$$\sigma^2 \geq \frac{(\mu - n)(\mu - 1)}{n-1}.$$

The same as discussed in Subsection 5.3.2, the upper limit is different for the cases of $n = 3$ and $n > 3$ as required by the data μ, and the σ^2 value before the current modification. For $n = 3$, the expression of $n_{\max}^{U}$ should be used and it requires that

$$3 = n < n_{\max}^{U} = \mu + \frac{1}{2} - \sqrt{\sigma^2 + \frac{1}{4}}.$$

From this we solve for the upper limit of σ^2

$$\sigma^2 < (\mu - n)(\mu - n + 1) = (\mu - 3)(\mu - 2).$$

For $n > 3$, a similar procedure, but using the expression of $n_{\max}^{L}$, leads to another upper limit of σ^2

$$\sigma^2 < \frac{(\mu - n)(\mu - n + 2)}{2}.$$

The upper limits for both cases have a strict inequality, which have been required by the strict constraint $u_i = 1/(1 - a_{ii}) > 1$.

6

ANALYSIS OF PHONE DURATION IN *TIMIT* INFLUENCED BY CONTEXT, STRESS AND LOCATION*

Abstract

In this chapter we will analyse the phone duration as a function not only of the identity of the phone, but also of its context, stress and location within a word and within a sentence. The analysis is based on the TIMIT speech database, which consists of continuous read sentences from many speakers. The various effects on duration are investigated with the purpose of adding knowledge into a speech recogniser so that the performance of the recogniser may be improved. The complex nature of phone duration is illustrated with these speech data. Two methods of analysis are used. The first method illustrates directly the durational statistics of individual factors. The second method is an adapted analysis of variance *of all the 11 chosen factors.*

*The major content of this chapter has been published in a paper in Speech Communication (Pols, Wang & Ten Bosch, 1996).

6.1 Introduction

In the previous chapters, only context-free phone durations have been dealt with. This is reflected so far in two aspects of the techniques used in this thesis. First, the durational statistics are collected *separately* for each context-free phone segment. Second, the HMMs are defined on the *isolated* monophones.

However, the phone duration is actually *not* context-free. It is influenced by many factors which have complex relations with each other. Much research work has been done to reveal the complex nature of segmental duration (Nooteboom, 1970), and extensive data can for instance be found in Van Santen (1992b) and Van Santen & Olive (1990) about the various context effects on vowel duration. Such knowledge about context-dependent segmental duration can in principle also be used in the technical implementation of automatic speech synthesis (Van Heuven & Pols, 1993; Van Santen, 1992a & 1994) or recognition (Gong et al., 1994; Jones & Woodland, 1993; Monkowski et al., 1995; Picone, 1989). In recognition practice, what one usually does in incorporating durational knowledge is actually based on an *assumption*, that the phone duration bears information additional to the spectral information at the frame level. We also follow this assumption in the whole thesis work.

Factors such as local context, amount of stress, and specific location of a phone segment generally have certain effects on the *spectral* measures of speech. On the other hand, the same factors have also effects on *durational* distribution of the phone segments. So-called context-dependent HMMs, such as triphones, model all kinds of context-dependent variations *within* these triphone units. When we follow the aforementioned assumption, we can have a different modelling approach from the triphone. In our approach of this thesis, we will only use monophone HMMs to model the within-segment *spectral* variations, and based on that we will explicitly model the context-dependent *durational* variation using separate *duration models*. This way we hope to separate the context effects on durational variation from those on *spectral* variation. Our approach may lead to lower recognition performance than the triphone approach, due to lack of modelling of context-dependent spectral variations. However our approach may lead to insight into durational characteristics, being an interesting phonetic parameter by itself.

In natural speech, phone duration is influenced by various factors. However, the factors that are important and feasible to model will depend on specific engineering purposes. In Van Santen (1992a & 1994), the purpose was high quality speech synthesis by rule using the voice of a single speaker. Therefore, as many factors as possible should be taken into consideration to make the synthesised speech natural-sounding. The parameters for some factors may have to be estimated from very few realisations. Although these

samples are rare, they are important to model when such durational contexts do occur in the text to be synthesised. For the purpose of automatic speech recognition, on the other hand, contextual duration modelling should play a role in correcting the improperly modelled duration. The phone duration is already modelled to some extent by the context-free monophone HMMs (Chapter 4 and 5). So it is not like in a (simple) speech synthesiser where without a duration model, all phone segments would simply be produced with equal lengths.

In this chapter, we will search through all tangible effects on the phone duration, such as the identity of the near context phones, the stressing of the central phone, and the position of the central phone within the word and within the sentence utterance. We will deal with the various special problems that have to be solved when searching the regularities based on durational factors from a speech database. First we will describe the speech database with respect to contextual effects on duration. Then we will discuss the durational effects found in this database. Finally we will try to analyse all the relevant contextual durational factors in a unified framework. Although the purpose of our modelling is different from that in Van Santen (1992b) and in Crystal & House (1988a & 1988b), we will compare our data with theirs on a few points. The whole chapter only deals with analysis of durational distribution, as a preparation for speech recognition in the next chapter. Nothing about recogniser implementations will be discussed in this chapter.

6.2 Durational information in the TIMIT speech database

For the purpose of extracting durational information, we use the 3,696 `si` and `sx` utterances in the training set of the TIMIT database (see Appendix 1.1 for more details). The many different kinds of variations in such a medium-sized database with many (462) speakers will be illustrated in this section. First, we will describe the difficulty encountered in extracting contextual durational information from TIMIT, and some technical treatments necessary for overcoming this difficulty. Then we will illustrate some durational distributions.

6.2.1 Mismatch problem and Dynamic Programming

TIMIT comes with manual labelling at both phone and word levels for all the sentence utterances. For the purpose of context-dependent duration modelling, we need to have additional information about whether a syllable is stressed and where it is located in a word and in an utterance. With TIMIT, unfortunately, the hand labelling does not include the actually realised word stresses and sentence accent. The only stress information available is about the lexical stress (being primary or secondary). Although the lexical stress is

only part of the whole matter of stressing in speech, it is still better than no stress information at all. We will investigate the durational behaviour related to such stress information.

However, using the lexicon to find stress and location information introduces a problem, as explained below. Each TIMIT utterance is accompanied by a word-label file and a phone-label file, the latter having no information assigned about stress or syllable location. Our task of assigning the missing information for each utterance then consists of three steps:

1. Generate a *norm* phone sequence for the utterance by concatenating the phone sequences of all the words in the word-label file, based on the TIMIT lexicon;
2. For all the vowel phones, get the stress and location marks based on the lexicon and location of words in the utterance;
3. Copy all the marks from the norm phone sequence to the *actual* phone sequence in the phone-label file.

The problem is in step (3) because the matching between norm and actual phone sequences is far from one-to-one. The following is an example of the (mis)matches between the two phone sequences for the same utterance:

```
word: (si1865 by mptf0)    now   there's        nothin        left         of   me
norm:                   ns n aw1 dh ae1 r z     n ah1 th ix n l eh1 f cl t ax v m iy1 ns
actual:                 ns n aw1 dh ix1   z epi n ah1 th ix n l eh1 f cl t ax v m iy1 ns
mismatch                           s   d   i
```

where 's', 'd' and 'i' show substitutions, deletions and insertions, respectively[1].

Since a perfect match is impossible, we can only rely on an optimal match. For that we make use of a dynamic programming (DP, see Appendix 6.1) algorithm at the phone-label level. The two phone sequences in the preceding example are already DP-matched. It can be seen that vowels are mostly matched with (other) vowels[2] when they are themselves not deleted (due to, e.g., ellipsis) from the actual pronunciation. When such a vowel-to-vowel match is found (it can be a vowel reduction or other kinds of colouring), the actual vowel instance is used to collect durational statistics whereas its stress and location information from the norm form in the lexicon are used[3]. In this utterance, the stress mark '1' (primary) of /ae/ is copied to /ix/ (in bold face). If the actual vowel instance does not find a match with a vowel, that vowel instance is only used with 'unknown' stressing and location information[4].

By comparing the norm and actual labels for the whole TIMIT training set with DP, we can produce a "confusion" matrix. In Table 6.1, a part of such a matrix is shown (the whole matrix in shown in Appendix 6.2). For the whole data set, 78.2% phone instances are correctly matched (120,599 out of

[1]A legend of the TIMIT phone symbols can be found in section 5 of Chapter 3.
[2]This is due to the correctly matched neighbouring consonants with DP.
[3]Of the total of 15,545 deleted phones, 7,272 are non-released stop bursts, and 3,237 are the closure phones. This is a substantial part of all deletions (67.6%).
[4]There are 42,912 matched out of a total of 45,572 (94.2%) *vowel* instances after DP.

Table 6.1 Part of the confusion matrix between the phone symbols in the norm sequences (left column) and the actual symbols (upper row), for the TIMIT training set. Here 17 chosen phones are shown, of which the first 9 are vowels. The column with 'Del' shows the deletion count of each phone, whereas the last row 'Ins' shows the counts of insertion by all the phones. The right-most column with '%c' shows the percentage correctly matched score of each norm phone, over all the 50 actual phones.

	iy	ih	ix	uw	aa	ay	ow	er	y	r	l	n	s	dh	p	b	vcl	...	Del	%c
iy	3920	129	257	6				1	15										63	90.1
ih	173	3068	1833	2	1	2	1	43		1	2	2	4				2		513	54.5
ix	148	284	1686	1				13				22	1				2		345	60.8
uw	1	7	202	1907			2	9			1								170	81.5
aa		1	4		1860	9	1	94											17	86.8
ay	1	16	9		72	1913				1									11	93.3
ow			5	6	1		1607	2											8	93.0
er		8	51	3	1			2912		86									27	92.4
y	5	10	23	3				30	924										127	91.8
r	6	8	44		4			465	1	4504							1		735	87.6
l			11		3		2	1			4274								282	98.7
n	1	3	10		1			5			2	6652	3				2		541	96.9
s			3					3					5861						204	98.1
dh	27	6	53									12	1	2363					99	89.5
p			3					3							2586	3			345	98.5
b	1	5	14		1			7		2			2			2176			268	98.0
vcl	6	12	49		5			4	2	2	1	7	3	3			7079		1567	87.7
...																				
Ins	55	41	170	6	16	1	3	14	46	73	23	16	16	2	57	1	43			

154,133, the latter being the total number of phones in the norm transcriptions).

Actually it is very important that the contextual information of the majority of the *vowel* instances can be identified (Peterson & Lehiste, 1960) via DP, since in this chapter we mainly concentrate on vowel duration influenced by contexts. Only in Subsection 6.2.5, durations of consonants and other segments are also measured. However, consonant durations will not be considered to receive influence from contexts.

The manual labelling of the TIMIT database makes it possible to collect the durational statistics based on actually pronounced segments. In Chapter 5, when only context-free duration is needed, we do not need the lexicon and all the hand-labelled phone instances can be used. When we need contextual, stressing and location information about the segments, we need the lexicon and we have to face mismatching. Consequently, we can only use a sub-set of the phone instances, but not all of them, for collecting durational statistics.

6.2.2 Durational distribution affected by stressing and location

First of all, we realise that it is better to collect durational statistics for two sub-sets of vowels separately, i.e., for long and short vowels. This is because, on the one hand, different vowels show different behaviour, and on the other

Table 6.2 Short and long vowels, their example words and vowels in them printed in boldface, mean duration (in *ms*) and percentage of instances in the TIMIT training set.

	vowel	example	μ (*ms*)	% instances		vowel	example	μ (*ms*)	% instances
short:	iy	(b**ea**t)	95	10.2	long:	ae	(b**a**t)	136	5.0
	ih	(b**i**t)	78	9.3		aa	(c**o**t)	123	5.0
	eh	(b**e**t)	93	7.2		ao	(ab**ou**t)	123	4.1
	ix	(ros**e**s)	51	16.2		ey	(b**ai**t)	127	5.0
	ax	(th**e**)	47	8.5		ay	(b**i**te)	155	4.2
	ah	(b**u**tt)	89	5.0		oy	(b**oy**)	169	0.7
	uw	(b**oo**t)	100	4.3		aw	(b**ough**)	161	1.6
	uh	(b**oo**k)	76	1.1		ow	(b**oa**t)	128	3.6
	er[5]	(b**ir**d)	95	9.1					

Table 6.3 Mean vowel duration (in *ms*) of TIMIT training set, for the short and long vowels separately, in stressed and unstressed syllables, in word-final, non-word-final positions and in monosyllabic words. *n* shows number of tokens.

	short vowel				long vowel			
	unstressed		stressed		unstressed		stressed	
	ms	*n*	*ms*	*n*	*ms*	*n*	*ms*	*n*
word-final	76	6,966	115	1,390	134	88	148	1,659
non-word-final	57	5,635	85	5,313	114	446	124	4,904
monosyllabic word	54	3,507	86	9,458	94	212	142	5,994

hand, a complete separation down to the single-vowel-per-category level will lead to the problem of insufficient data. Grouping of the vowels is based on the context-free mean duration measured for each vowel. This happens to coincide nicely with the general definition of short and long vowels. These two vowel groups are listed in Table 6.2.

The most influential factors on vowel duration are stress and location. In Figure 6.1, the durations are longer for vowels appearing in the last syllable of the utterance (pre-pausal lengthening). This holds for both stressed and unstressed conditions, and for both short and long vowels. In Table 6.3, the effect of stressing on the mean vowel duration is shown for short and long vowels appearing in word-final positions (excluding those vowels in mono-syllable words), non-word-final positions and in mono-syllable words. In all these positions, the stressed vowels are longer than the unstressed ones. Reading the table vertically, one can see that the within-word vowel position is also a factor on duration. In both Figure 6.1 and Table 6.3, it is clear that the factor "phone category" (long and short) is also important.

The mean durations and the numbers of tokens used in Figure 6.1 for position of syllables within utterance are compared with the data of Crystal & House (1988b) in similar situations, shown in Table 6.4. The number of short vowel categories in their data is much smaller because the reduced vowels were excluded. The numbers of tokens, as obtained from 2 short scripts (33

[5]Statistic data in this chapter are slightly different from those published in Pols et al., (1996) for which /er/ had been put in the category of long vowel.

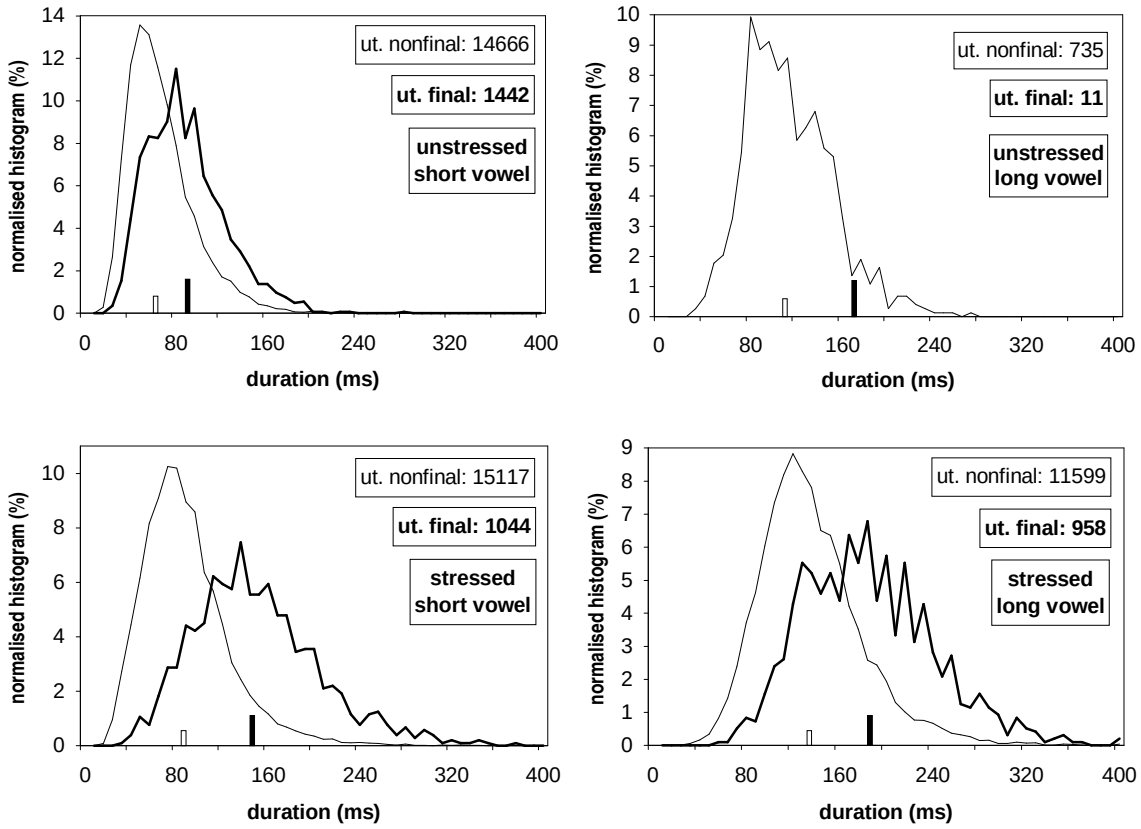


Figure 6.1 Effect of pre-pausal lengthening (utterance-final vs. non-final, dark line vs. thin line) and word stress (unstressed vs. stressed, upper panels vs. lower panels) on the duration (in *ms*) of short and long vowels (left vs. right panels). The numbers of phone instances in the training set are given. The mean durations for the two distributions in each panel are indicated by open or filled bars, respectively. The histogram for utterance-final unstressed long vowels is not plotted in the upper-right panel, because of the very small number of instances (11).

Table 6.4. Mean vowel duration in *ms*, for the short and long vowels separately, in stressed and unstressed positions, and in utterance-final (or pre-pausal) and non-final positions. Both the TIMIT data as well as data extracted from Crystal & House (1988b) are presented. The numbers of tokens involved (*n*) are indicated as well. Numbers of vowels in either long or short groups are shown between brackets.

	TIMIT short V (9)		Crystal & House short V (4)		TIMIT long V (8)		Crystal & House long V (7)	
	μ (*ms*)	*n*	μ (*ms*)	*n*	μ (*ms*)	*n*	μ (*ms*)	*n*
Unstressed	65	16,108	56	842	111	746	78	237
Stressed	88	16,161	93	601	136	12,557	141	1,091
utt. final unstr.	85	1,442	81	39	168	11	110	7
utt. final str.	145	1,044	147	78	181	958	202	125
utt. non-final unstr.	63	14,666	56	727	110	735	77	224
utt. non-final str.	84	15,117	85	253	132	11,599	134	628

sentences in total) read by 6 speakers, are also much smaller than TIMIT. Given these differences, the mean durations both in TIMIT and in Crystal & House (1988b) show quite a similar effect on utterance-final lengthening, stressing, and dividing into short- and long-vowel categories.

6.2.3 Effect of post-vocalic stops on vowel duration

In both classic literature (e.g., Klatt, 1976; Umeda, 1975) and recent data (Van Santen 1992b), the voicing of post-vocalic stops has a tangible effect on vowel duration (vowels followed by /p,t,k/ are shorter than those followed by /b,d,g/). However, using TIMIT data, if the whole set of 462 speakers is used, this difference almost disappears (upper panels of Figure 6.2, 461 and 455 speakers actually spoke the short or long vowels in the relevant contexts, respectively). Only a tiny difference in the tails of the curves may be seen,

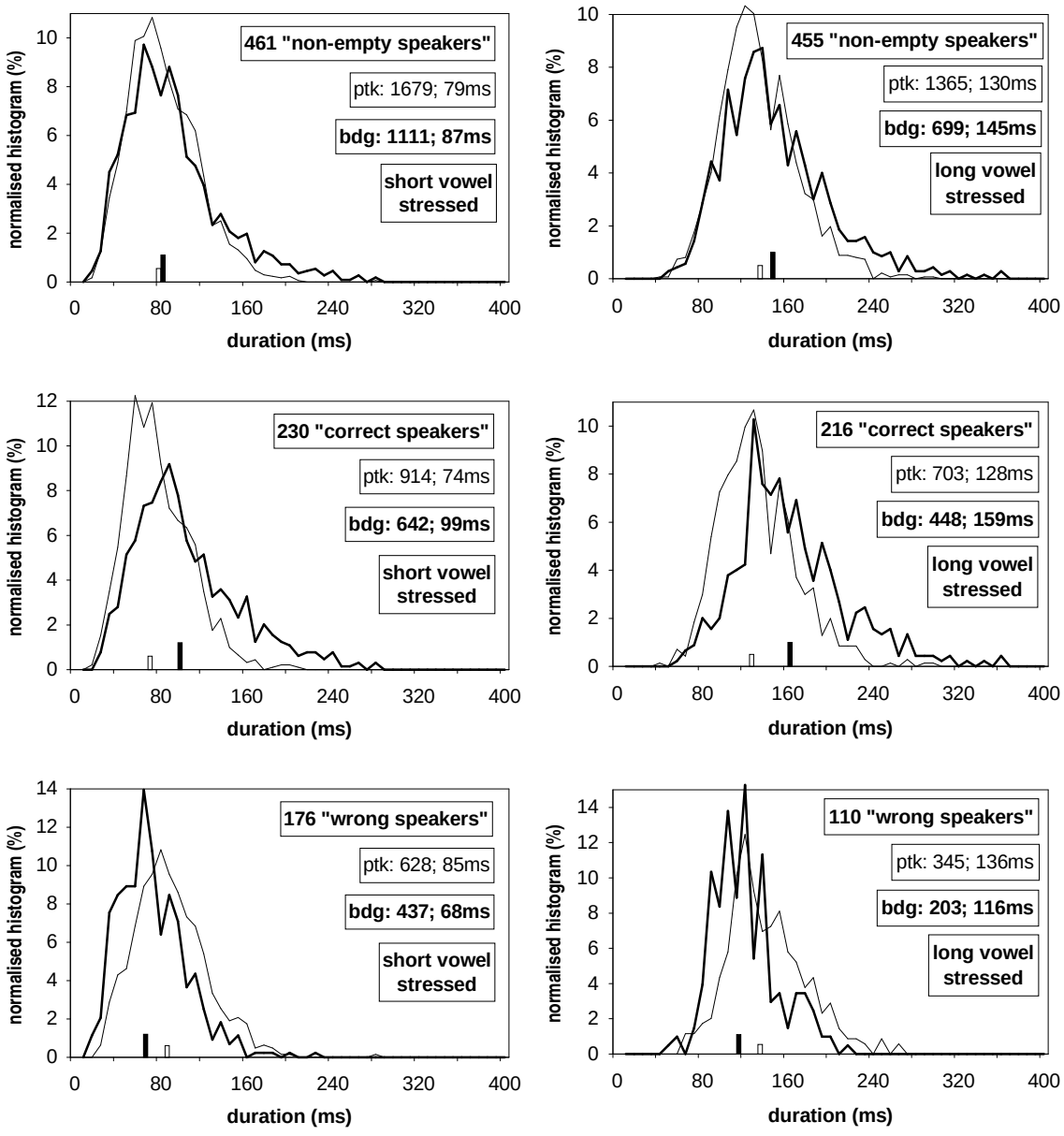


Figure 6.2 The upper 2 panels show the durational histograms of the effect of voicing of the post-vocalic stops on vowel duration, for short and long vowels separately. The lower 4 panels show the durational histograms of the "correct" (2 panels of middle row) and "wrong" (2 lower panels) speakers for short (left panels) and long (right panels) vowels. Only stressed vowel instances are used. Number of speakers, numbers of vowel instances, and mean durations in *ms* for /p,t,k/ and /b,d,g/ contexts, are shown for each panel.

whereas the differences in duration means (shown by the bars) are rather small. This is because many speakers simply show an opposite behaviour in their vowel durations before the stops, i.e., those followed by /p,t,k/ are longer (due to many other uncontrolled effects such as speaking style and rate). This group of speakers are temporarily called "wrong speakers", whereas the other group of speakers with a "conventional" durational behaviour, "correct speakers". The wrong and correct speakers compensate each other in the durational distributions of the two contexts (lower four panels of Figure 6.2).

For short vowels, 230 correct speakers spoke vowel instances in *both* /p,t,k/ and /b,d,g/ contexts, and 176 wrong speakers spoke vowels in both contexts. The total "non-empty" speakers is $461 > 230 + 176$, since here all speakers who spoke vowels in *either* of the two contexts are included. Given that each speaker only spoke 8 utterances, the number of vowel instances per speaker

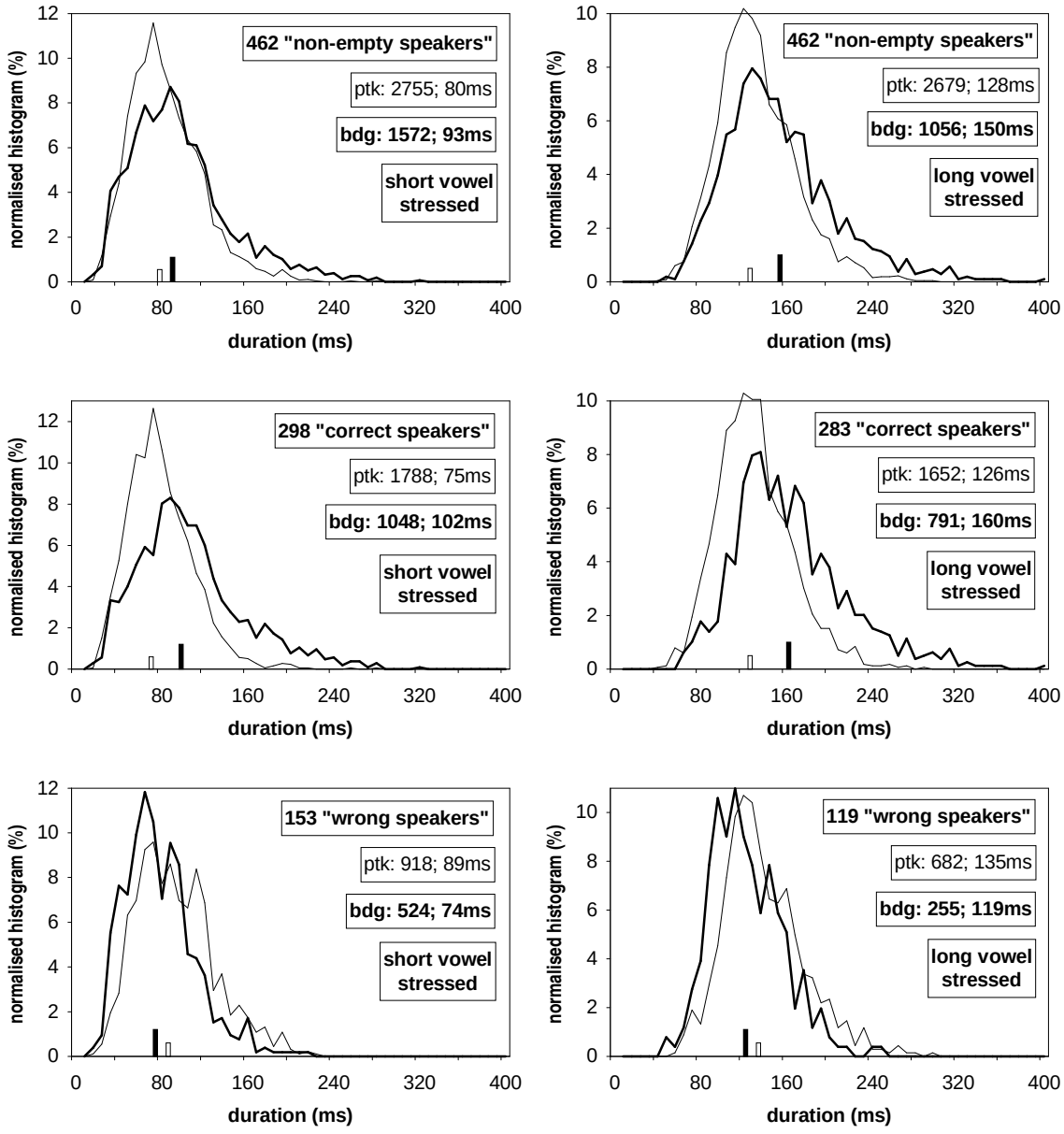


Figure 6.3 Same as Figure 6.2, but including vowel instances followed by unrealised stops.

per context varies between 0 and 10. The numbers of vowel instances do not sum to the total number either, of course.

In the above analysis, only those vowel instances that are followed by actually realised stop contexts, as given in the hand label files, were counted. There are a number of additional vowel instances which, according to the *norm* pronunciation, are also followed by the given contexts. They may receive the same influence from the following stops although these stop bursts are not realised. In order to verify this, we made a similar analysis as the one above but including all these vowel instances. These vowel instances and their right contexts are identified by DP between the norm and the actual labels. The results are shown in Figure 6.3. Although the numbers of both speakers and instances increased, and the difference in duration mean for long vowels increased slightly from the data excluding the unrealised contexts, as above, the problem of "wrong speakers" remains. The large difference in instance counts is an indication of the serious deletion of stop bursts in TIMIT (see footnote 3 on page 104 for a statistics).

6.2.4 Grouping utterances by speaking rate

The speaking rate at which the individual speech units (phones) are produced, is also a factor affecting the durational distribution. The use of this factor has been inspired by at least two observations. The first is that in a recent formal ARPA assessment of speech recognisers (Pallett et al., 1995), the speaking rate of speakers appeared to play a substantial role in affecting the recognition performance (the faster the rate, the worse the recognition). The second is that the speaking rate is an easily measurable parameter at recognition time, so that it can in principle be used to improve speech recognition behaviour (Jones & Woodland, 1993; Ohno & Fujisaki, 1995; Osaka et al., 1994; Siegler & Stern, 1994; Suaudeau & Andre-Obrecht, 1993).

According to the analysis and experience of Jones & Woodland (1993), the rate is best measured at the sentence level. The phone is chosen as the basic measuring unit. Furthermore, in order to emphasise the variation in rate while de-emphasising the variation caused by the intrinsic duration of the different phones in the phone inventory, a normalised phone duration τ, instead of the absolute duration d, is used:

$$\tau = \frac{d - \mu}{\sigma}, \tag{1}$$

where μ and σ are the mean and standard deviation of the duration of that phone in the training set. The speaking rate r of an utterance is defined as

$$r = \frac{1}{N}\sum_{i=1}^{N} \tau_i,$$

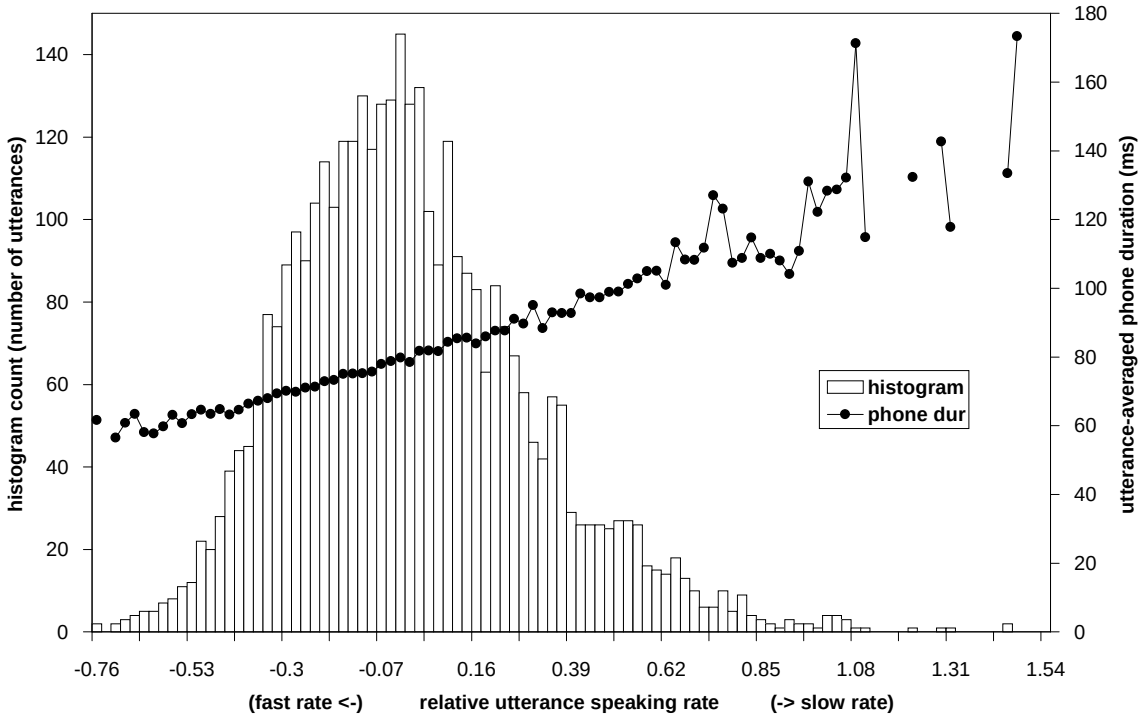


Figure 6.4 Histogram of the relative speaking rate (for a definition, see text) for all 3,696 utterances of the TIMIT training set. The curve with the marker • represents the utterance-averaged phone duration in *ms* (against right axis) in each histogram bin.

where N is the number of phone instances in that utterance[6].

We analysed the TIMIT training set (3,696 utterances) and obtained a histogram of utterances in normalised rate (Figure 6.4)[7]. The distribution is similar to the usual durational pdf of most phones, having a binomial-like distribution. For comparison, the utterance-averaged absolute phone duration in the corresponding histogram bins are also shown. It can be seen that the averaged absolute phone duration has a near-linear relation with the relative utterance speaking rate, particularly true in the middle region where counts are large. The effect of different intrinsic phone duration is not smoothed out through the two normalisations above for rate regions with small counts, thus showing irregularities.

Based on the rate measure of each utterance, the whole training set is divided into 3 groups corresponding to the fast, the average, and the slow utterance rate, each with an equal number of utterances (1,232). Here any speaker may have utterances belonging to different rate groups. After grouping based on the rate of *utterances* (see Figure 6.5 for two example phones), the *phone* durational histograms coincide with the utterance rate (e.g., phones found in a fast-utterance group are indeed shorter).

---

[6]Such a definition is for a convenient comparison with the phone duration. Literally, this is the reciprocal of 'rate'.

[7]In our earlier paper (Pols et al., 1996), this figure unfortunately contained an error since σ^2 was used rather than σ in formula (1).

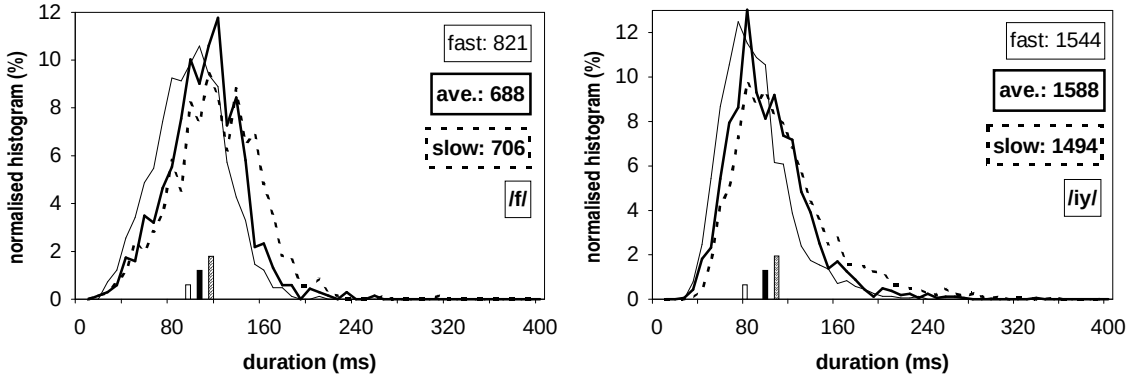


Figure 6.5 Histogram of the actual duration of two example phones /f/ and /iy/, separate for the three rate groups: fast (thin line), average (thick line) and slow (broken line), respectively. The number of tokens in each group is given, whereas the mean duration in each group is indicated by a bar.

6.2.5 Analysis of variance (ANOVA) of factors affecting duration

From the analyses above, effects on duration of some factors are shown in the *separate* histogram plots. In order to compare between the effects of different factors, we need to put all the factors into one framework of analysis.

A statistical technique "analysis of variance" (ANOVA) (e.g., Scheffe, 1959) should be suitable for indicating the relative significance of the factors. Actually, it is well known that the continuous speech signal contains a large amount of variations of a variety of types. Surprisingly, it is only until very recently (Sun & Deng, 1995) that we see such publications in which all the factors are analysed under a unified framework of ANOVA. Sun & Deng (1995) analysed the *spectral* variations (in MFCC) affected by factors such as phonetic class, speaker, and sub-segmental sequence of frames, using the TIMIT data. Their findings support experiences in the speech recognition practice, such as the improvement in performance by using the triphones.

The actual statistical model of the ANOVA used by Sun & Deng (1995) is fully nested between factors, partly because of the nested structure of the TIMIT data. We also chose to use such a nested model, for an additional reason that it is technically tractable. In our study, we analyse the variations in phone *duration* as explained in various factors. (The word "variation" is used instead of "variance", because in our calculation, the value is not normalised.) Vowels and other segments are treated differently for some factors, as explained below. The following 11 factors are considered:

R	Speaking rate	*Lu*	location of syllable in utterance
Cl	Broad phonetic class	*G*	gender of speaker
Ph	Phone	*Dr*	dialect region of speaker
Pt	Phone in context (only vowel)	*Sp*	speaker
S	Stress	*Sg*	phone segment
Lw	Location of syllable in word		

R has 3 levels[8]: {slow, average, fast} and is measured per utterance as explained in the previous sub-section. *Cl* has 8 levels: {stops, affricates, fricatives, nasals, semi-vowels and glides, long vowels, short vowels, pauses}. *Ph* represents the 50 reduced TIMIT phones (see Chapter 3). *Pt* only considers the right context of vowels as being either /p,t,k/, /b,d,g/[9], or the rest. For other phones *Pt* only has one level. Such a restriction for factor levels also applies to *S, Lw* and *Lu* because stress and location of syllables are defined by vowel segments. *S* has 3 levels, i.e., primary stress, secondary stress, or unstressed. *Lu* has 3 levels as being the last, the penultimate or the rest position, within the utterance. *Lw* has one level more than *Lu*, i.e., for vowels in monosyllabic words. *G, Dr* and *Sp* are obvious, and are provided by the file arrangement of the TIMIT database (all files of one speaker *Sp* are in one directory, all *Sp* in one dialect region *Dr* are in one directory at one level higher, and gender *G* is notified by the first letter of the *Sp* name, being either *m* or *f*). *Sg* refers to the actual instances of a given phone.

The relations between all the levels in all the $K = 11$ factors can be shown as a tree (Riley, 1992), part of which is given in Figure 6.6. To calculate the variation in each of the factors in terms of sum of squares (*SS*), a nested model for any two adjacent factors k and $k+1$ is used, where $k = 0, 1, \cdots K-1$. The total *SS* can be decomposed into a sum of *SS*s of all factors. Each SS_k is decomposed into $SS_{between}$ and SS_{within}, and the latter is further decomposed into $SS_{between}$ and SS_{within} at a lower factor. A derivation of the calculation formulae for the decomposition process is given in Appendix 6.3. For factor k,

$$SS_k = \sum_{i_{K-1}} \sum_{i_{K-2}} \cdots \sum_{i_{k+1}} \left[\sum_{i_k} \overline{Y}_k^2 C_k - \overline{Y}_{k+1}^2 C_{k+1} \right],$$

where each summation for i_k is over all the daughter nodes at factor k that belong to a same mother at factor $k+1$. Note that here the count C_k is the number of all the nodes at the lowest factor (indexed 0) that belong to node i_k, while $\overline{Y}_k$ is the mean over all such nodes.

A problem of computer memory is encountered in calculating the *SS* terms caused by the large number of nodes, that each require an accumulator and a counter to be allocated. Down to factor *Dr*, the total number of nodes are 67,680 (see Figure 6.6). Plain implementation down to the next factor *Sp* would require a multiplication of this value by a number between 8 and 59 (the number of speakers per *Dr* per *G*). In order to avoid this, an algorithm is

---

[8]The term 'level' used in the statistical literature denotes the number of discrete values that each factor can take. In a tree (see later) with different factors, each factor is referred to being at a 'depth' of a tree, to avoid confusion.

[9]Actually, these are the second-right phones since the immediate right phones following vowels are the closures, being either /cl/ or /vcl/, see Table 3.9 of Chapter 3. In the statistics of this section, the unrealised /ptk/ and /bdg/ contexts are also included. If these contexts were excluded, the variation in *Pt* would be slightly smaller.

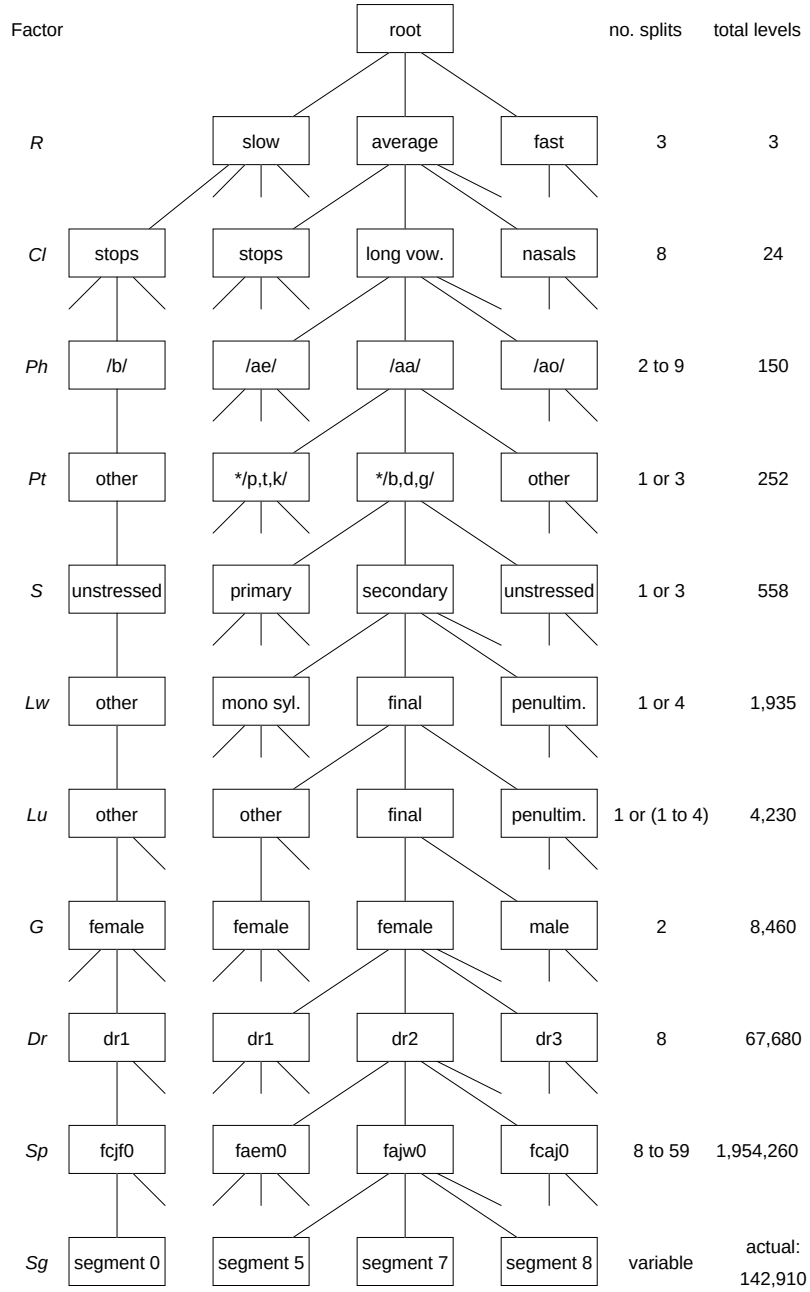


Figure 6.6 Part of the tree for ANOVA for the 11 indicated factors. The number of possible splits of levels per factor, as well as the cumulative numbers of levels, are indicated. The number of levels in the first 10 factors are theoretical, namely some of them may be empty. For the last factor *Sg* the number of actual non-empty cells is given.

designed which allocates temporal accumulators and counters for a single speaker. The results are put into global accumulators and counters after each speaker, and the temporal accumulators and counters are reset for use by the data of the next speaker.

The result of the calculation of the *SS* terms is given in the percentage of the variation explained by each of the 11 factors[10]. This is shown in Table 6.5.

[10]The *SS* in *Sg* is actually the remaining variation, which is different from the *SS*s of the other 10 factors. In the analysis no further factors below *Sg* are considered to play a role.

Table 6.5 Percentages of variation in terms of *SS* for the 11 factors calculated from the TIMIT training set. The upper section shows the results of a factor-ordering with the 'speaker factors' *G*, *Dr* and *Sp* following the 'phonetic factors' from *Cl* down to *Lu*. The lower section shows the results of swapping these two groups of factors. The last column shows the loss of calculated *SS* due to singleton cells (for a discussion see Appendix 6.3).

R	*Cl*	*Ph*	*Pt*	*S*	*Lw*	*Lu*	*G*	*Dr*	*Sp*	*Sg*	loss
2.00	15.59	25.10	0.29	0.38	0.84	0.94	0.27	2.29	15.65	35.83	0.82
R	*G*	*Dr*	*Sp*	*Cl*	*Ph*	*Pt*	*S*	*Lw*	*Lu*	*Sg*	loss
2.02	0.00	0.04	0.70	19.73	36.29	1.53	1.14	1.39	0.51	35.83	0.81

Table 6.6 From top to bottom the percentages of the variations per factor are presented, while increasingly omitting factors from the analysis. The top row contains all the factors, whereas the lowest row only contains the two factors *Sp* and *Sg*. The order of the factors for all analyses are the same (from left to right). Losses due to singleton cells are shown in the last column. For clarity, entries with the same value (within two decimals of precision) as the entry below are shown with a downward arrow ⇓.

R	*Cl*	*Ph*	*Pt*	*S*	*Lw*	*Lu*	*G*	*Dr*	*Sp*	*Sg*	loss
2.00	15.59	25.10	0.29	0.38	0.84	0.94	0.27	2.29	15.65	35.83	0.82
⇓	⇓	⇓	⇓	⇓	⇓	⇓	0.27		17.93	⇓	⇓
2.00	15.59	25.10	0.29	0.38	0.84	0.94			18.20	35.83	0.82
2.02	15.93	25.26	0.30	0.39	0.85				18.84	36.42	0.00
⇓	⇓	⇓	⇓	0.39					18.16	37.94	⇓
⇓	⇓	⇓	0.30						17.36	39.13	⇓
⇓	⇓	25.26							16.08	40.70	⇓
⇓	15.93								4.89	77.17	⇓
2.02									0.74	97.24	⇓
									1.64	98.36	0.00

In general, the contribution of each factor is order-dependent. The most tangible phenomenon seen from the percentages is, that when *Sp* is put before *Cl* and *Ph* (the lower section), variation in *Sp* is rather small, while when *Sp* is put after the splitting of the data by *Cl* and *Ph*, *Sp* explains a much larger percentage of variation. This is further confirmed by the data in Table 6.6, in which the order of the factors is kept while one factor at a time is omitted. When *Cl* and *Ph* are omitted, *Sp* explains a very small percentage.

In principle, when a factor is omitted from the analysis, other factors should account for more variation than otherwise. However, the above analysis about the interaction between *Sp* and *Ph* shows just the opposite situation: in the lower section of Table 6.5, along the ordering direction from left to right, when reaching *Sp* without first splitting by *Cl* and *Ph*, the variation in *Sp* is much smaller (0.70) than in the upper row (15.65).

Such a phenomenon that appears in databases with very many speakers such as TIMIT, may be explained as follows using some example durational histograms in Figure 6.7. The relatively wide distributions as shown at the bottom for each individual phone over all the speakers, indicate that, if the data are split by phones, there is a large durational variation across the speakers. When the data are not split by phones, the durational histogram for every individual speaker tends to be about the same (the right column), and

consequently the variation *across* speakers is very small[11]. (The right-bottom histogram for "All" is narrower than other bottom histograms).

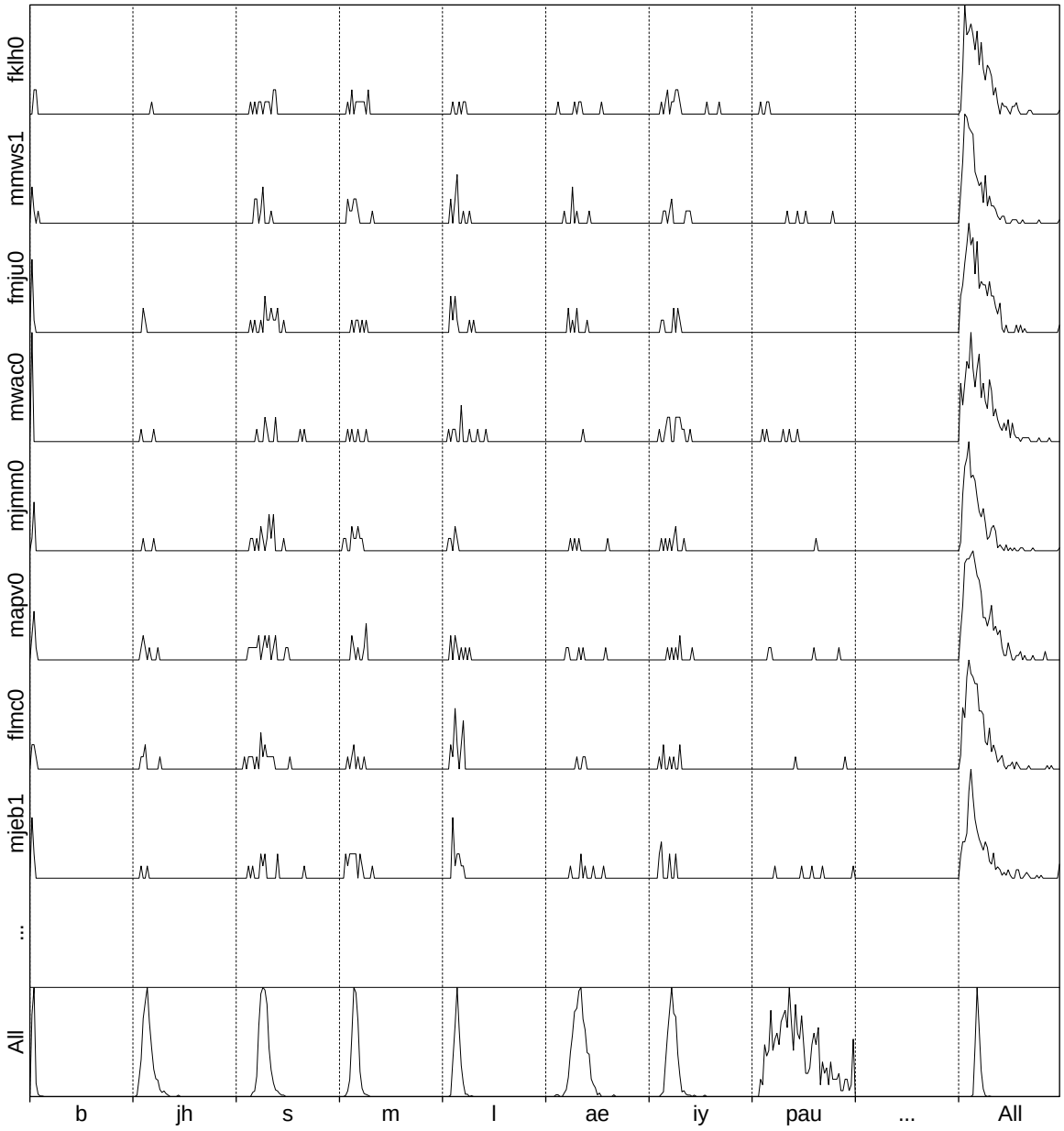


Figure 6.7 Durational histograms of 8 randomly selected speakers (one from each dialect region) and of 8 phones (one from each phonetic class) using TIMIT training set. The right-most column shows the durational histogram for each speaker over all 50 phones. The bottom row shows the histograms of the duration *means* across all 462 speakers, for particular phones, whereas the right-most histogram on the bottom row gives the histogram of the means of all the speakers, each averaged over 'all' of his phones. For maximum visibility, each histogram in the right column and in the bottom row is scaled to its own maximum height, whereas all other histograms are scaled to one unique maximum height across all of them. For an interpretation see text.

[11]One speaker might speak some phones faster but then other phones slower, than other speakers, when all the different speakers were forced to speak in a reading mode.

The above discussion indicates a general point to be noticed in such a nested analysis of variation. One should not just omit specific factors that actually play an essential role in the data, nor use an improper order of factors. When one omits *Ph*, one basically assumes that all the speakers are uttering non-speech sounds bearing no information, which is not the case. Furthermore, given our knowledge about speech, *Ph* should be before *Sp*, therefore the data reaching *Sp* down in the factor tree are already split by *Ph*. Although a completely proper order for all the factors is difficult to define, the above discussion points to some sensible considerations.

6.3 Conclusion

In this chapter, the phone duration as a function of various factors is analysed, using the TIMIT database. Both direct plots of the histograms, as well as ANOVA for the various factors involved, are used. Word stress (*S*) and syllable location (*Lu* and *Lw*) have a significant effect on duration as shown in the histogram plots using sub-sets of the speech data. Contradictory to general phonetic theory, the effect of the voicing of the post-vocalic stops (*Pt*) on the duration of the preceding vowel is not reflected in any consistent way across the many speakers. On the other hand, speaking rate (*R*) shows a consistent effect. Our attempt to improve recognition performance will mainly be based on the factors, which show significant effects on duration, in the analyses of individual factors.

The numerical results of the adapted ANOVA, as introduced in this chapter, will be considered as being preliminary for three reasons. First, in the factor design used, for practical reasons, all the factors involved were considered to be completely nested, which may be too much a simplification. Furthermore, interactions between factors are not analysed, because in the technical implementation of a duration model (Subsection 7.2.2), direct data partition on the factor cells will be used. Second, the total variations of the factors concerning the phone identities (*Cl, Ph* and *Pt*), and that of the phone segments (*Sg*) together comprise a rather large portion of the total variation (40.98% and 35.83%, respectively). This would have made the calculation of the variation in other factors inaccurate. Third, and the most tangible reason, the amount of data in TIMIT is still too limited for analyses at this scale. This is clearly seen in the serious mismatch between the number of cells needed for the factors deep down in the analysis tree and the number of non-empty cells at that depth (see Figure 6.6). Therefore the numerical results of the ANOVA will only be of referential value, and will *not* be used to guide the further study of duration modelling.

Both ways of analysis of context-dependent segmental duration make a necessary bridge between the general phonetic theories and the practice of automatic speech recognition (ASR). One of the reasons for the difficulties in

making the phonetic theories useful for ASR is that such theories have mainly been built and verified on rather unrealistic speech data, such as isolated short words or nonsense syllables. The attempt in this chapter started with a speech database TIMIT which contains continuously meaningful sentences read by many speakers, which matches the practical situation of speech recognition (in this thesis work). The fact that TIMIT is an existing, general-purpose database, makes the data in it more reliable (in the sense of being close to practice) than a database specially made to study some isolated factors. Then we searched through the database for consistent and useful *structures*, using every technique that is needed and available to us. We had some success in isolating and describing the interwoven factors. In practice, we have gained much insight, not only into the useful information in a practical speech database (which factors are important), but also into the special cares that one must take in such a *goal-oriented* data analysis (e.g., the ordering of the nested factors).

Appendix 6.1 Dynamic programming for symbol sequences

Dynamic programming (DP, see, e.g., Furui, 1989; Sakoe, 1992) can be used to match two sequences of symbols. Assume a test sequence $Q_t = t_1 t_2 \cdots t_T$ and a reference sequence $Q_r = r_1 r_2 \cdots r_R$, where each t or r is a symbol (in our case, a phone). In general, $T \neq R$, and at any time i, the corresponding symbols in the two sequences may or may not match. The situation of an exact match $t_i = r_j$ is referred to as a 'hit'. When a different symbol is aligned between Q_t and Q_r: $t_i \neq r_i$, it is called a 'substitution'. When a symbol exists in Q_r but not in Q_t, it is called a 'deletion', and the opposite situation is called an 'insertion'. When Q_r and Q_t do not match completely, there generally exist sub-sequences somewhere within Q_t and Q_r, that match each other. The task of DP is to find as many as possible matched sub-sequences in Q_t and Q_r.

The algorithm that we used is as follows (see Figure 6.8).

1. A 2-dimensional grid is made of cells $(i, j), i = 1, \cdots, T, j = 1, \cdots, R$, with the test symbols t_i along the horizontal axis and the reference symbol r_j along the vertical axis. Each cell (i, j) is associated with the following fields:
   - $c(i, j)$: the accumulated *cost* made from $(1, 1)$ up to (i, j);
   - dir: the direction of the best transition into (i, j), being vertical,

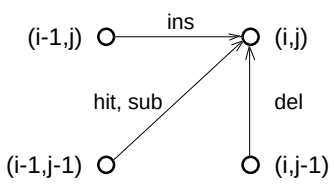


Figure 6.8 Possible transitions to cell (i, j) and the associated situations of (mis)matches.

horizontal or diagonal;

- counts for hit, ins (insertion), del (deletion), and sub (substitution).

2. For all i, then for all j, repeatedly fill in the fields of all the cells (i, j) according to the matching situation (being hit, sub, ins or del), and the transition made into (i, j):

$$c(i, j) = \min\{c(i-1, j) + p_{ins}, c(i-1, j-1) + p_{sub}, c(i, j-1) + p_{del}\},$$

where p's are the empirically determined penalties for different kinds of mismatches (typically, $p_{sub} = 4$, $p_{del} = 3$ and $p_{ins} = 3$), and the term p_{sub} is omitted for the 'hit' situation. To distinguish between 'hit' and 'sub', the symbols t_{i-1} and r_{j-1} are compared. Then the other fields of (i, j) are filled accordingly:

$c(i, j) = \min\{\cdot\}$	matching situation	dir =	Count incremented
$c(i-1, j-1)$	hit	diagonal	
$c(i-1, j-1) + p_{sub}$	sub	diagonal	Sub
$c(i-1, j) + p_{ins}$	ins	horizontal	Ins
$c(i, j-1) + p_{del}$	del	vertical	Del

3. In the end, the optimal path is found by tracing back from (T, R) along the 'dir' field of the cells, till $(1, 1)$. The count fields are used for global scoring of the mismatches.

It can be seen that the decision of an optimal transition is not made at each local step. Rather, all the possible single step transitions are 'registered' along with the costs. The actual decision for the optimal match between the two sequences is made during the back-tracing. Therefore, the path containing the smallest total cost, accounted for by fewest mismatches, will be chosen as the optimal path. A desired balance between the penalties for different kinds of mismatches can be controlled by adjusting the p values.

Appendix 6.2 Confusion matrix of norm vs. actual labels

The complete confusion matrix between the norm pronunciation according to the TIMIT lexicon (row entries) and the actual pronunciation (column entries) of all the phone instances found in the TIMIT training set, is shown on the following two pages. Here are some additional global statistics:

	N1		N2	N2/N1 (%)
total actual vowels	45,572	matched with any vowels	42,912	94.2
total actual consonants	97,338	matched with any consonants	92,397	94.9
total norm phones	154,133	total correct actual phones	120,599	78.2
total deletions	15,545	total deleted closures + stops	10,509	67.6
total substitutions	17,989			
total insertions	4,322			

	iy	ih	eh	ae	ix	ax	ah	uw	uh	aa	ao	ey	ay	oy	aw	ow	er	y	w	hh	r	l	el	m	n	en
iy	3920	129	1		257	6	2	6				2					1	15								2
ih	173	3068	41	1	1833	280	11	2	17	1		3	2			1	43				1	2	56		2	35
eh	4	104	2386	10	101	30	21		2		1	8			1	1	32						1		1	4
ae	1	77	589	2253	358	92	15		1	34		18	4		3		50			1		1	1	5		35
ix	148	284	35	1	1686	253	31	1	1								13								22	232
ax	200	258	65	3	1802	2400	344	1	16	19	2	120	1			8	57	5		1			59	1	1	81
ah		14	14		46	127	1562		27	7	7				1	1	1						9			1
uw	1	7	2		202	129	2	1907	24						7	2	9		8			1	6			
uh		13	1		98	25	1	12	345		18					2	169						27			
aa		1	1	4	4	23	79			1860	48		9		2	1	94			1						
ao		2			9	34	37	1	14	166	1760			10	1	18	172			2	2					1
ey	6	12	29	3	11	3	1					2079	4													
ay	1	16	2	5	9	5	8			72	1	2	1913	1							1		4			
oy	1										8			288		1			1		2					
aw			1	6			10			48					709	2	2	1	1							
ow					5	36	11	6	19	1	6			5		1607	2		24				3			
er		8			51	71	8	3	4	1	1						2912			1	86		2			
y	5	10			23	5		3	1		1						30	924		1						
w	1	1			2	3		2	3								1		2147		1		3			
hh		1	11		60	4				1										1603	1		1			
r	6	8	1		44	22	5			4		1					465	1	2		4504					2
l					11	4	1			3	2				2	2	1					4274	23			
el		1			2	8	2		4							1						115	744			
m		2									1								2					3529	4	
n	1	3	11		10	6	6			1							5			1		2	3	22	6652	25
en	1	2	1		6	3																		1	82	151
ng																								2	60	
ch		1			1	1															1	1			2	
jh		1			3																					
s					3												3									1
sh								1																		
z		1	1		10																				1	
zh																										
f					1												1									
th																										
v					1		1														2	1				
dh	27	6	3		53		4																		12	
q																										
p					3	3											3							1		
t	41	75	25		205	48	22		3	7	1	3				1	16		1	2	1	1	2		7	25
k	1	1	1		5	1	1		1			1				1	1	1								
b	1	5	2		14	3	1			1							7				2		3			1
d	23	53	16		107	20	26		4	6	1	6					13							1	2	6
g					8	2	1									1	3						1		1	
dx																										
cl	3	31	10	3	107	47	6	1	2	3	2	1					14				2	3	2		24	4
vcl	6	12	4	3	49	11	5		2	5		1			1		4	2		1	2	1			7	1
epi																										
pau																										
ns																										
lns	55	41	24		170	187	42	6	10	16	6	8	1		1	3	14	46	30	47	73	23	1	4	16	23

ng	ch	jh	s	sh	z	zh	f	th	v	dh	q	p	t	k	b	d	g	dx	cl	vcl	epi	Pau	ns	Del	%c	
											7											1	2		63	90.1	iy
2			4	2	1	1		9			19							4	4	2	6	8		513	54.5	ih
																								32	88.1	eh
3											21							5	1		5	4		126	63.0	ae
	1		1		1	2					3					1		20	10	2	24	1		345	60.8	ix
3			1	1	1		3		1		107		2					18	2	3	4	11		684	42.8	ax
																		2			1			85	85.8	ah
									1		29		1					2						170	81.5	uw
											2							3			3	1		61	47.9	uh
											13							1				3		17	86.8	aa
											19			11							1	2		33	77.8	ao
			1			1					3													9	96.6	ey
											9										1			11	93.3	ay
																									95.7	oy
											3													9	90.5	aw
											1							1				1		8	93.0	ow
											5													27	92.4	er
	2					2																		127	91.8	y
											1										1	1		19	99.1	w
			1								1		1					1			14	10		279	93.7	hh
1							1	1			31					1		28		1	7	3		735	87.6	r
								1			5		1					1				1		282	98.7	l
					1						16										1	4		12	82.8	el
3															1									62	99.6	m
58			3						1		22		8			2			10	2	11			541	96.9	n
			1								1										1	1		9	60.2	en
1142											10							1		1		1		33	93.8	ng
	759	1		4							1										1			25	98.2	ch
	10	973		1		1											1		1			1		56	98.1	jh
		1	5861	57	38			1					1						3		6	1		204	98.1	s
	11		1	1215															1		1			12	98.8	sh
			252	2	3623	32				1	9		1			1		1		1	9	11		218	91.6	z
	1	7		2	1	70														1				1	85.4	zh
				1			2193		3			1												13	99.7	f
			1					574		4	1		1			1			2					15	98.3	th
							12		1950	1	1									3	1			101	98.8	v
			1		1			147		2363	11		1			5		3			1	1		99	89.5	dh
																										q
				2				3			6	2586		5	3	2						8		345	98.5	p
1	29	3	3		1				1		90		3857			47		19			1	63		3382	83.8	t
	4		2	11				2			6		3	3771		7					1	4		709	98.6	k
			2					1							2176			1				1		268	98.0	b
3	1	26	20		4			1		2	30	1	13			2342		10	1			15		2299	85.1	d
							1				1			2			1177					4		269	97.9	g
																										dx
			2	17	2		5	3	2		413					18		1073	12278	64	22	65		1661	86.3	cl
2	2	1	3	1	5	38			34	3	30		1	1				657	20	7096	6	75		1576	87.7	vcl
																										epi
																										pau
																							7392			ns
2		1	16		3	1	1	3	1	2	1758		57	4	1	5	13	13	185	43	778	588				lns

Appendix 6.3 Decomposition of variations among factors

In classical textbooks on analysis of variance (e.g., Scheffe, 1959; Dunn & Clark, 1987), we only find calculation formulae for up to 4 nested factors, and only for equal number of factor levels at each split. For our special purpose, therefore, we have to develop our own formulae. We will also discuss problems of singletons and of empty cells.

The total variation in segment duration is given in terms of sum of squares SS. The total SS is

$$SS_{total} = \sum_{Y_0} (Y_0 - \overline{Y}_{grand})^2,$$

where Y_0 is the individual observation (in our case, the duration of a segment). The summation is over all the observations Y_0 in the database, and the grand mean is also taken over all Y_0. In order to decompose SS_{total} into SS_k explained in each factor $k = 0, 1, \cdots K-1$ (K being the total number of factors, see Figure 6.6 as an example with $K = 11$), we define for each factor k a mean value (Figure 6.9)

$$\overline{Y}_k = \frac{\sum_{i_{k-1}=1}^{I_{k-1}} \cdots \sum_{i_1=1}^{I_1} \sum_{i_0=1}^{I_0} Y_0}{\sum_{i_{k-1}=1}^{I_{k-1}} \cdots \sum_{i_1=1}^{I_1} I_0}, \tag{2}$$

i.e., the mean over all the observations at the lowest factor Y_0 that belong to the node i_k of factor k. I_k is the number of daughters at factor k from the same mother at one factor higher. Note that each I and Y is specified by the indices of the nodes in all the higher factors: $\overline{Y}_k = \overline{Y}_k(i_k, i_{k+1}, \cdots, i_{K-1})$, $I_k = I_k(i_{k+1}, i_{k+2}, \cdots, i_{K-1})$, while these indices are omitted in (2) for simplicity. We can also define the count C_k of Y_0 that belong to the node i_k of factor k,

$$C_k(i_k, i_{k+1}, \cdots, i_{K-1}) = \sum_{i_{k-1}=1}^{I_{k-1}} \cdots \sum_{i_1=1}^{I_1} I_0, \qquad C_0 = 1, \qquad C_1 = I_0. \tag{3}$$

Using (2) and (3) we can get recursive relations

$$\overline{Y}_{k+1} C_{k+1} = \sum_{i_k} \overline{Y}_k C_k, \qquad C_{k+1} = \sum_{i_k} C_k. \tag{4}$$

Using the above notations, we now begin to decompose the SS_{total}. We will prove that for any given total number of factors K in a system,

$$SS_{total} = \sum_{Y_0} (Y_0 - \overline{Y}_{grand})^2 = \sum_{Y_0} \left[(Y_0 - \overline{Y}_1)^2 + (\overline{Y}_1 - \overline{Y}_2)^2 + \cdots + (\overline{Y}_{K-1} - \overline{Y}_{grand})^2 \right], \tag{5}$$

with all the $\overline{Y}_k$, $k = 0, 1, \cdots K-1$ as defined above. Here $\overline{Y}_{grand} = \overline{Y}_K$. We will use induction for this. First, we check for the case with $K' = 2$ factors. The summation over Y_0 as in (5) can be done in a nested way, first over i_0 and then over i_1. We insert the term $\overline{Y}_1$,

$$\sum_{Y_0}(Y_0-\overline{Y}_{grand})^2=\sum_{i_1=1}^{I_1}\sum_{i_0=1}^{I_0}(Y_0-\overline{Y}_1+\overline{Y}_1-\overline{Y}_{grand})^2$$
$$=\sum_{i_1=1}^{I_1}\sum_{i_0=1}^{I_0}\Big[(Y_0-\overline{Y}_1)^2+(\overline{Y}_1-\overline{Y}_{grand})^2+2(Y_0-\overline{Y}_1)(\overline{Y}_1-\overline{Y}_{grand})\Big].$$

Using the indexing relations of Y and I given above, to take the irrelevant terms out of the summation, and using the relation (4) for $k=0$,

$$\sum_{i_0}Y_0=I_0\overline{Y}_1,$$

we find the cross term (taking the irrelevant terms out of the summation):

$$\sum_{i_0=1}^{I_0}2(Y_0-\overline{Y}_1)(\overline{Y}_1-\overline{Y}_{grand})=2(\overline{Y}_1-\overline{Y}_{grand})\left(\sum_{i_0=1}^{I_0}Y_0-I_0\overline{Y}_1\right)=0.$$

This proves the case for $K'=2$:

$$\sum_{i_1=1}^{I_1}\sum_{i_0=1}^{I_0}(Y_0-\overline{Y}_{grand})^2=\sum_{i_1=1}^{I_1}\sum_{i_0=1}^{I_0}\Big[(Y_0-\overline{Y}_1)^2+(\overline{Y}_1-\overline{Y}_{grand})^2\Big].$$

Then we assume that for $K'=K$ the decomposition relation (5) holds:

$$\sum_{Y_0}(Y_0-\overline{Y}_{grand})^2=\sum_{i_{K-1}}\cdots\sum_{i_1}\sum_{i_0}\Big[(Y_0-\overline{Y}_1)^2+(\overline{Y}_1-\overline{Y}_2)^2+\cdots+(\overline{Y}_{K-1}-\overline{Y}_{grand})^2\Big],$$

For $K'=K+1$ we add one factor above the factor $K-1$. The explicit summation over Y_0 now includes an additional summation over i_K, for all terms. All the terms for the case $K'=K$ still exist, while we insert the mean $\overline{Y}_K$ into the last term

$$\sum_{i_K}\sum_{i_{K-1}}\cdots\sum_{i_1}\sum_{i_0}(\overline{Y}_{K-1}-\overline{Y}_{grand})^2=\sum_{i_K}\sum_{i_{K-1}}\cdots\sum_{i_1}\sum_{i_0}(\overline{Y}_{K-1}-\overline{Y}_K+\overline{Y}_K-\overline{Y}_{grand})^2.$$

When we use the relation

$$\sum_{i_{K-1}}\overline{Y}_{K-1}C_{K-1}=\overline{Y}_KC_K,$$

the cross term is

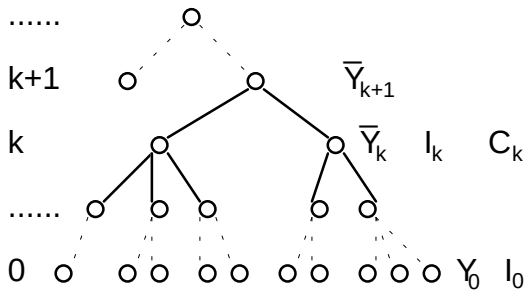


Figure 6.9 Illustration of part of the tree with identities Y, C and I for factor 0, k and k+1.

$$\sum_{i_K}\sum_{i_{K-1}} 2(\overline{Y}_{K-1}-\overline{Y}_K)(\overline{Y}_K-\overline{Y}_{grand})\sum_{i_{K-2}}\cdots\sum_{i_1}\sum_{i_0}1=\sum_{i_K}\sum_{i_{K-1}} 2(\overline{Y}_{K-1}-\overline{Y}_K)(\overline{Y}_K-\overline{Y}_{grand})C_{K-1}$$

$$=\sum_{i_K} 2(\overline{Y}_K-\overline{Y}_{grand})\left(\sum_{i_{K-1}}\overline{Y}_{K-1}C_{K-1}-\overline{Y}_K\sum_{i_{K-1}}C_{K-1}\right)=0.$$

This completes the proof that (5) holds for all $K'>2$. Furthermore, since

$$\sum_{i_{K-1}}\cdots\sum_{i_1}\sum_{i_0}(\overline{Y}_k-\overline{Y}_{k+1})^2=\sum_{i_{K-1}}\cdots\sum_{i_k}(\overline{Y}_k-\overline{Y}_{k+1})^2C_k,$$

where each summing term is the part of contribution of the variation in factor k given by the node i_k, summation of this term over all the structures from factor k upwards is just SS_k. Then we can write the decomposition relation as

$$SS_{total}=\sum_{k=0}^{K-1}SS_k=\sum_{k=0}^{K-1}\sum_{Y_0}(\overline{Y}_k-\overline{Y}_{k+1})^2=\sum_{k=0}^{K-1}\left[\sum_{i_{K-1}}\sum_{i_{K-2}}\cdots\sum_{i_{k+1}}\left(\sum_{i_k}\overline{Y}_k^{\,2}C_k-\overline{Y}_{k+1}^2C_{k+1}\right)\right],$$

where we used (3) and the recursive relations (4) to extend for each SS_k

$$\sum_{i_k}\cdots\sum_{i_0}(\overline{Y}_k-\overline{Y}_{k+1})^2=\sum_{i_k}(\overline{Y}_k-\overline{Y}_{k+1})^2C_k$$

$$=\sum_{i_k}\overline{Y}_k^{\,2}C_k-2\overline{Y}_{k+1}\sum_{i_k}\overline{Y}_kC_k+\overline{Y}_{k+1}^2\sum_{i_k}C_k=\sum_{i_k}\overline{Y}_k^{\,2}C_k-\overline{Y}_{k+1}^2C_{k+1}.$$

Another practical problem caused by the limited number of observations and the very large number of cells in the lower factors, is the loss of precision due to empty cells and singleton cells. When there is an empty cell at factor k, i.e., $C_k=0$, this sub-tree has no contribution to factor k and above. This part of observation is also absent in SS_{total}, therefore it does not cause a loss in the calculated variations.

For a singleton cell that contains just one observation Y_0' and $C_k'=1$, such a Y_0' does have contribution to SS_{total}, but may not have contribution to other SS_k, therefore causes a problem of calculation precision (SS_k sum to a value less than SS_{total}). To be exact, we search for places in the tree where $(\overline{Y}_k-\overline{Y}_{k+1})$ becomes zero due to a singleton. For all the factors $k>0$ except the lowest,

$$\overline{Y}_k-\overline{Y}_{k+1}=Y_0'-\frac{\sum_{i_k\neq i_k'}\overline{Y}_kC_k+Y_0'}{\sum_{i_k\neq i_k'}C_k+1},$$

which is in general non-zero. However, for SS_0 of the lowest factor, since there is only one $C_0=1$ and the rest are zero,

$$Y_0-\overline{Y}_1=Y_0'-\frac{\sum_{i_0}Y_0'C_0}{\sum_{i_0}C_0}=0.$$

This shows that the singletons cause loss of calculated variation only for the lowest factor in the factor tree.

7

INCORPORATING CONTEXTUAL DURATION KNOWLEDGE IN POST-PROCESSING OF *HMM* RECOGNITION*

Abstract

This chapter addresses the techniques of incorporating knowledge about segmental duration, as influenced by various contextual factors, by means of post-processing in an HMM-based recogniser. Our approach is: first of all, to use context-free HMMs for obtaining sentence hypotheses (the first phase) at the word level, and then to generate phone-level transcriptions (the second phase). The phone-level transcription is generated from the word-level transcription based on the norm pronunciation plus an optional word-juncture model (also derived from speech data). Afterwards, the duration scores of the identified phones with their contexts are obtained by means of a duration model. This model is built parametrically based on data durational statistics on a chosen set of contextual factors. The sentence hypotheses are re-scored with the acoustic score from the first phase and the duration score from the second phase. The new top word transcription is taken as the recognition output. Various implementation issues of these algorithms are discussed, and analyses on the moderately improved recognition performance are presented.

*This chapter is substantially extended from Wang, Ten Bosch & Pols (1996b).

7.1 Introduction

In Chapters 4 and 5, the statistical durational behaviour of the linear HMM without explicit state duration models has been analysed and has been fitted to the durational statistics of the phone segments. It was proven that such a modelling improves the durational behaviour of the HMM and that it moderately improves the phone segmentation performance. Applying such modelling gives insight into the relationship between the segmental duration and HMM behaviour via the HMM parameters. However, the durational statistics was measured for each phone segment irrespective of its contexts, and the HMM behaviour is only considered in a context-independent way. Since the complex nature of the phonetic segmental duration cannot be sufficiently covered by the context-independent durational statistics alone (Chapter 6), the modification of the context-independent HMMs alone is not sufficient in improving the recognition performance. Much important durational information in speech is indeed context-dependent. In this chapter the knowledge about context-dependent duration will be incorporated, using a technique different from the one in Chapter 4 and 5.

Although the Viterbi algorithm always gives the most likely frame sequence (with the highest score s_a, including acoustic and language models), we know, it does not guarantee the sequence with the most correct words. An "N-best" algorithm has been developed (Schwartz & Austin, 1991) which provides N sequences of the hypothesised words with the highest s_a, instead of only the "top best" as in Viterbi. Usually, additional knowledge sources should be used to find the hypothesis with most correct words among the top N best hypotheses. In our study, the knowledge source is about context-dependent duration, and it is added by means of duration scores s_D. The process of combining s_a and s_D to get the new top best is called re-scoring. The whole process is called post-processing (Figure 7.1).

Some special points in Figure 7.1 stem from the N-best program available to us, which only generates N-best hypotheses in forms of word transcriptions, but not transcriptions at phone level. On the other hand, our durational knowledge is given at phone level. To add the phone duration score, one has to generate phone transcriptions, which provide the identity and contexts of each hypothesised phone. Our solution to this problem is to use a two-phase procedure. The first phase is the N-best algorithm (using context-independent monophone HMMs) which outputs word transcriptions. In the second phase, phone transcriptions are generated based on the word transcriptions. The context-dependent durational knowledge will be built into a duration model external of the HMM.

In Section 7.2, a contextual duration model will be built, based on a set of 4 durational factors chosen out of the 11 factors in Chapter 6. Then in Section 7.3, the quality of the particular N-best algorithm, and a word-juncture model

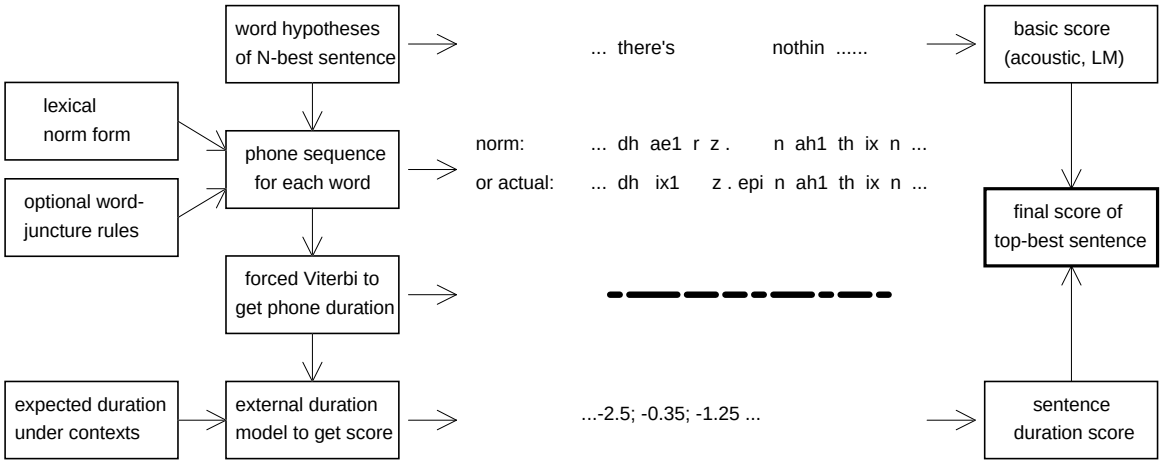


Figure 7.1 Word recognition with *N*-best post-processing. The middle part of the illustration is part of an example sentence "...there's nothin...", undergoing several steps in the post-processing. From top to bottom: hypothesised word sequence; norm or actual form of the corresponding phone sequences; hypothesised phone duration; and the duration scores (in logarithm) of the phones.

that converts from the norm to the actual phone transcriptions of the *N*-best hypotheses, will be discussed. In Section 7.4, first the duration scores of the utterance transcriptions are generated from the phone duration scores. Then some analysis of the score distributions will be given, for the purpose of a better understanding of the re-scoring process. In Section 7.5, several alternative re-scoring methods will be attempted and the recognition results will be compared. The last section will give a general discussion.

7.2 Contextual duration model

The contextual duration model should be a structured statistical model which describes the relations between the observed segmental duration and the various factors that influence the duration. In practice, only the most influential factors and their interactions, rather than all factors, need to be considered. The effect of the less important factors, even if included in the model, will be overridden by other technical mechanisms in the recogniser, when the duration model is built as part of it. On the other hand, the model should also be transparent to a phonetician, so that he can check whether the structure and the parameters of the model are phonetically correct. Given both the technical and "phonetic" notions above, our approach will be based on the simplest possible model, rather than a comprehensive one.

7.2.1 Factors included in the model

Based on the analysis of the contextual factors in the previous chapter, we make the following selection of factors out of the 11 factors as analysed in

Chapter 6. First of all, the factors 'phonetic class' and 'phone' will not be considered, since the phones themselves are already used as the basic units in recognition. Therefore the variation of duration due to these factors is already explained by using the phones (Within the data separated for a phone, there is no variation across different phones). The factor 'phone in context' will also be excluded since even the most tangible effect assumed by the phonetic theory (the vowel-lengthening-before-voicing effect) is unseen in our multi-speaker data. Furthermore, all speaker-related factors have to be excluded, since these are not identifiable at the time of recognition, using our particular (speaker-independent) recogniser. This leaves us the 4 factors shown in Table 7.1. As was done in the previous chapter, the levels of each factor are indicated in the table with their indices.

Each of the 33 non-vowel phones has only one level in the three factors *S*, *Lw* and *Lu*, meaning that these three factors are irrelevant for non-vowel phones. Each of the 17 vowel phones has a full range of levels, except for some dependencies between *S* and *Lw*, and between *Lw* and *Lu*. The total number of "cells" in the lowest factor *Lu* for all the 50 phones is thus

$$33 \times 3 + 17 \times 3 \times [2 \times (1+3+2+3) + (1+3+2)] = 1{,}323.$$

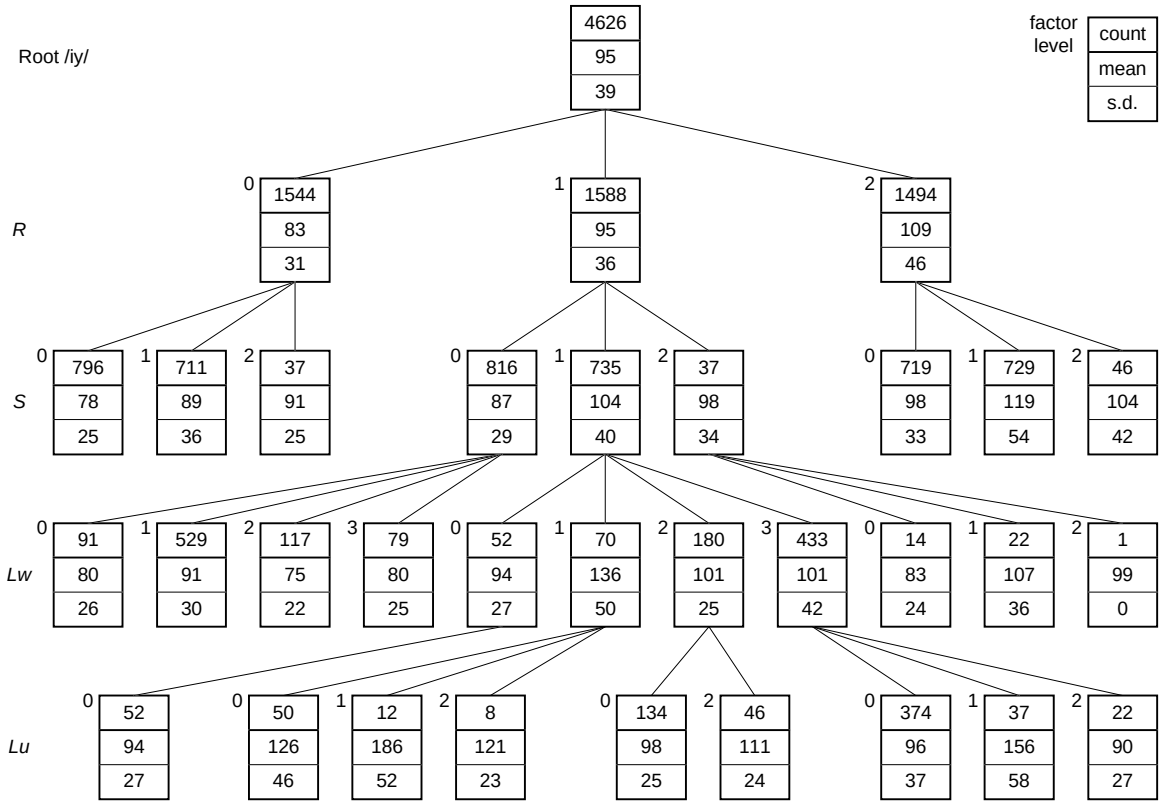
Parts of the factor trees for two example phones are shown in Figure 7.2. Some effects of splitting the data can be seen. For example, for phone /iy/, the duration mean for the slow rate (*R*=2) is larger (109 *ms*) than the fast rate (*R*=0) which is 83 *ms*. For the average rate (*R*=1), duration means for both primary stress (*S*=1) and secondary stress (*S*=2) are longer than unstressed (104, 98 and 87 *ms*, respectively). Furthermore, one can see that some cells in the lowest factor have very small, or even zero, counts. This problem of sparse data will be discussed in the next sub-section.

Table 7.1 The four contextual factors and their levels for vowel phones. Some dependencies between levels of different factors are shown with "if" clauses. For example, if *S*=2 (secondary stress), *Lw* cannot be 3 (monosyllabic word).

	factor	levels
R	speaking rate	0: fast; 1: average; 2: slow
S	stress (of vowels)	0: unstressed; 1: primary; 2: secondary
Lw	location of syllable in word	if *S*=0,1: 0: others; 1:final; 2:penultimate; 3:mono. if *S*=2: 0: others; 1:final; 2:penultimate
Lu	location of syllable in utterance	if *Lw*=0: 0: other positions if *Lw*=1: 0: other positions; 1:final; 2:penultimate if *Lw*=2: 0: other positions; 2:penultimate if *Lw*=3: 0: other positions; 1:final; 2:penultimate

7.2.2 Form of the duration model: parametrical

Based on the experience of Jones and Woodland (1993), the best-performing duration model has a form of either a smoothed histogram or a Gaussian parametrical model. The Gaussian model was slightly worse than histograms,


Root /iy/
4626
95
39
factor level
count
mean
s.d.
R
0 1544 83 31
1 1588 95 36
2 1494 109 46
S
0 796 78 25
1 711 89 36
2 37 91 25
0 816 87 29
1 735 104 40
2 37 98 34
0 719 98 33
1 729 119 54
2 46 104 42
Lw
0 91 80 26
1 529 91 30
2 117 75 22
3 79 80 25
0 52 94 27
1 70 136 50
2 180 101 25
3 433 101 42
0 14 83 24
1 22 107 36
2 1 99 0
Lu
0 52 94 27
0 50 126 46
1 12 186 52
2 8 121 23
0 134 98 25
2 46 111 24
0 374 96 37
1 37 156 58
2 22 90 27

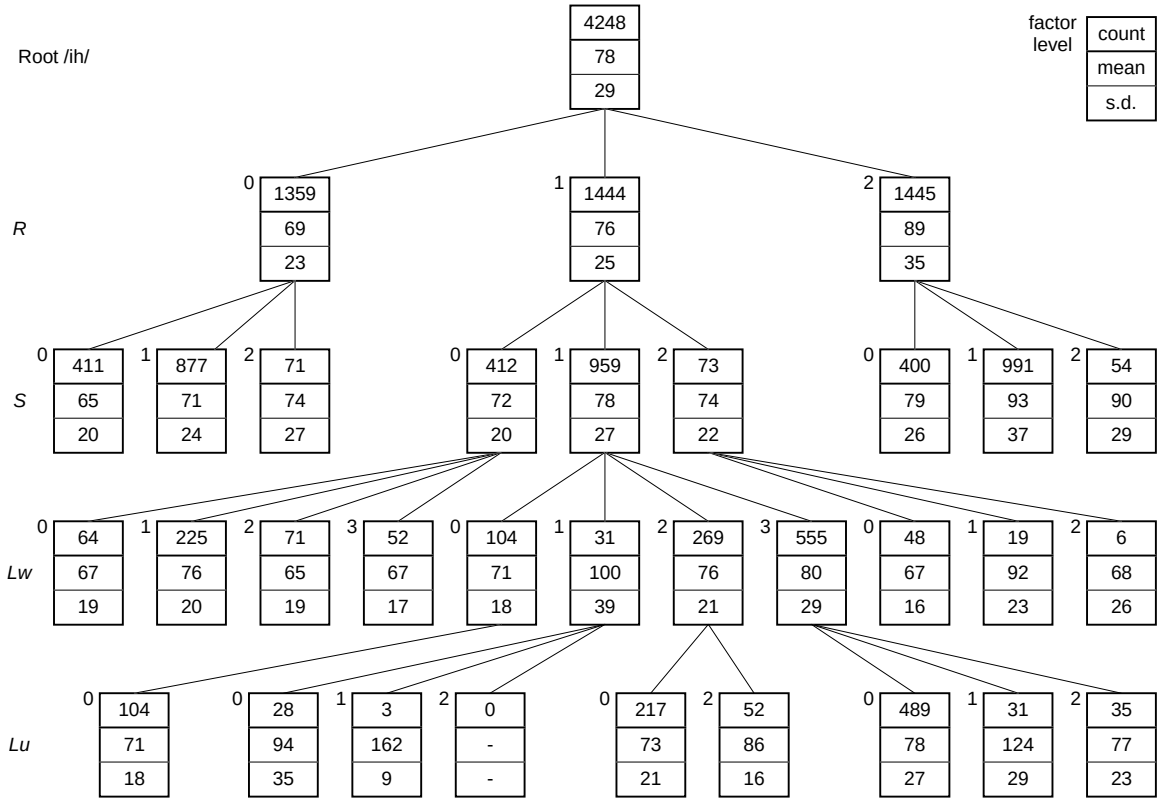

Root /ih/
4248
78
29
factor level
count
mean
s.d.
R
0 1359 69 23
1 1444 76 25
2 1445 89 35
S
0 411 65 20
1 877 71 24
2 71 74 27
0 412 72 20
1 959 78 27
2 73 74 22
0 400 79 26
1 991 93 37
2 54 90 29
Lw
0 64 67 19
1 225 76 20
2 71 65 19
3 52 67 17
0 104 71 18
1 31 100 39
2 269 76 21
3 555 80 29
0 48 67 16
1 19 92 23
2 6 68 26
Lu
0 104 71 18
0 28 94 35
1 3 162 9
2 0 - -
0 217 73 21
2 52 86 16
0 489 78 27
1 31 124 29
2 35 77 23

Figure 7.2 Parts of factor trees for phones /iy/ (upper part) and /ih/ (lower part). The counts are number of phone instances in the TIMIT training set. Mean and standard deviation (s.d.) are given in milliseconds (rounded to integer).

whereas these two forms of models are significantly better than the even simpler models with only maximum/minimum durations. So we will only consider the histogram and the parametrical models in our study. Furthermore, each model form can have several *methods of reacting* to the contextual factors. Jones and Woodland (1993) used speaking rate as the only factor, whereas there were three reacting methods. The best method is called a "partitioned" one which divides the training data set according to the levels of the factor under concern, while a duration model is built for each partition. The other two methods do not divide the training data set, but rather use parameters to alter the duration model for the whole training set. Although

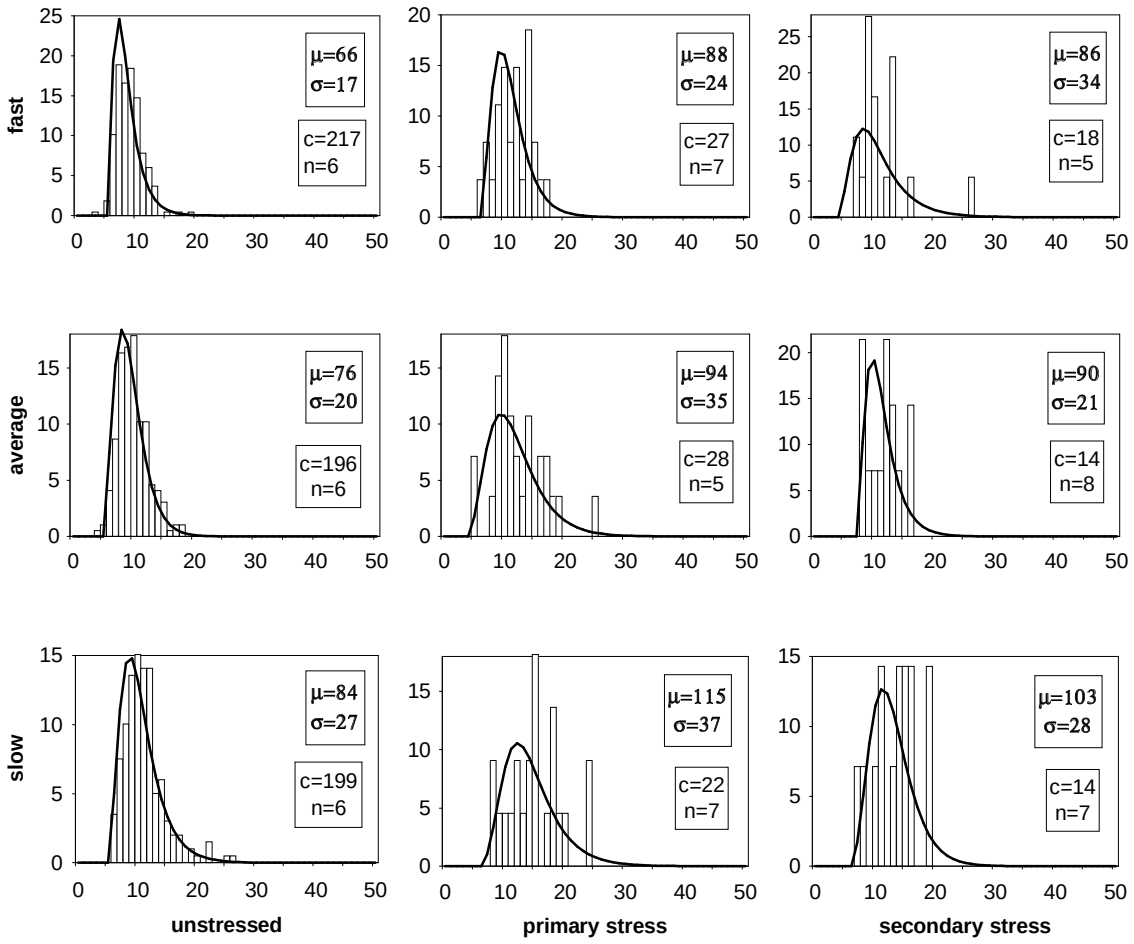


Figure 7.3 Normalised histograms (bars) and pdf's of the parametrical models (thick curves) of the phone /ih/ in word-final and non-utterance-final positions. Different situations for fast, average and slow speaking rates (from top to bottom) and for unstressed, primary and secondary stress (from left to right) are shown, respectively. The vertical axes of all the panels are in percentage, while the horizontal axes are in 8 milliseconds time steps. In each panel, mean (μ), standard deviation (σ) (both in millisecond), count (c) and the length (n) of the Markov model needed for fitting are also given.

the latter two methods are advantageous in having access to a larger training data set, they do not perform as well as the partitioned method. In our case, since we include a total of four factors, the problem of data size would be even more severe. However the partitioned method is attractively simple, whereas the other two methods would have much difficulty in coping with the interactions between the factors. We choose to use the partitioned method, while the problem of insufficient training data will be dealt with by parametrical modelling, as discussed below.

Due to the fact that the speech data for some phones will be rather limited under our specifications of levels in the four factors (a total of 1,323 cells in the lowest factor), we prefer to use a parametrical model over a histogram model. In our opinion, however, the worse score of Jones and Woodland (1993) by using the specific Gaussian pdf (probability density function) is because this pdf is not suitable for phone duration modelling. Using the estimated mean and variance to define a symmetrical Gaussian pdf would not fit well with the actual durational distribution, which generally has an asymmetrical shape (see Figure 7.3). Based on our study in Chapters 4 and 5, a weighted sum of negative-binomials provides a better form of pdf, which can be obtained by a Markov model. A smoothed histogram, on the other hand, will introduce artefacts due to the specific method for function smoothing, which may not be based on the actual problem to model.

A simple way to obtain the binomial-based pdf for a specific phone in context is to use the measured duration mean and variance to constrain the pdf. Unlike the Gaussian pdf, merely mean and variance do not determine a pdf uniquely. In general, different combinations of the mean and variance will leave a different amount of freedom in choosing the set of pdf's. In our study, we decided to use the following technique. We assumed an underlying linear Markov model that gives rise to the desired pdf. The number of selfloops was taken to be the minimal value allowed by the mean and variance. Then the whole region of mean and variance that can be fit by the Markov model was obtained. All but one selfloop probabilities were set equal and the last one served as a variable to satisfy both mean and variance (Appendix 7.1). When all selfloop probabilities were obtained, the binomial-based pdf could be calculated by, e.g., multiplications of the transition matrix of the Markov model (Chapter 4). Results of fitting durational histograms with the binomial-based pdf's for the phone /ih/ are shown in Figure 7.3. For simplicity, the figure only shows 9 example cells for 3 levels of R and 3 of S, for word-final ($Lw = 1$) and the "other" utterance positions ($Lu = 0$).

First, the duration mean and variance for all the 1,323 cells in the factor space were collected from the TIMIT training set. The great majority of these cells are non-empty. Then, for each of these cells (with down to one instance), a binomial-based pdf was calculated. For the few empty cells, all the pdf values were left zero. In later use of these parametrical pdf's, all the

evaluated (logarithmic) probabilities that fall below a minimal value p_0 will be replaced by p_0.

7.3 Generating word and phone transcriptions

The phone transcriptions will be generated in two phases. In phase one, word-level transcriptions will be generated by the *N*-best algorithm, and in phase two, the phone-level transcriptions will be generated based on the word transcriptions and a word-juncture model accounting for pronunciation variations such as between-word co-articulations.

7.3.1 The N-best algorithm and its quality

	we	always	thought	we	would	die	with	our	boots	on
0	will	it		the		riverbank		or	**boots**	**on**
1	will	it	all	inward	or	another			put	some
2	will	it	now	**we**	mean	by	the		bedroom	some
3	knows			the		riverbank		or	**boots**	**on**
4	heels	are		**we**	mean	by	the		bedroom	some
5	will	it	now	**we**	mean	by	the	third	of	some
6	will	it	all		in	another			put	some
7	will	it		the		riverbank		or	to	some
8	will	it	now	**we**	of	good	and		redwoods	**on**
9	will	it	only	the		giant			redwoods	**on**
10	heels	are		**we**	mean	by	the	third	of	some
11	we'll	let		the		riverbank		or	**boots**	**on**
12	will	it	can	**we**	mean	by	the		bedroom	some
13	heels	are		**we**	of	good	and		redwoods	**on**
14	will	it	the	worm	a	**die**	of a	third	of	some
15	will	it	the	worm	a	**die**	of	them	to	some
16	will	it	now	**we**	mean	by	the	far	to	some
17	will	it	all	inward	or		the		bedroom	some
18	will	it	the	worm	a	**die**	of	the	bedroom	some
19	will	it	now	**we**	mean	by	a	third	of	some
20	well	with	the			riverbank		or	**boots**	**on**
21	will	it	all	inward	or	another			put	**on**
22	we'll	work	the			riverbank		or	**boots**	**on**
23	will	it	only			by	the		bedroom	some
24	heels	are		**we**	mean	by	the	far	to	some
25	below	is	the			riverbank		or	**boots**	**on**
26	we'll	let	all	inward	or	another			put	some
27	**we**		now	is	the	riverbank		or	**boots**	**on**
28	will	it	the	worm	a	**die**	of	other	day	some
29	will	it	the	worm	a	**die**	of		evidence	**on**

Figure 7.4 The first row is the "should-be" utterance transcription (test/dr3/fpkt0/si908), whereas the rest are the top 30 hypotheses. Correct words are highlighted in bold face and are aligned together for easy reading only (they are not really time-aligned). Transcription at position 27 (shaded) has the most correct words (three). This example of a very badly recognised utterance is chosen for illustration purpose.

Table 7.2 Word percentage scores for the 1,344 utterances in the TIMIT test set using various numbers of N top hypotheses. "# hypotheses at N" are the actual number of hypotheses obtained due to a particular width of the beam-search, e.g., only 903 out of the total 1,344 utterances got actually N=30 hypotheses. Both "correct" and "accuracy" scores are calculated using the single sentence hypothesis which gives the highest word score among the N tops ("accuracy" scores include also the insertion errors). "Matched" scores are obtained by counting the correctly matched words in all the N tops.

N	# hypotheses at N	% correct	% accuracy	% matched
1	1344	79.07	76.97	79.07
5	1077	84.71	83.19	85.41
10	1012	86.01	84.59	87.06
15	976	86.60	85.32	87.84
20	950	87.07	85.99	88.46
25	923	87.37	86.10	88.85
30	903	87.68	86.39	89.20

There are several types of N-best algorithms (Schwartz & Austin, 1991), each trying to find back the good sentence candidates lost in the Viterbi algorithm, by registering alternatives at various transition levels. In this study we used a "word-dependent N-best", which registers alternatives at each between-state transition, and is dependent on the identity of the previous word[1]. Its typical performance in terms of provided sentence hypotheses can be seen in Figure 7.4 for an example utterance. It can be seen that some correct words not present in the top hypothesis can be found in other hypotheses.

To give an indication of the quality of such N-best algorithm, we ran the N-best recognition[2] with the whole TIMIT test set of 1,344 sentences, using a word-pair grammar derived from both training and test sets. The results are given in Table 7.2. This shows a consistent benefit by using more N top hypotheses[3]. The word correct score increases from 79.07% for $N = 1$ to 87.68% when using $N = 30$. This is the total space within which improvement may be gained by post-processing, since those words not present in any of the N-best hypotheses can never be found back[4]. (If the correct words in all the N

[1]There are two other N-best algorithms: The "word-lattice" algorithm only registers alternatives at the end of each word. The "sentence-dependent" algorithm is dependent on the word history of the whole sentence, and it finds completely accurate solutions. Both word-dependent and sentence-dependent algorithms are more accurate and computationally more expensive than the word-lattice one. The word-dependent algorithm is a compromise between the other two in both accuracy and computation cost.

[2]For a faster speed and the required computer memory space, we used the super computer SP2 of SARA of the University of Amsterdam.

[3]We used a different version of HTK from the one used in Chapters 3 and 5. Since the internal definition of language model is different in this version, the top-best score is different from (lower than) the scores in Chapters 3 and 5. Language match factors are not really well optimised.

[4]Although is it impossible to find back the lost correct words in a recognition process, it may be possible to find back some correct "concepts" in a speech understanding process, which is beyond the scope of this thesis.

hypotheses could somehow be combined, the score increases to 89.20%[5], see Appendix 7.2 for an attempt to benefit from this).

7.3.2 From word- to phone-transcriptions with timing

Our *N*-best hypotheses give us the transcription and timing at the word level only. In this subsection we describe a method to generate phone-level timing transcriptions from word-level ones.

7.3.2.1 Optional word juncture modelling

Phone transcriptions can be generated from the word sequence by using the norm pronunciations in the TIMIT lexicon. However, as has already been noted in Chapter 6, the actual pronunciation (i.e., the actual phone sequence for each word) frequently deviates rather severely from the norm one. Furthermore, it is noticed (see, e.g., Giachin et al., 1991) that the most severe transcription deviations occur at the word junctures. For example, the word sequence "what time" is actually pronounced as /w aa cl t ay m/ rather than /w aa cl t.cl t ay m/ ("cl" is a closure phone and "." indicates the word boundary), as concatenated from the norm phone sequences of the two words. In this study, we decided to model only the phenomenon of pronunciation deviation at word junctures, although generally in within-words locations deviations may also occur, such as vowel reduction (Van Bergem, 1995). The conversion mechanism from a norm pronunciation to an actual phone sequence can be called the *word-juncture model*. Such models can be extracted from the hand labels in the TIMIT training set. The procedures of generating and using the juncture model are depicted in Figure 7.5. The detailed procedures will be discussed in the following paragraphs.

We first define the "juncture area" as the sub-sequence of phones at the word junctures that may undergo some deviation from the norm form. We have to take into account that both vowels and consonants can occur at word boundaries (true for English). The juncture area is defined on the norm form[6]. Starting from the boundary between a pair of words and stepping into the two words, the treatment is the same for both directions. If the first phone is a vowel, only this vowel is included in the area and the stepping stops. If the first phone is not a vowel, the stepping continuous to include all the non-vowels before a vowel is encountered. Once such a juncture area is defined on the norm form, dynamic programming[7] (DP, see Chapter 6) is used to match

---

[5]This score is obtained by aligning the correct words from all the hypotheses with the word sequence of the correct sentence, using a DP algorithm. Any correct word present in any hypothesis is counted as a "match".

[6]The norm form is the standard, based on which we compare the actual forms. The actual form has a great diversity of variations in forms and is thus difficult to define.

[7]DP is needed because in TIMIT phone label files, word boundaries are unknown.

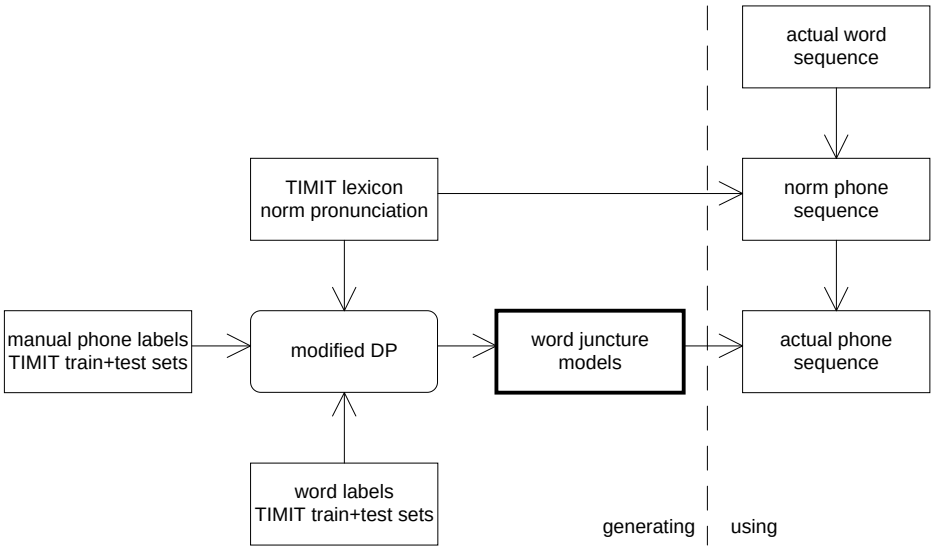


Figure 7.5 Procedures of generating and using word juncture models.

the phone sequences of the norm and the actual labels of a whole sentence. The matched phone sequence in the juncture area of the actual labels is obtained as the actual "juncture cluster". In order to diminish the unwanted situation that a vowel is matched with non-vowels, we developed an improved DP which uses strong penalties for mismatches between phones of different classes[8] (among them are vowels and consonants). If the same norm juncture cluster is realised in the database by several different actual juncture clusters, the one that occurs most frequently (the winner) is used.

We introduce three types of word-juncture models, differing in forms:

- type-1: explicit rules;
- type-2: list of actual juncture clusters given based on word-pairs;
- type-3: list of actual juncture clusters given based on norm juncture clusters.

In all three types of models, we can define the "input" and "output" fields of the model. For the type-1 model, the input is compact categories of norm juncture clusters (Giachin et al., 1991). For the type-2 model, the input is specified by the pair of words directly. For both types of models, the output field contains the actual phone sequence of the juncture area. The typical total number of type-1 models is 10, and an example of them is "consonant.same-consonant ⇒ consonant", where "⇒" separates the input field from the output field. For the type-2 models, a total of 15,617 different word-pair based juncture clusters are found in the TIMIT training plus test sets[9], among which 8,815 are different from the norm. It is not necessary to

[8]We used $p_{ins} = p_{del} = 3$, $p_{sub} = 4$, and $p_{cross\_class} = 10$.
[9]For a large coverage of the word-juncture models.

store the norm juncture clusters, since they can be generated from the lexicon. An example fragment of the type-2 models is as follows:

```
liked to ⇒ cl t 1 1
object to ⇒ cl t 1 1
respect to ⇒ cl t 1 1
subject to ⇒ cl t 6 7
invoked technology ⇒ cl t pau t 1 1
```

Two integer numbers are given at the end of each model. The first number indicates the number of the realisations for the output, whereas the second number is the number of all different realisations for the given input.

When we group the non-norm juncture clusters according to their norm juncture clusters, instead of the word pairs as above, we get the type-3 models. The input fields are presented by these norm clusters. We end up with 1,654 such clusters (total 8,401 clusters if norm ones are also included). The 5 lines in the fragment of the previous example (all having input cluster /cl k cl t.cl t/) are then summarised into one type-3 model as follows:

```
cl k cl t.cl t ⇒ cl t 9 11
```

The output /cl t/ with the highest total number of realised instances (9) in the search is taken (the winner)[10], whereas the alternative /cl t pau t/ with a small count (1) is discarded. It can be seen that the type-3 cluster-based models are not only more compact than the type-2 word-pair-based ones, but also have a larger number of instances among which to choose the winner, thus being statistically more reliable.

The type-3 model can be seen as a compromise between type-1 and type-2. We will use this compromised model in our study. In generating a "best-guess" actual phone sequence (the right part of Figure 7.5), first the norm phone sequences are generated for all actual words (e.g., /w aa cl t cl t ay m/ for "what time"), then juncture areas are defined between all these word pairs (for this example: /cl t.cl t/), and each norm juncture cluster will then define the input field. For each juncture cluster we search through the list of juncture models to find the one with a matched input field (/cl t.cl t/), and the output field (/cl t/) of that model is then copied into the juncture area[11] to get one complete "actual" phone sequence (/w aa cl t ay m/). When no model is

[10]To sum up for the number of winner instances (the first number in the lines), only those lines with the winner outputs are used. However the calculation of the second numbers (11) takes into account all the instances found. Line 5 in the previous example has output /cl t pau/ which is different from the winner, therefore the first "1" (not a winner) is not used while the second "1" is used, in this example, to get the numbers 9 and 11, respectively.

[11]In most cases the word is long enough to allow the treatment on both borders to be done separately. There is just one exception when the word is too short according to our definition of juncture area, namely the word "a" with only one vowel phone. In this case, the only vowel belongs to the junctures of both sides. Our decision for this word is to do juncture modelling *only* with its right-side context word.

found for the input juncture, the norm form is retained. When this is done for all the word pairs in an utterance, the best-guess phone sequence for the utterance is generated.

One way to compare the quality of the cluster-based models (type-3) with a set of rules (type-1) is to count the percentage of the instances that our clusters coincide with each of the rules (We take the 11 rules used by Giachin et al., 1991), using the data set. This comparison is given in Table 7.3. Note that the cluster areas defined for the input fields of the rules are generally smaller than those of our cluster-based models[12]. The rule inputs take 1 or 2 phones from the end of the first word and 1 phone from the beginning of the second word. Our input clusters are not defined in terms of fixed number of phones, but according to the actual situation, and thus can include up to 3 consonants on each side, plus possible closure phones[13] before each stop. Therefore in comparing, we actually merged all those different cluster input fields together which have the same phones at the word boundaries as defined in the rule inputs. For example, the cluster input fields

```
p t.k r
p t.p
p t.t r
k t.t
```

will all be compared with rule (2) /stop stop.stop/ in Table 7.3, while the last /r/ is irrelevant.

Furthermore, some of the rules in Giachin et al. (1991) have overlapping inputs if longer clusters are considered in the juncture area. For example the phone in rule 8 before phone /t/ may be /f/ or /n/, thus this input overlaps with rule 6. For the purpose of counting a fair coverage of the rules by our list of clusters, we removed all the overlaps from the rule inputs (for the above example, input "[f|n]t.d" is excluded for counting with rule 8). Rule 11 cannot be compared since its juncture area is outside our defined area. Let us make one more remark related to these comparisons. Most rules give rise to phone deletion in their output. After such a deletion, the remaining phones may belong to either word of the pair. For the output of the rules such a placement makes a difference. However, in generating the actual phone sequence, the word-boundary positions are irrelevant since we only need the sequence for the whole utterance (see also the two treatments in Sub-subsection 7.3.2.2).

The statistics shown in Table 7.3 indicate a good match (95.56%) between our cluster-based models and the set of rules. Furthermore, in Giachin et al. (1991), only the rules 1, 2 and 11 were used as "compulsory" rules, whereas all others were "optional". Optional rules are not necessarily always applied.

---

[12]Therefore the input fields for rules are more general and representative, whereas the input for cluster-based models are exact descriptions of single events.

[13]We deleted these closure phones before comparing the juncture clusters with the rules since the stops in these rules are given in the form without the closures.

Table 7.3 A set of word-juncture rules according to Giachin et al. (1991), and statistics of the matching of our cluster-based juncture models with these rules. "C" and "V" refer to consonants and vowels, respectively. "Stop" here refers to the set of phones /p,t,k,b,d,g/. "sil" is matched against /q,pau,cl,vcl/ (see Table 3.9 in Chapter 3 for a list of phone symbols). "[f | n]" means /f/ or /n/. "Total win." is the total number of winners in all different input fields of the cluster-based models that match the input field of one rule. "Correct" is the number of winners that are also correctly matched with the output of the rule. The percentages are "correct/total winner". "Total" is the total number of instances corresponding to the same cluster input, but may have different cluster output from the winner.

	rule input	rule output	correct	total win.	% cor.	total
rule 1	C.sameC	⇒ C	316	347	91.1	414
rule 2	stop stop.stop	⇒ stop	1141	1145	99.65	1357
rule 3	t.y	⇒ ch	7	38	18	87
rule 4	d.y	⇒ jh	27	30	90	59
rule 5	V t.V	⇒ V dx V	197	221	89.1	635
rule 6	[f | n] stop.stop	⇒ [f | n] stop	206	217	94.9	319
rule 7	[s | z].sh	⇒ sh	103	103	100.0	137
rule 8	t.[d | dh]	⇒ sil [d | dh]	162	167	97.0	232
rule 9	V t.dh	⇒ V dh	148	148	100.0	165
rule 10	n d.dh	⇒ n dh	41	41	100	87
rule 11	dh ax.V	⇒ dh ih V				
rule 1-10			2348	2457	95.56	3492
all clusters				9047		16346

Therefore the few low percentages do not have to indicate a mismatch, but are simply an indication of such possible exceptions in realisation. Furthermore, in our "lookup-table"-like models, a larger number of useful regularities than those covered by the 11 rules, may be included. The actual number of such regularities cannot be seen from the statistics, though, because we do not have the rules, of which the input fields can be used to check how many winners out of the total of 9,047 are "correct". However, the fact that 9,047 is much larger than 2,457 is a strong indication of the coverage of a large number of regularities.

It must be noticed that the word juncture modelling should be treated as an optional step in our post-processing process, the reasons for this will be discussed in the next sub-subsection.

7.3.2.2 Two treatments to get phone duration

In our two-phase procedure, the first phase of generating word-level *N*-best transcriptions is separated from the second phase that generates phone-level transcriptions (using a forced-Viterbi procedure, which finds phone boundaries with given phone sequence). The two-phase procedure may be less accurate (as compared with a single step procedure that directly generates phone-level transcriptions). There could be several alternative ways to implement the second phase, that each may lead to enhanced accuracy. The main aspects to be considered for the second phase are:

Table 7.4 Two treatments for the second-phase of post-processing procedure.

	treatment 1	treatment 2
Aspect 1: forced Viterbi at level of	utterance	word
Aspect 2: word-juncture modelling	with	without
Aspect 3: front-end optimised for	phone recognition	word recognition

Aspect 1. the forced-Viterbi performed at utterance or word level;
Aspect 2. with or without word-juncture modelling;
Aspect 3. which kind of optimality for front-end processing.

Since these three aspects are somehow related, we will not perform tests on all combinations of these aspects (8 conditions in total). Instead, we only test on two combined conditions, called "treatments" hereafter (Table 7.4).

First of all, it must be mentioned that, in the first-phase *N*-best recognition, each word has a single norm pronunciation based on the lexicon, plus an optional pause at its end. This must be kept in mind when considering the three aspects for the second phase, in the two treatments. With treatment 1, we are mainly concerned with optimality at individual steps of the post-processing. For aspect 1, we choose to perform the forced-Viterbi on the whole utterance; therefore the word-boundaries from phase one are ignored. Since all the between-word transitions are retained in the whole utterance, word-juncture models (aspect 2) can be applied, in order to be closer to the actual pronunciation in each test utterance. Additionally, closure phones /cl/ and /vcl/ are added in front of the stops not located at the beginning of a *sentence* (Figure 7.6). The phone sequences obtained this way may actually be better than what we can get in a single-step procedure when only the norm pronunciation is available. The front-end acoustic processing (aspect 3) is optimised for phone-recognition, since the recognition process in the forced-Viterbi is closer to the situation of performing a phone recognition than a word recognition (the language model only knows simple phone connections, not the complicated word connections). Taking the experience from Chapter 3, this includes a state-specific transformation of the acoustic vectors, and a set of language-matching factors (LMF) optimal for phone recognition.

On the other hand, with treatment 2, we attempt to keep the situation as close as possible to a single-step *N*-best program that directly produces the phone transcription. In such a situation, the phone transcriptions are produced in the same process as the word transcriptions, which is a word recognition process. Therefore, the front-end condition (aspect 3) is chosen as the optimal one for word recognition, i.e., the one without acoustic vector transformations (Chapter 3), and the accompanying optimal LMF. In order to be consistent with the norm word pronunciation that one would use in a single-step *N*-best program, word-juncture modelling will not be applied (aspect 2). For the same reason, the forced Viterbi is also performed at the

word level, i.e., within the phone sequence of each word (aspect 1). This way a better precision in phone duration will be obtained (Figure 7.6). Closure phones /cl/ and /vcl/ are added in front of the stops not located at the beginning of a *word*. For word-initial stops, optional closure phones are added. In the recognition process, the probabilities of phone sequences with and without the initial closures are compared and the best sequence is taken. At the end of each word, an optional pause (/pau/) is included.

In general, the phone sequences, the contexts and the durations of the phone segments in the two treatments may all be different, and both different from the single-step recogniser. For example, even with treatment 2, the duration of a specific phone instance may still be different from the phone duration generated from the single-step *N*-best recogniser (due to differences, e.g., between language models in the second phase, and in the single-step recogniser). Figure 7.6 shows that the hypothesised phone durations from the two treatments can be very different. For example, the segment boundary times for the phone sub-sequence /vcl g ih v pau/ in treatment 2 for the word "give" is aligned with the word boundaries, while /vcl g ih v/ in treatment 1 is not, since the former does the forced Viterbi at the word level, while the latter does it at the utterance level. The word "a" in treatment 1 is realised as /ix/ (due to word juncture modelling) while in treatment 2 it has the norm form /ax/. The pauses /pau/ are added by the forced Viterbi in treatment 2 to the ends of all words except for "a". It is hard to predict which of the two treatments will be better, in terms of producing phone transcriptions. Therefore, we will perform tests based on both treatments.

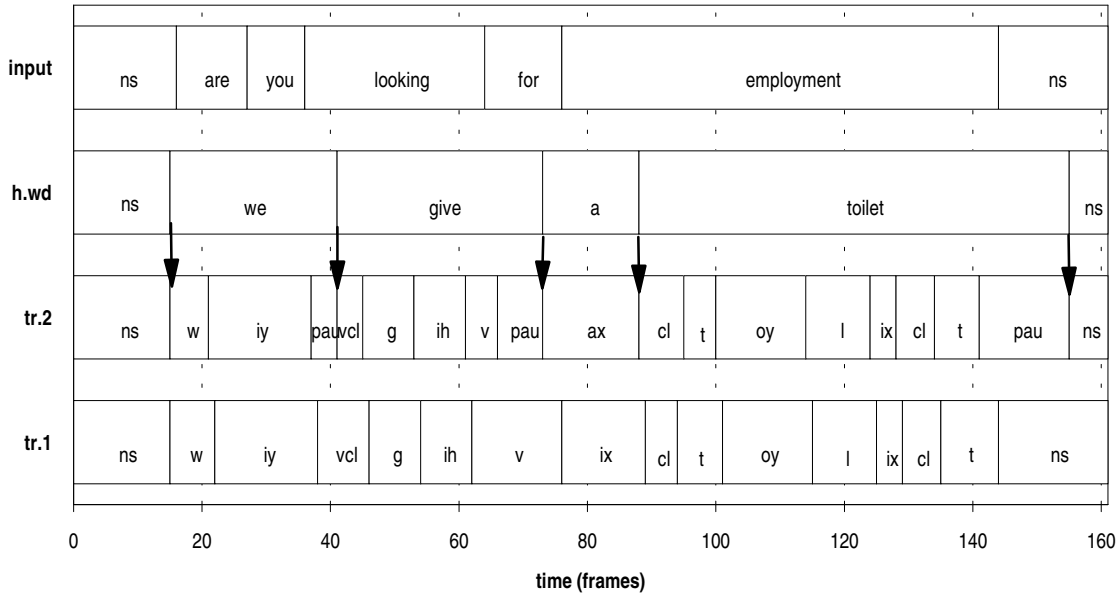


Figure 7.6 Timing of the input utterance "are you looking for employment", hypothesised as "we give a toilet". The four rows from top to bottom show the timing of the input word, the hypothesised words (h.wd), the treatment-2 phones, and treatment-1 phones (tr.2 and tr.1),

respectively. The row for "tr.2" is put closest to the word sequence for easy check of the alignment of word borders (shown with the dark arrows). "ns" at beginning and the end of each row indicates a segment of non-speech silence.

7.4 Duration scores

In this section, we will discuss various ways of obtaining the utterance duration scores from phone duration scores, and we will analyse distributions of duration scores.

7.4.1 From phone duration score to utterance duration score

Given the hypothesised duration d_i and the contexts of a phone instance i, the duration score (in logarithm) is $s_d(d_i) = \log s_d^0(d_i)$, where $s_d^0(d_i)$ is the context-dependent (CD) duration model[14] built in Subsection 7.2.2. Three types of utterance scores will be tested, all normalised by the total number of phones I in the utterance with a total duration D. The first is a "direct score"

$$s_D^D = \frac{1}{I}\sum_{i=1}^{I} s_d(d_i).$$

The second type of utterance duration score is called "local score" because it corrects the duration shift locally for each phone duration:

$$s_D^L = \frac{1}{I}\sum_{i=1}^{I} s_d(d_i - d_{i0} + \mu_i).$$

The third type of utterance duration score is called a "global score":

$$s_D^G = \frac{1}{I}\sum_{i=1}^{I} (w_G \Delta_i^A + \Delta_i^B),$$

because it corrects the shift by two measures Δ_i^A and Δ_i^B, which are related to the global shift of utterance. w_G is a weighting factor. The two deviation measures Δ_i^A and Δ_i^B will be described below.

In the re-scoring process to be discussed in the next subsection, we wish to assign a transcription with a low duration score, which contains the same I as the correct transcription, but with wrong phones and their contexts, and/or with each phone having a large duration deviation from its mean. To achieve this, in addition to the normalisation by I, we use an assumed phone duration d_{i0}, so that each phone has *an equal contribution* to the global shift of the utterance duration. Here the shift is defined as $(d-\mu)/\sigma$, where μ is mean

[14]We made three sets of such models, based on data of the TIMIT training set, the test set, and both sets, respectively. Although the figures in Subsection 7.2.2. are based on the training set, the models used from here on are based on the test set.

and σ is standard deviation. Therefore the relation between the shifts at the phone- and the utterance-levels should be:

$$I\frac{d_{i0}-\mu_i}{\sigma_i}=\frac{D-\mu_D}{\sigma_D}, \tag{1}$$

where the utterance-level statistics μ_D and σ_D are obtained by summation[15]

$$\mu_D=\sum_{i=1}^{I}\mu_i;\sigma_D=\sqrt{\sum_{i=1}^{I}\sigma_i^2},$$

where μ_i and σ_i are the CD durational statistics of the phone instance i (i indexes within the utterance, not the phone inventory). From (1), then,

$$d_{i0}=\mu_i+\frac{D-\mu_D}{I\sigma_D}\sigma_i.$$

The situation for the actual duration d_i of each phone instance is further complicated by the asymmetrical shape of the binomial phone durational pdf, such that μ_i in general does not coincide with the peak position $d_i^{\max}$, and by the fact that d_i may lie on either side of μ_i, of d_{i0}, and of $d_i^{\max}$ (see Figure 7.7). The complete deviations of phone scores including all different situations are summarised in two deviation measures:

$$\Delta_i^A=\begin{cases}-\left|s_d(d_{i0})-s_d(\mu_i)\right|, & \text{if } d_{i0}\geq d_i^{\max};\\ s_d(\mu_i)+s_d(d_{i0})-2s_d(d_i^{\max}), & \text{otherwise;}\end{cases}$$

$$\Delta_i^B=\begin{cases}-\left|s_d(d_i)-s_d(d_{i0})\right|, & \text{if } (d_i-d_i^{\max})(d_{i0}-d_i^{\max})>0;\\ s_d(d_{i0})+s_d(d_i)-2s_d(d_i^{\max}), & \text{otherwise,}\end{cases}$$

[15]Probability theory indicates that the mean and variance of the sum-variable is the sum of those of the individual random variables, if they are independent of each other.

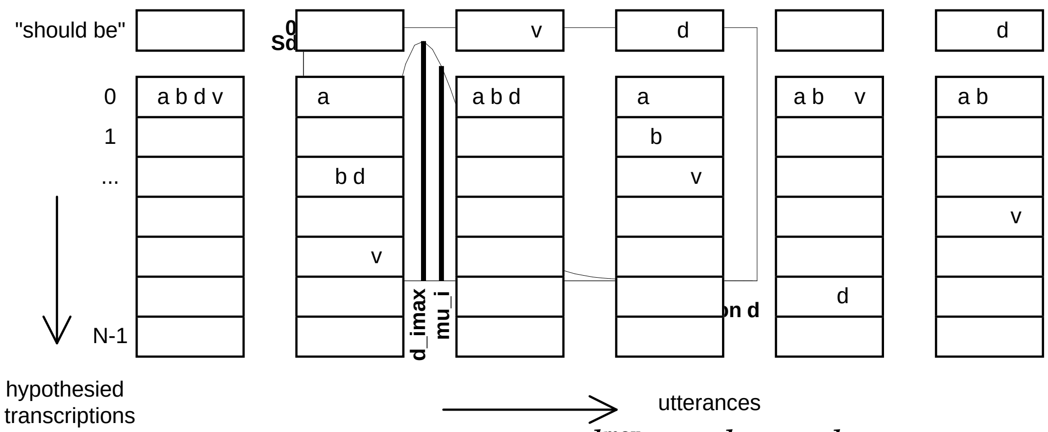


Figure 7.7 Illustration of the four typical values d_i^{max}, μ_i, d_{i0} and d_i of a durational pdf $F_d^0(d)$ used for calculating the normalised phone duration score

Figure 7.8 Various situations for 6 example utterances showing different situations. Each column of blocks shows some positions of N-best hypotheses, whereas the blocks are ordered by the acoustic scores (a) in the first-phase transcription generation. The separate blocks above the columns denote the "should be" transcriptions for the utterances. (b) indicates the transcriptions with the most correct words (real best). (d) and (v) indicate the transcriptions with highest duration score and score in the forced Viterbi back tracking, respectively.

where Δ_i^A indicates the score deviation of d_{i0} from μ_i, and Δ_i^B characterises the score deviation of d_i from d_{i0}.

7.4.2 Duration score distribution

Before the final re-scoring process which combines the first-phase acoustic score s_a and the duration score s_D (one of the three in previous section), we give an analysis of the distribution of the duration scores, to see under which conditions it is possible to improve the word recognition by re-scoring. After the first-phase N-best recognition, N transcriptions $T_n, n = 0, 1, \cdots, N-1$ for each utterance are output (T_0 is the "top", i.e., with highest s_a). If s_D for T_n, $(n > 0)$ is higher than s_D of T_0, it is *possible* that T_n gets to the top position after the re-scoring process. Therefore, we first checked to which extent this is true, given our particular treatments.

The complete situation is illustrated in Figure 7.8. For each utterance, among the N transcriptions, we use a total of 4 different marks to indicate the transcription which ranks highest with one of the 4 kinds of scores, respectively. For each transcription, the "acoustic" score[16] s_a (with mark "a") is obtained from the first phase N-best recognition. The duration score (with mark "d") is one of the three types of s_D obtained with one of the two treatments in the second phase. There is another acoustic score s_V, obtained in the second phase of the forced Viterbi back-tracking (an automatic segmentation process), which has a mark "v". The last mark "b" indicates the transcription which contains the largest number of correct words for the

[16]The name "acoustic" here should not be taken too literally since language model is also involved in obtaining this score.

utterance (the actual best). Of course a transcription may have more than one mark. We denote such coincident situations by, e.g., "a=b"[17].

The situation of the duration score distribution is not really ideal; e.g., the transcription with the highest s_D is in general not the best one (d≠b). However, we give the following statistics of the score distributions, for instance for a comparison between the two treatments. For the purpose of analysis, we also collect the statistics for the "should-be" transcription (we mark it "w") of each utterance. For this transcription we can generate s_D and s_V, but s_a is of course unavailable (in the first phase, when the utterance signal is input to the recogniser, only the N transcriptions are generated with their s_a). Since this "w" transcription is always "correct", in the figure the best "b" only means the best among the N actual transcriptions. The coincidence "d=a" also excludes "w". The statistics are given in the following two tables. Table 7.5 shows the word counts, whereas Table 7.6 shows the utterance counts, both collected for the two treatments.

The percentages of utterance counts for both treatments are rather low. For example, in Table 7.6, 18.6% (percentage of utterances with d=b, with treatment 1 and direct duration type) indicates that only a small portion of the utterance would benefit from their high duration scores. With this low percentage of an utterance, for the rest of the utterances the transcriptions with even fewer correct words than the top best (a) will be chosen in the re-

Table 7.5 Percentage correct word scores (and the word accuracy scores between brackets), collected from transcriptions with various marks (e.g., for "d", the transcription with the highest duration score for each utterance is used). Scores are shown for two treatments each with three types of utterance duration scores, on the TIMIT test set. The "best possible" score is collected from the best transcription (b) for each utterance. "⇑" indicates same value as in above entry.

Best possible (b)	treatment	utterance duration type	(d)	(v)
87.68 (86.39)	1	Direct	73.51 (63.85)	81.63 (75.64)
		Global	73.41 (64.02)	⇑
		Local	73.75 (64.63)	⇑
	2	Direct	70.18 (63.52)	67.73 (65.64)
		Global	70.07 (63.22)	⇑
		Local	73.21 (63.95)	⇑

Table 7.6 Percentages of utterance counts, collected from transcriptions with various specific situations of mark coincidence (e.g., situation "d=b" only includes those utterances with a transcription that has both highest duration score (d) and most correct words (b)). Note that for those situations *irrelevant* for duration (d), the percentages of utterance counts are the same (indicated by "⇑") for the three utterance duration types .

treatment	utterance duration type	d=b	d=b=a	d=w	v=w	b=a	d=a	v=a	d=v	d=v=b
1	direct	18.6	14.6	27.8	66.7	60.9	15.0	43.6	22.0	16.3
	global	19.6	15.6	27.7	⇑	⇑	16.4	⇑	22.7	17.4
	local	19.1	15.6	27.3	⇑	⇑	16.4	⇑	23.2	16.9
2	direct	22.4	19.2	21.9	17.2	60.9	20.5	29.5	18.8	15.3
	global	21.8	19.0	20.2	⇑	⇑	20.5	⇑	19.1	15.3
	local	20.0	15.6	32.9	⇑	⇑	16.2	⇑	15.0	12.5

[17] We use the notation "$a=b$" as a sufficiently clear shorthand for $T(a)=T(b)$.

scoring. Then the total word score, after re-scoring, may be higher or lower than the "lower limit" 79.07% in Table 7.2, i.e., the word-correct score of the top best without *N*-best process.

On the other hand, an interesting phenomenon can be seen in Table 7.5. If we use the mark "v" to choose one out of the N transcriptions for each utterance, the word score with treatment 1 (81.63%) is already higher than the "baseline" (79.07%) without *N*-best re-scoring. This means that one can increase word recognition scores by merely using s_V.

7.5 Re-scoring and results

When we obtained the duration scores s_D for all the N transcriptions of an utterance, the process of re-scoring is simply to choose the transcription with the highest total score s_t:

$$s_t = s_a + w_D s_D,$$

where s_a is the acoustic score from the first phase recognition, w_D is the duration weighting factor, and s_D is one of the three types of utterance-level duration scores s_D^D, s_D^G or s_D^L (Subsection 7.4.1), respectively. We applied this re-scoring process to all the 1,344 TIMIT test utterances with various w_D optimised for each treatment (and w_G for the "global" type of utterance duration score). For the results presented in this section, the optimisation was performed on the test set. The best word scores for different conditions are shown in the column "full" of Table 7.7.

As has been seen above, the formal way of re-scoring shown above to choose the new top is based on the total score s_t of the transcription (we call it "full" re-scoring). From the rather small score improvements of this re-scoring method, one can argue whether there are other scoring methods, that are not exclusively based on s_t. We call these re-scoring methods "selective", because they select some of the utterances for re-scoring, based on various marks available for each utterance. We can have a total of six selective re-scoring methods that each uses a different mark situation. These methods can be put

Table 7.7 Percentage word correct (and accuracy) scores for TIMIT test set, with two treatments, each with three utterance duration types (u.d.t), and re-scored with either "full" or "selective" methods. Four different mark situations are used for the selective methods. Scores in shading are lower than the base-line top score (79.07%/76.97%) without re-scoring.

treatment	u. d. t.	full	selective			
			v=d	v≠a	d≠a	v=d≠a
1	direct	79.13 (76.99)	79.90 (77.21)	77.55 (71.19)	73.52 (63.85)	79.90 (77.21)
	global	79.11 (77.01)	79.78 (77.09)	77.47 (71.18)	73.40 (64.03)	79.78 (77.09)
	local	79.09 (76.95)	79.85 (77.15)	77.70 (71.60)	73.76 (64.63)	79.85 (77.15)
2	direct	79.36 (77.30)	78.63 (76.54)	71.46 (66.24)	70.08 (63.52)	78.63 (76.54)
	global	79.26 (77.19)	78.54 (76.44)	71.42 (66.03)	70.11 (63.27)	78.54 (76.44)
	local	79.15 (77.05)	78.92 (76.81)	74.33 (67.39)	73.21 (63.95)	78.92 (76.81)

into two groups:

1. selecting a transcription based on mark combination other than s_t;
2. selecting a transcription based on mark combination and not on top (≠a).

The first group includes "v", "d" (already shown in Table 7.5), and "v=d". The second group includes "v≠a", "d≠a", and "v=d≠a". The word scores by the last four selective methods are all shown in Table 7.7.

For all cases with treatment 1, all the selective re-scoring methods give better improvement than the treatment-2 re-scoring process. This is an indication that s_V and/or s_D of treatment 2 (within-word forced-Viterbi and no word-juncture models) are less accurate than treatment 1. An opposite situation is observed for "full" re-scoring, where the scores for treatment 2 are higher than treatment 1. The various s_D of treatment 2 are obtained with the same phone sequence as in obtaining s_a, whereas in treatment 1 the phone sequences are generally different (due to word-juncture models). In a full re-scoring process, where both s_a and s_D are used, this disagreement in phone sequences can lead to a worse result in re-scoring.

Among the word scores by the selective methods in Table 7.7, only for "v=d" and "v=d≠a" with treatment 1, the scores increased from the base-line scores. Furthermore, scores for these two methods are exactly the same. This means that, based on the actual marks, all those transcriptions with "v=d" are implicitly not on top (≠a) of the original N transcriptions. In conclusion based on Table 7.5 and Table 7.7, the best re-scoring results[18] are 81.63% (75.64%) with "v", and 79.90% (77.21%) with "direct" selective method and "v=d", both from treatment 1. These are both marginal improvements from the base-line scores without re-scoring: 79.07% (76.97%). The fact that the score with "v" alone is higher than score with "v=d", is an indication either that s_D generated is not sufficiently accurate, or that s_D and s_V do not match each other well for an utterance. Future study is required to verify and optimise this.

7.6 Conclusion and discussion

In this chapter, the context-dependent (CD) durational knowledge is built into parametrical duration models based on four contextual factors (rate, stress, positions in word and in utterances). This CD durational knowledge is incorporated into the recogniser (using monophone HMMs) in the post-processing using N-best transcriptions. Special technical treatments were applied to cope with the practical problem of having an N-best program that

[18]More careful engineering in optimising the two-phase re-scoring procedures may further improve the word scores (including the parametrical CD duration models, word-juncture models, different combinations of aspects in making the two treatments, and finer adjustment of the weighting factors).

only generates word-level transcriptions, whereas the duration model is at the phone level. Marginal improvement in word recognition scores is achieved, after preliminary optimisations on the various system settings.

Duration scores of phone instances in the *N*-best transcriptions are the basic forms of durational knowledge to be incorporated into the whole recognition process. The whole procedure of incorporating such a durational knowledge given the available two-phase *N*-best algorithm, can be summarised into the following six steps, as presented in this chapter:

1. Build CD duration models for all phones under the chosen four factors;
2. Generate the sequence of phone instances from the word-level *N*-best transcriptions, based on the lexical norm pronunciation plus a word-juncture model;
3. Estimate phone duration using a forced-Viterbi procedure on the phone sequence;
4. Obtain a phone duration score based on the phone identity and contexts from (2) and duration from (3), using the CD duration model of (1);
5. Accumulate an utterance-level duration score based on phone duration scores in (4);
6. Combine the acoustic score from the first-phase *N*-best recognition with the duration score in (5), for each transcription, and find the new top.

The technical limitation of having to use a two-phase procedure in generating phone-level transcriptions, actually gave us a chance to look in more detail at the effects of various aspects, such as the effect of pronunciation variations at word boundaries. In most sub-word-based word recognisers, as in our own, the phone sequence for each word is defined as the single norm pronunciation (plus a few specific variations). Although it is desirable to model a full range of pronunciation variations, it is impractical to do so in the process of recognition, simply because of the huge number of variations for all the words in the lexicon and the huge number of word connections. With our two-phase procedure, the first-phase generates a single word sequence in each of the *N*-best transcriptions, based on a single norm pronunciation. Modelling the pronunciation variation for this word sequence in the second phase, then becomes simple (furthermore, we only model it at word junctures). Recognition scores after the re-scoring show that using word-juncture models is better than not using them.

Appendix 7.1 Markov model-based durational pdf

In this appendix we give the steps for obtaining the binomial-like durational pdf using an underlying Markov model. Note that this Markov model does not necessarily have any relation with the HMMs used in the recogniser, except that both govern a durational pdf to be a binomial-like function.

Given the durational statistics mean μ and variance σ^2 of a phone in a specific contextual situation, the required number of selfloops n in a linear Markov model is obtained in a way similar to that in Subsection 5.3.2 of Chapter 5. The fitting condition here is further simplified to allow also the selfloop probabilities a_i, $i = 1,2,\cdots,n$, to be equal (the region of fitting (μ,σ^2) is larger), since we do not have to fit the acoustic part of the Baum-Welch equations. The lower limit used for choosing n is then changed from $\tilde{n}_{\min}$ to $n_{\min} = \mu^2/(\sigma^2+\mu)$. We now look for the possible region of σ^2 for given n and μ. From the basic relations between these variables in Chapter 4

$$\mu = \sum_{i=1}^{n} \frac{1}{1-a_i}; \qquad \sigma^2 = \sum_{i=1}^{n} \frac{a_i}{(1-a_i)^2},$$

one extreme situation with equal a_i is

$$\sigma_1^2 = \frac{\mu}{n}(\mu - n).$$

For another extreme we let all but one a_i approach zero and the last one a_n vary freely (the location of a_n in the model is actually irrelevant to the durational statistics, see Chapter 4). This gives

$$\mu = (n-1) + \frac{1}{1-a_n}; \qquad \sigma^2 = \frac{a_n}{(1-a_n)^2}.$$

From these we solve

$$\sigma_2^2 = (\mu - n)(\mu - n + 1),$$

and it is easy to verify that $\sigma_2^2 > \sigma_1^2$. Then the fitting region for σ^2 is

$$\frac{\mu}{n}(\mu - n) \le \sigma^2 < (\mu - n)(\mu - n + 1),$$

where the right-side strict inequality indicates that the upper border (all but one a_i being zero) cannot be reached. The fitting areas for some example n values are depicted in the upper panels of Figure 7.9. The following relations between the parameters can be seen in the figure. (1) For a given n, a small μ allows only a small region of σ^2 to be fitted (difficult); (2) For a given μ, very large (flat distribution) and very small σ^2 (sharp) can only be fitted with a relatively lager n.

The durational pdf of the context independent (CI) HMMs are fitted in Chapter 5 using numerical solutions with special initial points, which do not allow equal selfloop probabilities. It is better, then, that the context dependent (CD) HMMs in this chapter also use the same constraint, so that the CD HMMs and their CI counterpart have a good match in duration. The Markov-model based duration models then should also take this constraint. For this we take another lower limit in choosing n (copied from Appendix 5.3):

$$\tilde{n}_{\min} = \frac{\mu(\mu-1)+\sigma^2}{\sigma^2+\mu-1}.$$

The possible range to vary σ^2 also becomes tighter:

$$\frac{\mu-1}{n-1}(\mu-n) \le \sigma^2 < \begin{cases} (\mu-n)(\mu-n+1), & n=3; \\ \dfrac{(\mu-n)(\mu-n+2)}{2}, & n>3. \end{cases}$$

The left-side "≤" is from a value in the numerical procedure which is allowed to be zero. The right-side strict inequality is from the constraint that $u_i > 1$ required via the transformation $u_i = 1/(1-a_i)$. The above ranges are shown in the lower panels of Figure 7.9, and are used in the tests of this chapter.

When n is determined, we further choose $\{a_i\}$ that satisfy the given (or modified) (μ, σ^2). In the whole fitting area, an infinite number of different combinations of $\{a_i\}$ values will satisfy (μ, σ^2). We choose the simplest combination among them in which all but one a_i are set equal $a_i = a_1$, $i = 1, 2, \cdots, n-1$, while the last one a_n is left free to vary. This gives

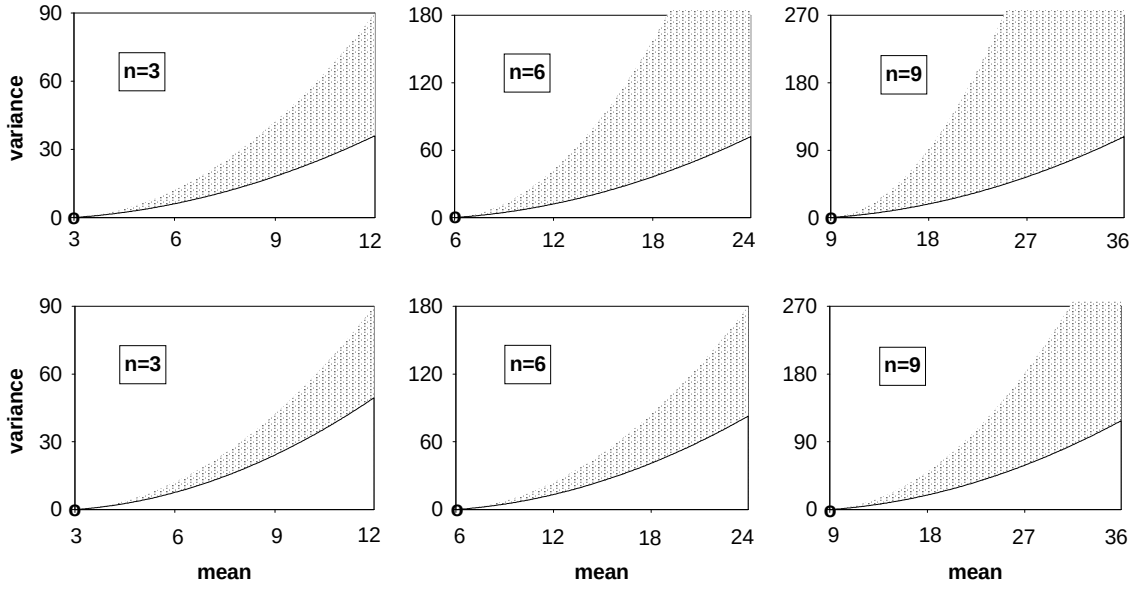


Figure 7.9 Shaded areas show the possible regions to fit duration mean and variance, given the model lengths (number of selfloops n=3, 6 and 9, respectively). For generality purposes, the unit for the mean is expressed in time steps (so if the actual time step is 8 *ms*, mean=10 in the figure denotes 80 *ms*), whereas the unit for variance is in (step)^2, (var=100 translates to 6400 $(ms)^2$). Upper panels: Curves of the lower borders $\sigma^2 = \mu(\mu-n)/n$ are for the extreme situation with equal selfloop probabilities a, whereas the upper borders $\sigma^2 < (\mu-n)(\mu-n+1)$ are for another extreme when all but one a approach zero and the last a is free to vary (the open upper border and the small circle at the beginning of the curves indicate the impossibility to reach). Lower panels: Lower borders $\sigma^2 = (\mu-1)(\mu-n)/(n-1)$. Upper border for n=3 is the same as in upper panel, whereas for other n values is $\sigma^2 < (\mu-n)(\mu-n+2)/2$.

$$\mu = (n-1)\frac{1}{1-a_1} + \frac{1}{1-a_n};$$

$$\sigma^2 = (n-1)\frac{a_1}{(1-a_1)^2} + \frac{a_n}{(1-a_n)^2}.$$

Using the same transformation $u_i = 1/(1-a_i)$ as in Chapter 5 to simplify the procedure, we first obtain the solution for u:

$$u_1 = \frac{1}{n}\left[\mu \pm \sqrt{\frac{n(\mu+\sigma^2)-\mu^2}{n-1}}\right],$$

where the sign before the square root should be chosen in such a way that $u_i > 1, i = 1,n$, which is required by $a_i > 0$ via the transformation. Then $u_n = \mu - (n-1)u_1$, and the selfloop probabilities are $a_i = (u_i - 1)/u_i$, $i = 1,n$.

Appendix 7.2 Algorithm for generating new transcriptions

As can be seen in Table 7.2, the "matched" word score (89.20%) using the correct words in different *N*-best transcriptions, is higher than the word-correct score using single transcriptions (87.68%, both at *N*=30). After the first-phase *N*-best recognition, all the *N* transcriptions are available. Therefore we can try to combine the correct words from the different transcriptions into a single new transcription. In this appendix we describe an algorithm that we developed, which generates new transcriptions based on the *N*-best transcriptions. The new transcriptions were not actually used in the re-scoring process, unfortunately, since the word best scores from a pilot test were not higher than those of the original *N*-best transcriptions.

Our algorithm of generating the new transcriptions is based on the rationale that at each approximate position in time, the most frequently occurring word in the *N*-best transcriptions (see Figure 7.4) should be taken as a word hypothesis in a new transcription. All these word hypotheses with

```
   (1) Collect all the word instances in the N-best transcriptions,
merge those with the same word identities and with both beginning and
end times aligned with each other within a given margin, into a "slot".
Store the counts of the word instances for each slot, and store the
acoustic score as averaged over the input word instances. Also stored
are the beginning and end times for each slot;
   (2) Go through the whole time range of the utterance with two time
pointers for beginning and end, and on the way create new word-
transcriptions by linking the slots. When a slot is encountered by the
two pointers, link this slot to the ends of all existing partial
transcriptions which ended within a given margin of the beginning time
of the slot. If the slot has not been linked to any existing
transcription, a new transcription is created with this single slot;
   (2.1) The right-to-left between word branching is provided by (2),
while the left-to-right branching is guaranteed as follows: when a slot
is linked to the end of each partial transcription, this partial
transcription before linked with this slot is cloned, making it ready to
be linked with other slots. Each just-extended partial transcription is
checked with a threshold to see if it has to be pruned here;
   (3) The total instance-count of all the slots for each created
transcription is averaged by the number of words in the transcription. A
required number of transcriptions with both beginning and end times
falling within the margins of that of the utterance, and with the
highest "averaged word instance counts" are output, together with the
new acoustic scores for the words. The acoustic score for the
transcription is accumulated by the scores of the words.
```

Figure 7.10 Algorithm of generating new word-transcriptions.

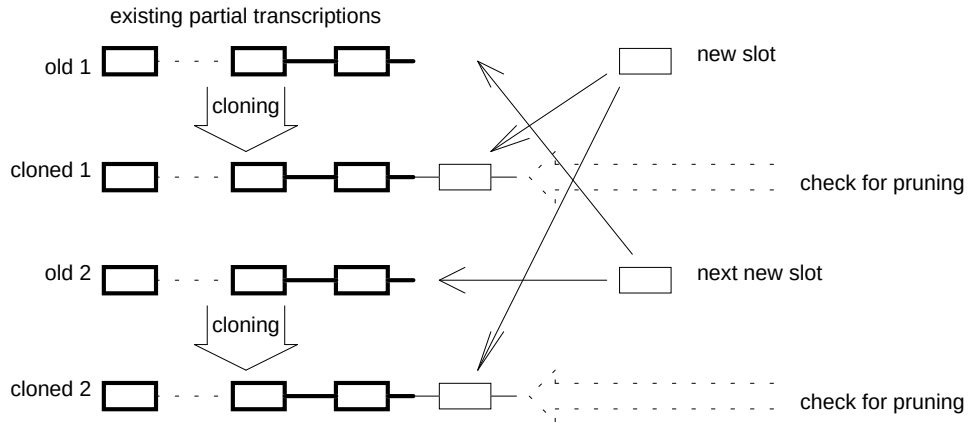


Figure 7.11 Process of linking the word slots into transcriptions. The left part drawn in thick lines shows the existing partial transcriptions, where each block denotes a word. "Old 1" and "old 2" are two existing partial transcriptions before being linked with the "new slot", whereas the "cloned" ones are their copies *before* linking. The new slot (thin line) is appended to the two cloned version transcriptions and they are checked for pruning. The original versions are ready for the next new slot (if any).

beginning and end times matched to other hypotheses are linked to form a sentence transcription. The actual algorithm that has been developed and that has been shown to work is given in Figure 7.10.

The concept of left and right between-word branching is illustrated in Figure 7.11. The idea is that on both forward and backward directions, each word slot must be linked with all the other word slots which fall within the time margin. "Branching" in the algorithm above involves the situation that one word is linked with multiple words. However, the actual branching is achieved by *copying* either the single slot (from the right side) or the whole partial transcription (from the left side). This is simply because each different linkage must result in a new transcription. What we need is not a compact linked graph, but all the distinct transcriptions.

Pruning of new transcriptions is needed to deal with limitations of computer memory[19]. The pruning mechanism determined empirically is performed on each partial transcription just linked with a new slot. The "averaged word instance counts" of the partial transcription is compared with a pruning threshold, that is determined by the numbers of word instances and total slots, and by the current usage of the memory. The threshold is increased iteratively for a heavier pruning if a fixed memory allocation is reached. By adjusting the threshold, always a maximal number of full transcriptions allowed by the memory is generated.

[19]A typical utterance will have $K = 30$ slot positions, and $s = 10$ slots per position, for $N = 30$. Then, without pruning, a total of $s^K = 10^{30}$ new transcriptions will be generated.

	we	**always**	**thought**	**we**	**would**	**Die**	**with**	**our**	**boots**	**on**
0	will	it		the		Riverbank		or	redwoods	**on**
1	will	it		the		Riverbank		or	**boots**	**on**
2	will	it	now			Riverbank			redwoods	**on**
3	will	it	are			Riverbank			redwoods	**on**
4	will	is	the			Riverbank			redwoods	**on**
5	will	it	now			Riverbank		or	**boots**	**on**
6	will	it	the			Riverbank		or	to	some
7	will	it	are			Riverbank		or	**boots**	**on**
8	will	is	the			Riverbank		or	**boots**	**on**
9	will	it	the			Riverbank		or	put	**on**
10	will	it	the			Riverbank		or	day	some
11	will	it	now			Riverbank		or	to	some
12	will	it	the			Riverbank		or	put	some
13	will	it	can			Riverbank			redwoods	**on**
14	will	it	are			Riverbank		or	to	some
15	will	it	the	worm		Another			**boots**	**on**
16	will	it	now			Riverbank		or	put	**on**
17	knows		the			Riverbank			redwoods	**on**
18	will	is	the			Riverbank		or	to	some
19	will	it	now			Riverbank		or	day	some
...										
...										
97	will	it	the	**we**	the	**Die**		another	**boots**	**on**
98	will	it	the	worm	a	Giant		or	**boots**	**on**
99	will	is	can			Riverbank		or	to	some

Figure 7.12 Some new transcriptions output from the algorithm for (test/dr3/fpkt0/si908). Correct words are printed in bold, and aligned. Transcription at position 97 (shaded) has four correct words as compared with three correct words in the original best transcription.

The time margin must be determined empirically. It must be large enough to allow the timing deviations of the same word in the different input transcriptions. However, a margin that is too large would allow too many wrong alignments. The final margin is chosen to be 3 time frames in both directions. Some output transcriptions (out of the total of 100 for each utterance) of this algorithm for the same example utterance in Figure 7.4 is shown in Figure 7.12.

8

GENERAL DISCUSSIONS AND FINAL CONCLUSIONS

Abstract

This chapter concludes this thesis, and puts it in a broader perspective in general automatic speech recognition (ASR) research. Insights obtained in this study about durational behaviour of the HMM, the complicated nature of the context-dependent durational distribution in a database, and the interactions between the HMM and the distribution, are summarised. Concrete results achieved in improving the recognition scores are also presented, together with the technical limits. A general notion is presented on knowledge incorporation into machines, which is derived from the experience of the current study on incorporating the specific durational knowledge into statistical ASR. Implications for possible future research are discussed.

8.1 Introduction

In this thesis, we have tried to incorporate durational knowledge into a statistical automatic speech recognition (ASR) system. The knowledge-based approach and the statistical approach have traditionally been two distinct approaches in ASR. Therefore our job of trying to combine the two approaches was not very straightforward. Special ideas and techniques necessarily had to be developed. In the previous chapters, we have concentrated mainly on the specific details of the individual attempts, thus the reader still deserves a more systematic conclusion that will hopefully be insightful. In this chapter, we first conclude what we have actually achieved in incorporating durational knowledge (Section 8.2), then we develop a general notion about knowledge incorporation into machines, as derived from our experience (Section 8.3). Section 8.4 discusses various limits of the current study. The closing section points to some future studies related to the current study.

8.2 Concrete achievements in durational knowledge incorporation

In this study, we chose to incorporate the knowledge on segmental duration into the HMM-based statistical ASR system. The "baseline" HMM system, before any of our specific knowledge is incorporated, consists of a continuous-density recogniser with monophone HMMs. All improvements are analysed relative to this baseline system. After all our various attempts, which were presented in the technical chapters (3 to 7) of this thesis, we can now see what has been achieved. The achievements can be listed in terms of answers to the following questions:

1. Is durational knowledge incorporation necessary and possible in HMM-based ASR or not;
2. How can such an incorporation be achieved; and
3. Does this incorporation improve the recogniser's performance or not.

These questions are discussed in the following subsections. In Subsection 8.2.1, only the overall searching path of the project will be discussed, whereas the actual improvements in performance scores will be presented separately in Subsection 8.2.2. In Subsection 8.2.3, results from analyses of context dependent duration will be presented. Finally, Subsection 8.2.4 discusses the contribution of the current study to ASR research in general.

8.2.1 Searching path of the project

At the very beginning of this thesis project we realised that there was a lack of any existing method appropriate for durational knowledge incorporation.

Our actual practice has really been a process of looking for, and testing of, many possible methods for accomplishing this incorporation. During the project, several issues appeared to be of prime importance, which then became the guidelines of our project. The main issues can be listed in the following four points (each with a pair of related, or contradictory, aspects):

1. Separation between context independent (CI) and context dependent (CD) duration measures;
2. Searching for regularities in a speech database and analysis of HMM durational behaviour;
3. Extracting durational knowledge and incorporating it;
4. Improving the parameter values of the HMM of a conventional recogniser (old structure), and adding new structures to the recogniser.

These issues are treated in the following discussion in an interwoven way, because they are reflected in various chapters along the search path of the project.

We measured the segmental duration (of phones) in two ways, i.e., CI (context independent) and CD (context dependent). CI duration was dealt with in Chapters 4 and 5, and CD duration in Chapters 6 and 7. (The conceptual scheme in Figure 8.1 is an attempt to represent all elements of the present project). In the whole project, for both CI and CD duration modelling, the HMMs used have been monophones (for simplicity and a tangible effect of duration modelling).

Based on this separation between CI and CD duration measures, the modelling schemes for them were also different. Since CI duration was modelled by the HMMs themselves, it is important to first analyse the durational behaviour of these HMMs. In Chapter 4 we theoretically analysed the standard HMMs to check whether their CI durational behaviour can be appropriate. We first analysed the HMMs with general left-to-right topologies, and came to the conclusion that even the simple structure of linear HMMs is potentially capable of modelling phone duration quite well. Here we obtained a relation between the HMM parameters (the selfloop transition probabilities), with a given structure (linear), and the statistical parameters of the segmental duration (mean and variance). In Chapter 5, it was observed numerically that the values of the HMM parameters (as trained with the standard Baum-Welch maximum-likelihood (ML) algorithm) are not always optimal in modelling the CI-durational distributions. Then these values were further modified via an indirect method, which however fits well in the fundamental mathematics and algorithms of the HMMs. This method was a Baum-Welch algorithm modified to include extra constraints on the durational statistics measured from speech data. The durational behaviour was improved, as shown in Chapter 5, both in durational pdf fitting with the data histograms, and in the performance of automatic segmentation.

However, such an improvement was not propagated consistently to a better performance of word recognition (see the next subsection for a discussion of these scores).

Due to the fact that the actual behaviour of segmental duration in speech is very much context dependent, the CI duration modelling by the HMMs themselves was then judged to be insufficient. The useful information about CD duration regularities in speech was not used, and our knowledge on CD duration was not incorporated. Because we used monophone HMMs in the current study, it was then decided to model the CD durational behaviour by an *external* duration model, rather than by the HMMs themselves. Therefore we did not have to analyse the HMM behaviour as before. We only had to analyse the speech data for the purpose of building the external duration model. It was insufficiently known *which* "contextual factors" are important for a specific multi-speaker speech database (we refer to this situation as "unknown structure" of the duration model). Both the structure and the parameters of this CD duration model were sought in Chapter 6, under various analysis frameworks, but always being separated from the HMM recogniser itself. However, selection of the factors or building of the structure was also based on the concern that in the next step, these should in principle be applicable in the actual recognition process. Judging from the durational distribution on separate factors using the actual database (TIMIT), some factors (as part of the model structure) showed to be informative (speaking rate, stress and locations) whereas others were abandoned (voicing of post-vocalic plosives). Some other factors (e.g., phone categories) already exist in the HMM recogniser; therefore it would be redundant to include them explicitly at the contextual level.

In Chapter 7 it was described how the CD-durational knowledge, in terms of the four chosen factors, was incorporated into the recogniser by using the external CD duration model, as part of an additional structure to the conventional HMM recogniser. The additional structure consists of a CD data duration model, and a post-processing mechanism which uses the *N*-best sentence hypotheses from the first-phase recognition using the monophone HMMs. The additional durational knowledge is combined with other knowledge in the system in a process of re-scoring the first-phase hypotheses. The recognition score was improved marginally by incorporating this knowledge (see the next subsection for a discussion of these scores).

The conceptual scheme in Figure 8.1 serves as a compact illustration of the project (mainly Chapter 4 to 7), concerning the CI (above the dashed line in the figure) and CD duration modelling (below the dashed line). In all phases of the work shown in this figure, the TIMIT database is used throughout. Chapter 3 of this thesis was concerned with optimisation on the front-end processing for a good baseline system. This chapter is not directly related to duration modelling, and is thus not shown in Figure 8.1. Finally,

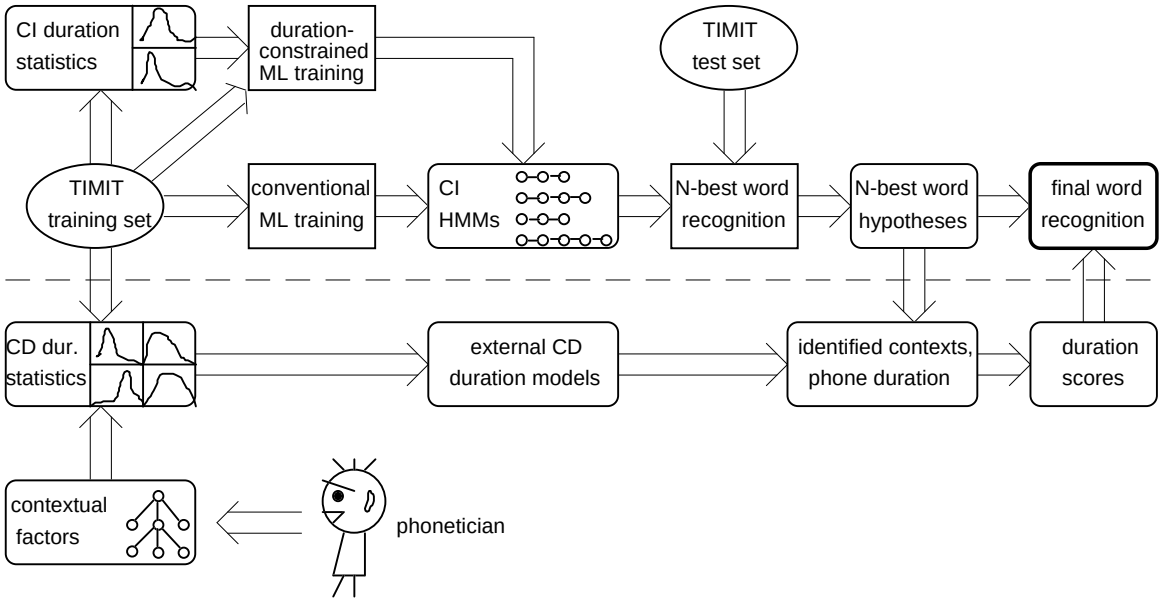


Figure 8.1 Conceptual scheme of context independent (CI, above the dashed line) and context dependent (CD, below the line) durational knowledge incorporation in HMM-based recognition, together with the process of collecting CI or CD durational statistics. Although the phonetician (or any other speech scientist) is designing the "contextual factors" at this moment, he is actually responsible for the whole process shown in this figure.

much of the mathematical and technical development, which represents a substantial and essential part of the efforts spent in this thesis project, has been put in the appendices at the end of each chapter.

8.2.2 Score improvements and technical limiting factors

All the scores presented here are based on the tests performed on the 1,343 `sx` and `si` utterances of the TIMIT test set. Only the accuracy scores (for words or phones) are presented here, whereas other scores and more detailed analyses can be found in the technical chapters. Furthermore, as is the usual way to report scores (or improvements) in ASR literature, it will not be indicated whether each score leads to a significant change from the baseline, in the sense of a specific statistical test[1].

The activities presented in Chapter 3 mainly addressed the optimisation by means of linear transformations on the front-end pre-processing parameters. These have resulted in slight differences in performance scores between the processing schemes with a (global or state-specific) transformation and without any transformation (original). The impact also differs for word and phone recognition. The word-accuracy score decreased

[1]One may calculate (e.g., Chollet, 1995) a significance-interval at 95% level as $p \pm 1.96\sqrt{(p(1-p)/N)}$ where p is the baseline score in percentage and N is the number of instances in the test (11,025 word instances and 51,681 phone instances in TIMIT test set). Therefore for p=80% of word scores, the significant difference in scores is about 0.75% and for p=66% of phone scores, it is about 0.4%.

from 82.51% original to 81.64% with the global transformation and to 81.90% with the state-specific transformation. The original phone-accuracy score of 66.02% decreased to 64.66% with the global transformation, and increased to 66.85% with the state-specific transformation. (These tests already used the variable lengths n for phone HMMs chosen to be suitable for duration modelling in Chapter 5). These different front-end processing schemes were used in Chapter 7 for the two different treatments which emphasise either a phone-level or a global-level optimality. A more convincing increase in scores was hampered by the fact that the particular technique of linear transformation only removed parameter correlation. However such a removal would only improve the modelling accuracy significantly, if the data distribution in the transformed space coincides with a mixture of *uncorrelated* Gaussians. This is generally not the case.

The CI-duration modelling by durationally constrained training in Chapter 5 resulted in a somewhat complicated situation. We compared scores using three different sets of HMMs: (1) with fixed length n=3; (2) with different length n chosen to be suitable for each phone; and (3) with further durationally constrained training. For automatic segmentation of phones, the percentage of correct segments within a 20 *ms* margin in both directions increased from 84.6 using fixed n=3 to 86.9 with the suitable n and further increased to 87.1 with the constrained training. This is an indication of the positive effect of duration modelling at phone level. The phone recognition accuracy score increased slightly from 65.68% using fixed n, to 66.02% with suitable n, but then slightly decreased to 65.99% with the constrained training. For word recognition tests, the accuracy scores decreased from 85.23% with fixed n to 82.51% with suitable n, and to 82.60% with further constrained training. A comparison of the performances for phone segmentation, phone recognition and word recognition gives an indication that the CI-duration modelling only leads to improvements of the phone-level performances. One of the main limits for score improvement is that the linear HMM topology with the suitable n, chosen on the basis of duration mean and variance, did not always allow a good fit for the "minimal duration" of each phone. The reduced scores above are an indication of the seriousness of this problem, for instance the situation when a phone instance in the test set is really shorter than the minimal duration of that phone HMM. A solution to this problem might be to use HMMs with skip transitions. However, this would cause the technique of Chapter 5 to become too complicated (the complicated situation was shown in Appendix 4.1 of Chapter 4). Another limitation comes from the numerical method used to solve the constrained maximum-likelihood equations, which did not allow for accurate duration fitting of all 50 phones.

The N-best re-scoring using the external CD duration model in Chapter 7 gave rise to a marginal improvement in scores. The best result among all

different test conditions was an increase from 79.07% word-correct of the baseline to 79.90% with re-scoring (although the highest theoretically possible score by optimal use of the 30 transcriptions per utterance was 87.68%). For this test condition, the phone duration scores were estimated based on a phone sequence for each word-level transcription. This phone sequence was generated based on the lexical phone sequences of all the words in the transcription, plus a word-juncture model that modifies the phone sequences at the word boundaries. This word-juncture model, also estimated from the database, accounts for the phone variations most frequently occurring at the word boundaries due to for instance coarticulation. Thus the phone sequence generated with the juncture model was closer to an actual phone sequence than the norm sequence was. Greater improvement was mainly hindered by technical reasons, such as the fact that we had to use a two-phase re-scoring procedure. More careful engineering (such as more accurate CD duration models, and further optimisation of the weighting factors for combining the duration scores and acoustic scores) will further increase the word-correct scores.

8.2.3 Analysis of CD durational statistics

The CD statistics of vowel duration were analysed in Chapter 6 using the TIMIT database. In order to collect durational statistics based on factors like stress and syllable locations, information on these kinds of properties of the phone instances was needed. However, the stress information is only given as one norm form per word in the TIMIT lexicon, whereas the actual phone sequences in the utterances deviate seriously from the norm. In order to copy the word stress information in the lexicon onto the exact phone sequence for each utterance, a dynamic programming (DP) procedure at the symbolic level was used. This DP finds an optimal match between the norm phone sequence with the actual phone sequence in an utterance. This DP was also modified in such a way that vowel instances in one sequence are matched with vowel instances in the other sequence as much as possible, because vowels may be pronounced as other allophones, but mostly they are not deleted.

Two methods were used for the analysis of durational distributions, i.e., an analysis of the *individual* contextual factors and an analysis of variance (ANOVA) of *all* the eleven chosen factors. First, durational distributions on individual contextual factors were obtained, such as stress, syllable position within word and within utterance, and speaking rate. Most factors were found to be significant in influencing durational distributions. The voicing of post-vocalic plosives did not show a systematic effect on vowel durations in the many-speaker TIMIT database, contradictory to the usual assumption that voiced plosives cause the preceding vowels to be longer than voiceless plosives do. This fact is attributed to the effect of too much speaker-variation

in the database and insufficient speech material per speaker. Based on this finding, this last factor is not included in the duration models in Chapter 7, where CD durational knowledge is incorporated.

The purpose of the second method (ANOVA) was to reveal the relative importance of all the factors in one integrated analysis. This analysis includes a total of eleven factors, amongst which also the speaker-related factors such as gender and dialect region, and the phone-category related factors such as phone and phone in immediate contexts. After solving the technical difficulties for the ANOVA calculation, we obtained useful insight into the importance of the order of the eleven factors in the totally nested ANOVA design. This nested design was taken for technical simplicity. However, the resulting percentages of the "sum-of-squares" in the eleven factors showed too much irregularity. This is also attributed to the very large number of cells in some of the factors and the still rather limited amount of data (number of the phone instances) in TIMIT. Therefore the numerical results of the 11-factor ANOVA were not used to guide the choice of the final four factors to be included in the CD duration models in Chapter 7.

8.2.4 Contribution of the current study in ASR research

There are in general many different approaches in ASR research to improve the modelling accuracy and system performance, as briefly reviewed in Chapter 1. The approach in this thesis was first to choose an aspect of speech knowledge as given in terms of a phonetically defined parameter (segmental duration), and then to find various possible ways to incorporate the chosen type of knowledge into the ASR system. Although the knowledge is defined "phonetically", the actual representation of that knowledge (in terms of number of factors and values of parameters of durational statistics) has been obtained from a general-purpose speech database. Therefore, our approach can be characterised as incorporating *statistically-represented* knowledge into a *statistical-structured* (HMM) system. Although both the knowledge and the recognition system are statistical, the definition of the structure (or skeleton, or "determiner") of the knowledge has been influenced by the phonetic theory on duration (see the phonetician in Figure 8.1).

Although during the development some of the system components of the standard HMM-based recogniser have been modified as well, all such modifications have been motivated by the global aim of duration modelling, rather than by other purely technical criteria of the HMM, as is the case in some other research activities in ASR. Therefore the experience of the current research can serve as complementary, or as an alternative, to other types of research in ASR.

In the general introduction of Chapter 1, we mentioned the other two major approaches in duration modelling for HMM-based ASR, i.e., the hidden

semi-Markov models (HSMM) and minimal-duration approach. Our approach in this study is totally different from both of these approaches. A comparison between our recognition score improvements and those of these two approaches would be difficult because the recognition tasks and other aspects are very different. For example, using only a constraint of minimal phone duration, Gupta et. al. (1992) reduced the word error from 9.2% to 7.2% in a large vocabulary isolated-word task. This improvement was obtained mainly due to a correction in errors made in the Viterbi search. It is apparent that such a correction cannot be equally beneficial for a continuous-speech task, since then words follow each other in the search. On the other hand, no other durational information is used, besides the minimal phone duration. The approach of HSMM, on the other hand, is not used often, mainly due to its complexity (up to D (maximum duration) times of HMM in training, where the typical D is 25 frames, see, e.g., Mitchell et al., 1995). The performance improvement using HSMM can be typically shown in a very-small vocabulary isolated-word task in Russell and Moore (1985), where the error score reduced from 26.5% to 14.7%. However, because the test set only contained 100 repetitions of 3 word-pairs, such as /league, leek/, to emphasise the importance of the duration cues, the usefulness of the approach cannot be easily extended to other recognition tasks.

The type of research in the current study was not necessarily aimed at a purely technical improvement of the ASR system performance, but rather involved a methodological study of durational knowledge incorporation into a statistical system. Therefore, the main contribution of the current study to the general ASR research is not in terms of specific techniques that lead to an optimal recognition performance. Instead, the main contribution is an answer to the question as to whether incorporation of durational knowledge, which is inspired by phonetics (independent of the HMM system), can in principle improve the system performance. The answer is yes: but only once the difficulties of knowledge representation are solved using techniques such as ours, not just because the principle is correct. The particular techniques we used are not necessarily the only ones for the purpose of durational knowledge incorporation. However, our experience in solving the difficulties, as well as our knowledge of what kind of difficulties one encounters when trying to incorporate knowledge into statistical systems, may be insightful for similar research.

The answers to the three basic questions at beginning of Section 8.2 can be summarised as follows:

1. Durational knowledge incorporation into the HMM-based ASR is advisable, since the standard HMMs and the whole recogniser lack this knowledge, and is possible, as shown by the CI and CD techniques;
2. Incorporation is achieved by analysing where the knowledge is missing in the HMM-recogniser, and where in the recogniser the particular

durational knowledge can be incorporated in certain representation forms (CI pdf's and CD models);
3. The current techniques improved recognition performance marginally.

8.3 General limitations of the current study

First of all, ASR is currently a very complicated technique, which in principle contains many system components, each of which requires optimisation. We have mainly concentrated on one single source of knowledge (duration) to incorporate, whereas all other system components were left in their standard implementations. Therefore it was impossible to reach the global optimum in performance.

In the course of this research project, we have tried to keep most aspects as simple as possible. For example, we only used monophone HMMs for the tests of incorporating both CI and CD durational knowledge. If triphone HMMs would have been chosen for CD modelling, the insight would have been very different, since then a substantial part of the context effects would be modelled *within* the HMMs. Furthermore, we only used speaker-independent recognition. Since we mainly used TIMIT as training and test material (which includes many speakers who each spoke very few sentences), a more optimal solution to cope with the great speaker variability in this database would have been a "speaker adaptive" system setup, in which a systematic grouping procedures among the speakers would be involved. On the other hand, if other speech databases had been available to us which suit our purpose of duration modelling better, perhaps more insight would have been obtained.

We have also tried to use all the basic system components of our recogniser as "standard" as possible, while avoiding any improved components as reported in the literature. Some of these improved components worth mentioning (because they would indeed introduce a substantial difference from the standard implementation that we used for duration modelling) are the following. We only fitted the durational statistics at the level of the whole phone HMMs. If we would have started with the HSMM, the whole fitting procedure would be a global optimisation on duration fitting at both the state level and the model level. More complicated and careful engineering would be needed, that might lead to better performance. In this respect the so called segmental HMMs might also lead to different results if we had based our duration modelling on them. Finally we would like to mention that the search algorithms we used were very straightforward Viterbi (and its N-best variant). Other search strategies would certainly lead to differences, not only in speed, but also, and more importantly, in a totally different way of incorporating the durational knowledge (e.g., not in the post-processing phase), thus leading to different insight and scores as well.

The limited time available for this project also prevented us from exploring the current and other methods further to achieve better results in many aspects.

8.4 A general (philosophical) notion on knowledge representation and incorporation into machines

The current state of the art in ASR is that the statistical approach outperforms the rule-based "knowledge engineering" approach. However the debate will certainly go on for quite a while before definite conclusions (if any) can be drawn on which approach is better. Our current viewpoint is that both approaches apply knowledge about speech in the recogniser but do *represent* that knowledge in different ways (rules versus, e.g., our pdf's). After our specific technical implementation of the knowledge incorporation, we are now able to see how the durational knowledge can be represented in the HMM-based statistical system, and how such knowledge can be used by the recogniser to perform better.

It is an important issue how to incorporate any human knowledge into machines in order to make them work better. To put our experience about durational knowledge in this wider context, it is now clear that the main difficulty (especially at the beginning) is the *representation* of knowledge. In a rule-based recogniser, knowledge is represented in a form that is easy to understand by humans, i.e., the explicit mappings between conditions and actions. However such a representation is difficult to implement in a statistically-based system.

Another problem in searching for a representation form for specific pieces of knowledge is that, we, as developers of the machine, must also understand the knowledge in the given representation. To use the concrete example of our durational knowledge incorporation, the knowledge in the given representation must also be understandable for phoneticians.

A combination of the two above-mentioned problems of knowledge representation for humans and for machines, is the key to solve the problems. We believe to have found one solution, i.e., a representation both understandable by humans and by machines. This representation is "structure plus parameters". Of course, the structure must fit both the data *from* which the knowledge is *extracted*, as well as the machine *into* which the knowledge will be *incorporated*. When the knowledge as well as the machine are statistical, the parameter values will be obtained by appropriate estimation methods. We are not able to show that this way of knowledge representation is the optimal one, but it has been shown to work with our specific practice. We can recall that at the beginning of the project, without knowing whether the HMMs (and the recogniser) already behave optimally in duration modelling, and if not, where the possible target structure is in the

recogniser into which to allocate durational knowledge, it was not clear how to link the phonetic durational knowledge with the HMM-recogniser.

Using the structure-plus-parameter representation for the purpose of adding new pieces of knowledge, sometimes the parameters of the existing structure can be modified, and in other situations new structures will have to be added. Under the new structure, of course, new parameter values should be obtained. Thus the whole process towards an optimal knowledge representation may be iterative. The added structure may be inspired by human observations of external phenomena, such as the practice of Deng (1996) to add a speech production model into the HMM-based recogniser structure (and to tune its parameters). The author of this thesis views such a practice as a good direction in ASR research.

It is our belief that the general structure-plus-parameter representation of knowledge, as well as the iterative way of knowledge incorporation, has indeed a general applicability, although so far it is only an extrapolation from our own specific experience of durational knowledge incorporation. In this study, we first of all checked whether the existing structure of HMM transition topology and the probability parameters already contained sufficient knowledge on the duration of whole segments. This not being the case, we then incorporated CI durational knowledge into the existing structure by using constrained training that altered the parameter values. Furthermore, CD durational knowledge was partly incorporated into a new structure, i.e., in the form of external duration models. Another part of the durational knowledge was incorporated by the re-scoring mechanism, which can also be considered as a sub-structure of the whole recogniser. Furthermore, all the fine structures sought from the database (organised by "factors") can also be interpreted as sub-structures in the duration model to accommodate the statistical durational knowledge.

8.5 Closing remarks and future studies

We have seen that the duration modelling approaches presented in this thesis are rarely seen in the literature. This is because the entry point of durational knowledge incorporation that we took to improve ASR is not the usual one, indicating that the field of knowledge incorporation is still immature. Our experience with these approaches shows the possibility of extracting both (statistical) structures and parameters from real data, and of incorporating them into an operational system (although not real-time). Probably the most useful comment about what can be learned from our experience is, that one has to analyse *both* the durational behaviour of HMM, and the durational regularities in the database, to see whether and how the chosen knowledge can be incorporated. We also showed the possibility to combine knowledge at different levels in speech (e.g., frame-level spectral versus long-term slow-

changing), which so far has been difficult for HMM-based statistical systems without using special techniques such as ours.

To finish this thesis, we will present below a list of possible future research topics, which are related to the current study of durational knowledge incorporation. Some of these topics have already been mentioned in this chapter or in earlier technical chapters. Such topics can include aspects at various levels, such as methodological, phonetic, or technical. Some grouping of the topics has been applied:

1. **Direct extensions of the knowledge incorporation paradigm of the current study:**

- Use all techniques and viewpoints in this study to incorporate knowledge about other long-term speech features (e.g., the prosody);

2. **Refined technical implementations:**

- Perform the state-specific linear transformation (as in Chapter 3) on the basis of segmentation (at the state level) of the speech data during Viterbi training;
- Investigate a more integrated combination of the linear discrimination analysis (LDA) at the state level with a state-specific transformation, both theoretically and experimentally (following, e.g., Sun, 1996);
- Proceed with the investigation of the acoustics-related durational pdf (Chapter 4), with the goal to relate the duration measure with a segmental acoustic measure, and further implement this in training and recognition algorithms;
- Use duration mean, variance, and minimal duration of each phone to constrain the training of monophone HMMs with topologies including skip transitions (thus probably more advanced statistical techniques are required), in order to obtain better CI duration modelling;
- Make a similar fully-nested-ANOVA with the four factors used in recognition of Chapter 7, i.e., *Stress, Rate, Location within word* and *Location within utterance,* plus either a *Phone class* or individual *Phones*, in order to verify the problem of insufficient data in the current study, by comparing the resulting percentages with those of the eleven factors;
- Make better CD duration models by, e.g., leaving the duration models empty for those factor cells with less than three instances, in order to prevent the very narrow durational pdf's which might lead to unwanted effects in the *N*-best re-scoring process (suggested by R. Moore, 1996);
- Collect CD durational statistics, both for the duration analysis (Chapter 6) and for the duration models (Chapter 7), but this time not based on the hand labelling provided by TIMIT, but based on a Viterbi back-tracking on the phone sequences of the known utterances (in the segmentation mode). The justification for this is that the hand labels may not be the optimal ones as seen by the HMM recogniser in the process of recognition;

- Further optimise the re-scoring mechanisms in using duration scores;
- While still keeping, e.g., the four factors in the CD duration models, concentrate on the effect of each individual factor by merging the cells among the other three factors;
- The approach of word-juncture modelling in Chapter 7 can be further refined in order to improve the generality of the models extracted from one data set to other data sets;
- Ways should be found to calculate the amount of interaction between the nested factors (Chapter 6);

3. Interesting approaches inspired from literature but not yet investigated in the current study:

- Use HSMM and further constrain its duration mean and variance (and minimal duration) at the (phone) model level;
- Use triphone HMMs (of somehow tied parameters to reduce the total number of parameters) to repeat the two main approaches in this study (Chapter 5 for CI and Chapter 7 for CD duration), to see how much effect of duration modelling still remains;
- Include mechanisms of speaker adaptation (for multi-speaker databases such as TIMIT), and build duration models also based on sub-divisions of speaker groups;

4. Speech databases and their usage:

- Find or produce speech databases that are more suitable for investigating isolated factors, extract the durational regularities based on the same set of factors as used in this study to compare the similarities, and build duration models for recognition tests;
- Use the presently unused two fixed `sa` utterances per speaker in TIMIT for various tests. The fact that this subset of sentences mainly contains speaker variation should be reflected in a different distribution of factor variations (suggested by J. Bernstein, 1996);

This list could easily be extended further. Given the insight obtained during this study, as well as given the challenges in the current ASR research and the potential improvement based on our experience, it is certainly worthwhile to proceed along these lines of research.

Summary

Automatic speech recognition (ASR) is a method for recognising spoken messages by computers. In the present-day state-of-the-art ASR, there are two competing approaches. The statistical approach based on HMM (hidden Markov model) currently outperforms the rule-based knowledge-engineering approach. Various attempts to combine the two approaches also exist. The research presented in this thesis took a viewpoint that in both of these approaches knowledge about speech is used, but this knowledge is represented in different ways. Our way to combine the two approaches is an attempt to incorporate specific knowledge into the HMM-based statistical ASR system. The technically feasible methods of knowledge incorporation were sought out in this thesis work, both based on the structure of HMM-based recogniser, and based on the complicated duration regularities in speech data.

In Chapter 1, first of all the current state-of-the-art in ASR is reviewed, leading to the conclusion that technical improvements are still necessary and possible for ASR. In our view the history of ASR development can be considered as a gradual process of incorporating specific knowledge about speech into the recognisers. Therefore each improvement generally implies the incorporation of a specific piece of knowledge. The current study concentrates on the knowledge about durational behaviour of the phonetic segments (phones), for reasons that there is a rich body of literature about this knowledge, and that the currently most successful HMM techniques have not appropriately incorporated this knowledge. Having chosen the HMM as the basic recogniser structure for this study, the problem of incorporating durational knowledge has two sides, namely on the one hand the durational behaviour of the HMM, and on the other hand the durational behaviour of the phonetic segments themselves as observed in the actual speech database. We were first of all confronted with the *linkage* problem between these two aspects, namely that there is no appropriate *representation form* of this knowledge, that can be used both to collect the knowledge from the database, and to incorporate the knowledge into the HMM-recogniser. This defines the general paradigm of the current study as a methodological one: searching for appropriate representations and searching for feasible ways of incorporation. Other technical specifications for the whole thesis work are also represented in this chapter, such as the use of monophone HMMs (for a manageable complexity and a tangible effect of duration modelling), and the (main) use of the TIMIT multi-speaker continuous-speech database (this database being well documented and close to the situation of continuous speech recognition).

It was decided to incorporate knowledge about context independent (CI) and context dependent (CD) durations separately. The CI durational

behaviour was investigated first. In order to find the relationship between the CI durational behaviour of the HMMs and the CI distribution of the segments, the mathematical basics of HMM are briefly reviewed in Chapter 2, together with a simple durational distribution for the case of a single state of HMM. General technical specifications for ASR research are reviewed, and the basic setups of the recogniser used in the current study are presented.

Based on the type of research of the current study, that was also defined in Chapter 1 as being technical, all different effects of knowledge incorporation should be tested in terms of the performance of a recogniser. Therefore a baseline system had to be built and optimised before any extra durational knowledge is incorporated. In Chapter 3 the optimisation was achieved by linear transformations of the front-end vectors to remove the correlation in them. Both filterbank parameters and mel-scale cepstral coefficients (MFCC) were tested, but ultimately MFCC plus their time-derivatives were chosen for the baseline system. Both a discrete- and a continuous-density system were tested, but only the latter was used in the rest of the thesis. The transformation on MFCC was performed using either the vectors in the whole database (global) or only the vectors assigned to an HMM state (state-specific). Slightly different impacts for phone and word recognition on the baseline performance with different transformation schemes were observed. These results are used for different purposes in the later chapters. The various implementations of linear transformations clarified the limitation of this technique in terms of its capability to improve the performance. This limitation exists mainly because the linear transformations that we used only removed the correlation, and do not improve significantly the modelling accuracy for non-Gaussian speech data in general.

Chapter 4 serves as a theoretical preparation for Chapter 5. In Chapter 4, the durational behaviour of the general left-to-right HMM is analysed. It is shown, with the help of theoretical durational pdf (probability density function), that even a linear HMM, as the simplest special case of a left-to-right model, is rich enough for modelling the single-peak binomial-like durational distributions of most phones. Therefore it is unnecessary to introduce for instance hidden semi-Markov model (HSMM) to repair the durational behaviour at the state level. Relations between the parameters of linear HMMs and the two lower statistics (duration mean and variance) of the phone segments were obtained, in order to be used in Chapter 5. The same relations for HMMs with skip transitions were also derived, but their complexity prevented their use in later chapters.

In Chapter 5, attempts were made to incorporate the CI durational knowledge (in the form of the CI duration mean and variance) into the HMMs. Several paradigms of training procedures were reviewed, including the one for HSMM, and the one for the standard HMM used by us with extra

constraints on segmental durational statistics. The improved training procedure was embedded in the standard Baum-Welch maximum-likelihood (ML) framework. The durationally constrained ML equations were only solved numerically, giving duration fit for most of the phone HMMs in the system. This set of HMMs lead to better segmentation scores, indicating a better duration modelling accuracy. However, no (systematic) improvement in phone or word recognition was achieved.

CI durational modelling was thus considered to be insufficient, both because it did not lead to much improvement in system performance, and because the actual durational distribution is not context independent for sure. The influence of various contextual factors on phone duration was systematically analysed in Chapter 6, to obtain context dependent (CD) durational knowledge. Durational distributions influenced by individual factors, as well as a nested ANOVA including all the 11 chosen factors, were used to reveal the CD durational behaviour. A number of factors appeared to be significant in influencing vowel duration, such as word stress, syllable locations within words and within utterances, and speaking rate, which would be used in the duration models for recognition in Chapter 7. The factor of voicing of post-vocalic stops did not show a systematic effect on the duration of the preceding vowels, thus this was not used in the next chapter.

In Chapter 7, four of the 11 contextual factors were chosen to be incorporated in the recogniser by means of a duration model that is external to the HMMs. Still monophone HMMs were used to generate the first N "N-best" sentence transcriptions for each utterance, at the word level. These word transcriptions were further used to generate phone-level transcriptions using the norm lexical pronunciation plus a word-juncture model. This model was derived from the same database and describes the pronunciation deviations from the norm at word junctures. The phone instances, with their duration and contexts identified, resulted in a phone duration score based on the duration model. The phone duration scores were integrated into the utterance duration score, and this was combined with the already available acoustic score of the N-best sentence transcription. The transcription with the highest combined score was taken as the new top-best. The word correct score of the new top-best transcriptions was marginally higher than the original top-best without this "re-scoring" process. In other words, the CD durational knowledge was incorporated into the recogniser in the post-processing phase.

The whole development of the current study indicates a possibility to incorporate statistically formalised knowledge on duration into a statistical recognition system based on a given structure of HMM. However the structure of this knowledge is defined by a phonetic parameter, being segmental duration. CI and CD types of durational knowledge were incorporated in different ways. The experience of the current study can be useful for incorporation of other long-term speech parameters (such as the

pitch contour) into the frame-based HMM recognisers, which so far had been a difficult problem. Since the overall paradigm of the whole research project is new, a viewpoint on knowledge incorporation into machines was presented in Chapter 8, derived from our specific experience of incorporating durational knowledge, as a contribution to ASR research in general. This viewpoint takes an iterative process of knowledge representation in the form of "structure-plus-parameters".

The current study revealed that incorporating durational knowledge into HMM-based ASR is useful. To achieve this, however, in-depth analyses are required both on the structures presented in the speech data related to segmental durational behaviour, and on the structures of the given (HMM-based) recognition system in modelling various aspects including duration. Furthermore, the improvement of recognition performance will have to rely on careful engineering, both accomplished within this thesis work, and in possible future studies.

Samenvatting

Automatische spraakherkenning (ASH) is een methode om gesproken teksten te herkennen met een computer. Er zijn twee concurrende benaderingen in de moderne ASH. De statistische benadering, gebaseerd op HMM (hidden Markov model), presteert op dit moment beter dan de regel-gebaseerde kennis benadering. Er bestaan ook diverse methodes om de twee benaderingen te combineren. Het in dit proefschrift gepresenteerde onderzoek neemt als uitgangspunt dat kennis over spraak in beide benaderingen gebruikt wordt, maar dat die kennis op verschillende manieren gerepresenteerd is. Onze manier om de twee benaderingen te combineren is die van het toevoegen van specifieke kennis aan het HMM-gebaseerde statistische ASH-systeem. Diverse technische mogelijkheden om kennis te incorporeren zijn in dit promotieproject onderzocht, zowel gebaseerd op de structuur van de HMM-herkenner, alsook op de ingewikkelde duursystematiek van de spraakdata zelf.

In hoofdstuk 1 wordt allereerst de huidige stand van zaken in ASH geëvalueerd; dit leidt tot de conclusie dat technische verbeteringen nog steeds mogelijk en noodzakelijk zijn. Naar onze mening kan de geschiedenis van de ASH-ontwikkeling gezien worden als een geleidelijk proces van het toevoegen van specifieke kennis over gesproken taal aan de herkenner. Iedere verbetering wordt dan ook over het algemeen gerealiseerd door het toevoegen van weer een stukje specifieke kennis. Dit onderzoekproject concentreert zich op de kennis over het duurgedrag van fonetische segmenten (fonemen), enerzijds omdat er een rijke literatuur bestaat over dit onderwerp, en anderzijds omdat in de huidige meest succesvolle HMM-technieken deze kennis meestal nog niet is geïncorporeerd. Wanneer eenmaal voor HMM is gekozen als basisstructuur voor de herkenning heeft het incorporeren van duurkennis nog steeds twee kanten, enerzijds het duurgedrag van de HMM zelf, en anderzijds het duurgedrag van de fonetische segmenten zoals waargenomen in het geselecteerde spraakdatabestand. Wij liepen allereerst tegen het probleem op om deze twee aspecten te verbinden; er is namelijk geen geschikte representatievorm van deze kennis die zowel gebruikt kan worden om kennis te verzamelen vanuit het spraakdatabestand, alsook om die kennis te incorporeren in de HMM-herkenner. Dit maakt dat het algemeen paradigma van dit onderzoek methodologisch van aard is: zowel het zoeken naar geschikte representaties alsook het zoeken naar geschikte vormen van implementatie. Ook andere technische aspecten van dit promotie-onderzoek, zoals het gebruik van monofoon-HMM's (om de complexiteit van het model in de hand te houden en om duidelijke effecten van duurmodellering mogelijk te maken), en het gebruik van het TIMIT spraakdatabestand, worden in dit hoofdstuk toegelicht. Dit Amerikaans-

Engelse TIMIT spraakdatabestand, dat continue spraak van veel verschillende sprekers bevat, is goed gedocumenteerd en benadert de situatie van het herkennen van continue spraak.

Er werd besloten om de context-onafhankelijke (CO) en de context-afhankelijke (CA) duurkennis apart te incorporeren, waarbij het CO-duurgedrag allereerst is onderzocht. Teneinde de relatie te vinden tussen het CO-duurgedrag van de HMM's en de CO-duurdistributie van de segmenten, wordt in hoofdstuk 2 de wiskundige basis van HMM in het kort gepresenteerd. Als voorbeeld wordt de eenvoudige duurdistributie van een HMM met slechts één toestand gegeven. Algemene technische specificaties van ASH-onderzoek passeren de revue, terwijl tevens de basisconfiguratie wordt gepresenteerd van de herkenner die in dit onderzoek wordt gebruikt.

Aangezien dit type onderzoek in hoofdstuk 1 als technisch is gekwalificeerd, dienen ook alle verschillende effecten van kennistoevoeging te worden getoetst in termen van verbeterde prestatie van de herkenner. Daartoe werd een basissysteem geconfigureerd en verder verbeterd alvorens additionele duurkennis toe te voegen. Deze optimalisatie werd in hoofdstuk 3 bewerkstelligd door de analysevectoren lineaire te transformeren teneinde de correlatie daarin te verwijderen. Zowel filterbankparameters als op mel-schaal gebaseerde cepstrale parameters (MFCC) werden getest, maar uiteindelijk werden MFCC's, plus hun afgeleiden in de tijd, gekozen voor het basissysteem. Zowel een systeem met een discrete distributie als een met een continue distributie werd getest, maar alleen het tweede systeem werd in de rest van het onderzoek gebruikt. De transformaties op MFCC werden ofwel gedaan met alle vectoren van het hele databestand (globaal), of alleen met de vectoren behorend bij een specifieke HMM-toestand (toestand-specifiek). Deze beide transformatiesystemen lieten slechts kleine verschillen zien ten aanzien van foneem- en woordherkenning. Deze resultaten zijn van belang voor de verschillende in latere hoofdstukken te behandelen aspecten. De verschillende implementaties van deze lineaire transformaties geven een beter beeld van de grenzen van deze technieken als het gaat om het verbeteren van de herkenningsprestaties. Deze beperking ontstaat vooral omdat de gebruikte lineaire transformaties weliswaar de correlatie verwijderen, maar daarmee nog niet noodzakelijk de nauwkeurigheid van het modelleren van niet-Gaussische spraakdata verbeteren.

Hoofdstuk 4 functioneert als een theoretische voorbereiding op hoofdstuk 5. In hoofdstuk 4 wordt het duurgedrag van de zogenaamde algemene links-naar-rechts HMM onderzocht. Met behulp van de theoretische kansdichtheidsfunctie wordt aangetoond dat zelfs het eenvoudigste lineaire model al krachtig genoeg is om de binomiaal-achtige duurdistributies met een enkele piek, die karakteristiek zijn voor de meeste fonemen, te modelleren. Het is dan ook onnodig om bijvoorbeeld het hidden semi-Markov model (HSMM) te introduceren teneinde het duurgedrag van een toestand in het

Markov-model te repareren. Relaties tussen enerzijds de parameters van lineaire HMM's en anderzijds de gemiddelde duur en de variantie van de foneemsegmenten worden afgeleid teneinde die in hoofdstuk 5 te kunnen gebruiken. Soortgelijke relaties werden afgeleid voor HMM's met zogenaamde skiptransities, maar die waren ongeschikt voor verder gebruik vanwege een te grote complexiteit.

In hoofdstuk 5 wordt geprobeerd de CO-duurkennis (in de vorm van gemiddelde en variantie van de CO-foneemduur) aan de HMM's toe te voegen. Verschillende paradigma's voor trainingsprocedures passeren de revue, zoals die voor HSMM alsook die voor de door ons gebruikte standaard HMM inclusief extra restricties ten aanzien van segmentele duurstatistiek. De verbeterde trainingsprocedure wordt ingebouwd in het zogenaamde standaard Baum-Welch maximum-likelihood (ML) kader. De ML-vergelijkingen met duurrestricties konden alleen numeriek worden opgelost, en leidden tot duuraanpassingen van de meeste foon-HMM's in het systeem. Deze set van HMM's resulteerde in betere segmentatiescores, hetgeen een indicatie is van de grote nauwkeurigheid van deze duurmodellering. Er werd echter geen (systematische) verbetering in de foon- en woordherkenning bereikt.

We moeten dus concluderen dat context-onafhankelijke duurmodellering onvoldoende is, zowel omdat deze niet tot veel verbetering leidt in de prestaties van het systeem, alsook omdat we eigenlijk wel weten dat de werkelijke duurdistributies zeker niet context-onafhankelijk zijn. In hoofdstuk 6 wordt de invloed van verschillende contextuele factoren systematisch onderzocht, teneinde context-afhankelijke (CA) duurkennis te verwerven. Zowel duurdistributies die slechts beïnvloed worden door steeds één enkele factor, alsook een geneste ANOVA met alle elf geselecteerde factoren, werden aangewend om het CA-duurgedrag te ontsluiten. Een aantal factoren, zoals woordklemtoon, plaats van de lettergreep in het woord en in de zin, en spreektempo, bleek ook werkelijk van invloed te zijn op de klinkerduur. Deze factoren zullen dan ook in hoofdstuk 7 worden gebruikt om te trachten duurmodellen aan te wenden voor betere herkenning. In onze data varieerde de klinkerduur niet systematisch als gevolg van het stemhebbend of stemloos zijn van de daarop volgende plofklank; deze factor wordt dan ook in het volgende hoofdstuk niet gebruikt.

In hoofdstuk 7 worden vier van de elf contextuele factoren daadwerkelijk geïmplementeerd in de herkenner in de vorm van duurmodellen die extern zijn aan de HMM's. Nog steeds worden monofoon-HMM's gebruikt om de eerste N beste zinstranscripties op woordniveau per gesproken uiting te genereren. Deze woordtranscripties dienen vervolgens als basis om de foontranscripties te verwerven, met gebruikmaking van zowel een normatief uitspraaklexicon alsook van een model op woordgrensniveau. Dit woordgrensmodel wordt opgebouwd vanuit hetzelfde databestand en

beschrijft de afwijkingen ten opzichte van de normatieve uitspraak op de woordgrenzen. Iedere foonrealisatie, tezamen met zijn duur en locale context, resulteert in een foon-duurscore gebaseerd op het duurmodel. Alle foon-duurscores worden samengevoegd tot één duurscore op het niveau van de uiting, en deze score wordt gecombineerd met de al beschikbare akoestische score van de *N*-best zinstranscriptie. De transcriptie met de dan hoogste gecombineerde score wordt gekozen als de nieuwe topbeste. De correcte woordscore van deze nieuwe topbeste transcripties was iets beter dan die van de oorspronkelijke topbesten, verkregen zonder dit "herscorings" proces. Met andere woorden, enige CA-duurkennis is toegevoegd aan de herkenner in deze post-processing fase.

De hele ontwikkeling van dit onderzoekproject illustreert een mogelijkheid om statistisch geformaliseerde kennis over segmentduren toe te voegen aan een statistisch herkenningssysteem dat gebaseerd is op een bepaalde HMM-structuur. De structuur van deze kennis was echter gedefinieerd door een fonetische parameter, namelijk segmentele duur. CO-en CA-kennis werden op verschillende manieren toegevoegd. De ervaringen van dit onderzoekproject kunnen gebruikt worden om ook andere supra-segmentele spraakparameters (zoals een toonhoogtecontour) aan een frame-gebaseerde HMM-herkenner toe te voegen, hetgeen tot nu toe een moeilijk probleem was. Omdat het in dit onderzoekproject gehanteerde globale paradigma nieuw is, wordt in hoofdstuk 8 aandacht besteed aan de manier waarop kennis kan worden toegevoegd aan een systeem. Deze algemene visie is afgeleid van onze specifieke ervaringen met het toevoegen van duurkennis als een bijdrage aan het ASH-onderzoek in het algmeen. Deze visie kan men omschrijven als een iteratief proces van kennisrepresentatie in de vorm van "structuur-plus-parameters".

Dit onderzoek liet zien dat het toevoegen van duurkennis aan HMM-gebaseerde ASH nuttig kan zijn. Teneinde dit te bereiken zijn echter diepgaande analyses nodig, zowel met betrekking tot de structuren die aanwezig zijn in de spraakdata ten aanzien van segmentduurgedrag, alsook met betrekking tot de structuren van het betreffende (HMM-gebaseerde) herkenningssysteem waar het modelleren van diverse duuraspecten betreft. Bovendien zal het verbeteren van de herkenningsprestaties gebaseerd moeten zijn op zorgvuldige technieken, zoals wij die in dit onderzoek hebben toegepast, en ook in verder onderzoek hopen te kunnen toepassen.

在隐式马尔可夫模型（HMM）为基础的连续语音识别技术中增加关于音素时长的知识

博士论文摘要　　作者　王雪

目前自动语音识别主要采用两种技术，即以HMM为基础的统计方法，和以规则集为基础的人工智能方法。在这两种方法中，通常前者比后者具有较高的识别率。也可以两种方法结合使用。本文认为无论使用哪种方法，尽管在执行中表达方式不同，但都运用了语音的知识。为提高系统的识别率，可在HMM识别系统中人为增加具体选定的知识。本文探讨了技术上可行的增加知识的方法，HMM识别器的内部结构，及语音信号中重要的时长规律。

第一章综述了自动语音识别技术的现状，并指出进一步改善的必要性和可能性。我们可以将自动语音识别技术的整个发展过程看成是逐步向识别器内增加语音学知识的过程，每前进一步增加一特定知识。由于目前一般的HMM识别技术中没有很好地加入有关语音音素时长的知识，而语音学有关这方面的文献又说明了它的重要性，因此，选定了语音音素的时长特性对HMM识别器的影响作为本文的研究方向。该研究方向包含两个方面：一是HMM识别器本身的时长特性，二是从语音库中查到的语音音素的时长特性。目前对时长的知识还没有提出一个合适的表达方式，它既可以用来人为地从语音库中抽取音素时长知识，又可以用来将此知识加到HMM识别器中。本文重点研究上述两方面的问题，同时给出这一研究中的其它一些技术规范，如使用单音HMM（旨在限定复杂程度及突出时长效应）及使用多说话人的连续语音库ＴＩＭＩＴ。

音素时长特征本身又有邻域无关（ＣＩ）及邻域相关（ＣＤ）之分。我们首先探讨ＣＩ时长。为得到HMM的ＣＩ时长特性与音素片段的ＣＩ时长分布间的关系，第二章简要地综述了必要的HMM数学基础，及单一HMM状态的时长分布，进而描述了语音识别领域里常用的技术判据及本项研究中使用的识别器。

向识别器中加入各种语音学知识产生的效果，需通过向识别器加入各种语音学知识前后的运行情况来鉴定，因此在增加任何时长知识之前，应该对基本语音识别器进行优化。第三章中讨论了对识别器前沿处理参数的优化问题。首先采用对滤波器阵及倒谱系数（ＭＦＣＣ）两种参数进行线性变换来优化识别器（本文后继部分中仅采用ＭＦＣＣ及其动态参数）。此种线性变换方法分别基于全部语音库的ＭＦＣＣ及每一个HMM状态的ＭＦＣＣ。这种优化方法分别用于离散及连续观察密度的识别器中。实验结果表明：这些不同的线性变换分别使识别器对单一音素及对整个词的识别有些微小的提高。这些变换将用于后继章节的不同目的。上述实验结果说明这一优化技术对于改善识别器的效果是有限的。其原因是由于线性变换仅去除参数间的相关性，而这对于一般的非正态多维语音参数来说，模拟精度并没有显著提高。

第四章为第五章做理论准备。第四章首先分析了一般左至右转移结构HMM的时长特性。结果表明：既使最简单的单路结构HMM已能胜任一般音素单峰式的二项概律分布之良好模拟。这里得出了单路HMM模型参数与时长之期望值及方差间的关系，并将用于第五章。虽然也得出了包含跨越转移的HMM之类似关系，但因过于复杂而不将使用于后继章节的技术中。

第五章试图将ＣＩ时长知识加到HMM中。首先对各种训练步骤进行了比较，包括将采用的在标准步骤中增加时长约束的方法。这个改进的方法仍然基于Baum-Welch最大似然法（ＭＬ）之框架。带时长约束的ＭＬ方程组仅以数值方法解出，这使系统中大多数单音HMM获得时长逼近。这一组HMM导致改善的自动切分音素之运行，显示了对时长模拟精度之改善。但这没有改善对音素及单词的识别率。

由于ＣＩ时长模拟没有显著改善系统性能，也由于实际的时长分布并非与邻域无关，仅作ＣＩ时长模拟显然是不够的。第六章系统地分析了各种邻域因素对时长之影响，从而获得ＣＤ时长知识。分别进行了单一因素对时长之影响的分析，以及一个包括十一个因素的嵌套式方差分析ＡＮＯＶＡ。有些因素对时长有显著影响，例如重读音，音节在词及句中位置，以及说话速率。这些因素将用于第七章识别器的时长模型中。

第七章研究采用由十一个因素中之四个因素所建立的时长模型的识别。由于元音后接破擦音之清浊度对元音时长没有系统地影响，故本章未采用这一因素。首先采用单音HMM得到数个第一步假设单词序列；再根据各单词的辞典发音及一个词隙模型，从单词序列中获取假设音素序列。（这个词隙模型亦是从同一个语音库中提取的，它描述词与词转接时的发音变化。）每一音素在获得了全部邻域标识之后，即由时长模型得到一个时长计分值。从这一计分值累加得到全句的时长计分值，再与第一步假设之声学计分值相结合。具有最高总计分值之单词序列即为最终识别序列。换言之，ＣＤ时长知识是在后处理步骤中加入。这一最终识别序列的单词正确识别率比不用时长模型有一定的提高。

整个研究显示了将统计式知识加到由HMM确定结构的统计式系统中是可行的。然而这一知识的结构是由语音学参数确定的。ＣＩ及ＣＤ时长知识是由不同方法加入的。本研究之经验对在短时谱为基础的HMM识别器中加入其他长时语音参数亦会有用，这些至今仍为困难课题。第八章给出由增加时长知识得到的经验总结，即向一般自动机中增加一般知识的路径。这路径是“结构加参数”的知识表达方法的一种迭代过程。

实验结果表明：向HMM语音识别器中增加时长知识是有用的。但为了做到这一点，要求深入研究语音信号中有关时长的信息结构，以及HMM识别器有关时长的特性。对识别器的改善亦有赖于认真的工程手段，这分别报道于本论文中及有待后继工作。

References

Adlersberg, S. & Cuperman, V. (1987): "Transform domain vector quantization for speech signals", *Proceedings ICASSP'87,* Dallas, TX, 1938-1941.

Allen, J.B. (1994): "How do humans process and recognize speech?", *IEEE Trans. Speech Audio Processing* **2**(4), 567-577.

Anastasakos, A., Schwartz, R. & Shu, H. (1995): "Duration modeling in large vocabulary speech recognition", *Proceedings ICASSP'95,* Detroit, Michigan, 628-631.

Aubert, X. & Ney, H. (1995): "Large vocabulary continuous speech recognition using word graphs", *Proceedings ICASSP'95,* Detroit, Michigan, 49-52.

Austin, S., Schwartz, R. & Placeway, P. (1991): "The forward-backward search algorithm", *Proceedings ICASSP'91*, Toronto, Canada, 697-700.

Bahl, L.R., Jelinet, F. & Mercer, R.L. (1983): "A maximum likelihood approach to continuous speech recognition", *IEEE Trans. Pattern Anal. Machine Intel.* **5**(2), 179-190.

Bartkova, K., Jouvet, D. & Moudenc, T. (1995): "Using segmental duration prediction for rescoring the N-best solution in speech recognition", *Proceedings ICPhS'95,* Stockholm, Sweden, 248-251.

Baum, L.E., Petrie, T., Soules, G. & Weiss, N. (1970): "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains", *The annals of mathematical statistics* **41**(1), 164-171.

Bernstein, J. (1996): "Use of the sa utterances in test", persoanl communication at *ICSLP'96*, Philadelphia, PA.

Bocchieri, E.L. & Wilpon, J.G. (1993): "Discriminative feature selection for speech recognition", *Computer Speech and Language* **7,** 229-246**.**

Bourlard, H., Hermansky, H. & Morgan, N. (1996), "Towards increasing speech recognition error rates", *Speech Comm*. **18**, 205-231.

Bourlard, H. & Wellekens, C. (1990): "Links between Markov models and multilayer perceptrons", *IEEE Trans. Pattern Anal. Machine Intel.* **12**(12), 1167-1178.

Boves, L., Landsbergen, J., Scha, R. & van Noord, G. (1995): *NWO Priority Programme: Language and speech technology*.

Bronkhorst, A.W., Bosman, A.J. & Smoorenburg, G.F. (1993): "A model for context effects in speech recognition", *J. Acoust. Soc. Am.* **93**(1), 499-509.

Brugnara, F., Falavigna, D. & Omologo, M. (1993): "Automatic segmentation and labeling of speech based on hidden Markov models", *Speech Comm.* **12**, 357-370.

Burshtein, D. (1995): "Robust parametric modeling of durations in hidden Markov models", *Proceedings ICASSP'95,* Detroit, Michigan, 548-551.

Chollet, G. (1995): "Evaluation of ASR systems, algorithms and databases", in: Rubio Ayuso, A.J. & Lopez Soler, J.M. (eds): *NATO ASI Series, Vol. F147, Speech recognition and coding: New advances and trends,* Springer Verlag, Berlin, 32-40.

Cole, R., Hirschman, L., Atlas, L., Beckman, M., Biermann, A., Bush, M., Clements, M., Cohen, J., Garcia, O., Hanson, B., Hermansky, H., Levinson, S., McKeown, K., Morgan, N., Novick, D.G., Ostendorf, M., Oviatt, S., Price, P., Silverman, H., Spitz, J., Waibel, A., Weinstein, C., Zahorian, S. & Zue, V. (1995): "The challenge of spoken language systems: Research directions for the Nineties", *IEEE Trans. Speech Audio Processing* **3**(1), 1-21.

Cosi, P., Falavigna, D. & Omologo, M. (1991): "A preliminary statistical evaluation of manual and automatic segmentation discrepancies", *Proceedings EUROSPEECH'91*, Genova, Italy, 693-696.

Cooley, W.W. & Lohnes, P.R. (1971): *Multivariate Data Analysis*, John Wiley, New York.

Crystal, T.H. & House, A.S. (1988a), "Segmental durations in connected-speech signals: Current results", *J. Acoust. Soc. Amer*. **83**, 1553-1573.

Crystal, T.H. & House, A.S. (1988b): "Segmental durations in connected-speech signals: Syllabic stress", *J. Acoust. Soc. Amer*. **83**, 1574-1585.

Dai, J., MacKenzie, I.G. & Tyler, J.E.M. (1994): "Stochastic modeling of temporal information in speech for hidden Markov models", *IEEE Trans. Speech Audio Processing* **2**(1), 102-104.

Davis, K.H., Biddulph, R. & Balashek, S. (1952): "Automatic recognition of spoken digits", *J. Acoust. Soc. Amer.* **24**(6), 637-642.

Davis, S.B. & Mermelstein, P. (1980): "Comparison of parametric representations for monosyllabic word recognition in continuous spoken sentences", *IEEE Trans. Acoust., Speech, Signal Processing* **28**(4), 357-366.

Dempster, A.P., Laird, N.M. & Rubin, D.B. (1977): "Maximum likelihood from incomplete data via the EM algorithm", *J. Royal Statistical Society* **39**, 1-38.

Deng, L. (1992): "A generalized hidden Markov model with state-conditioned trend functions of time for the speech signal", *Signal Processing* (Elsevier) **27**, 65-78.

Deng, L. (1996): "Production models as a structural basis for automatic speech recognition", *Proceedings 1st ESCA tutorial and research workshop on speech production models-4th speech production seminar*, Autrans (France), May, 1996, 69-80.

Deng, L., Kenny, P., Lennig. M. & Mermelstein, P. (1992): "Modeling acoustic transitions in speech by state-interpolation hidden Markov models", *IEEE Trans. Signal Processing* **40**(2), 265-271.

Deng, L. & Braam, D. (1994): "Context-dependent Markov model structured by locus equations: Applications to phonetic classification", *J. Acoust. Soc. Am.* **96**(4), 2008-2025.

Doddington, G.R. (1989): "Phonetically sensitive discriminants for improved speech recognition", *Proceedings ICASSP'89*, Glasgow, Scotland, 556-559.

Dumouchel, P. & O'shaughnessy, D. (1995): "Segmental duration and HMM modeling", *Proceedings EUROSPEECH'95*, Madrid, Spain, 803-806.

Dunn, O.J. & Clark, V.A. (1987): *Applied statistics: Analysis of variance and regression*, John Wiley & Sons, 103-122.

Ephraim, Y., Dembo, A. & Rabiner, L.R. (1989): "A minimum discrimination information approach for hidden Markov modeling", *IEEE Trans. Inform. Theory* **35**(5), 1001-1013.

Ephraim, Y. & Rabiner, L.R. (1990): "On the relations between modeling approaches for speech recognition", *IEEE Trans. Inform. Theory* **36**(2), 372-380.

Furui, S. (1989): *Digital speech processing, synthesis, and recognition,* M. Dekker, Inc. New York.

Gauvain, J.L., Lamel, L.F., Adda, G. & Adda-Decker, M. (1994), "Speaker-independent continuous speech dictation", *Speech Comm.* 15, 21-37.

Gauvain, J.L. & Lee, C.-H. (1994): "Maximum *a posteriori* estimation for multivariate Gaussian mixture observations of Markov chains", *IEEE Trans. Speech Audio Processing* **2**(2), 291-298.

Gauvain, J.L., Gangolf, J.J. & Lamel, L.F. (1996): "Speech recognition for an information kiosk", *Proceedings ICSLP'96,* Philadelphia, PA, 849-852.

Gersho, A. & Gray, R.M. (1992): *Vector quantization and signal compression,* Kluwer academic publishers, Boston.

Giachin, E.P., Rosenberg, A.E. & Lee, C.-H. (1991): "Word juncture modeling using phonological rules for HMM-based continuous speech recognition", *Computer Speech and Language* **5**, 155-168.

Goldenthal, W.D. & Glass, J.R. (1993): "Modelling spectral dynamics for vowel classification", *Proceedings EUROSPEECH'93*, Berlin, Germany, 289-292.

Gong, Y. & Treurniet, W.C. (1993): "Duration of phones as function of utterance length and its use in automatic speech recognition", *Proceedings EUROSPEECH'93,* Berlin, Germany, 315-318.

Gong, Y., Haton, J.-P. & Mari, J.-F. (1994): "Issues in acoustic modeling of speech for automatic speech recognition", *Proceedings CRIM/FORWISS workshop,* Munchen, Sept. 1994, 34-44.

Guédon, Y. (1992): "Review of Several Stochastic Speech Unit Models", *Computer Speech and Language* **6**, 377-402.

Gupta, V.N., Lennig, M. & Mermelstein, P. (1987): "Integration of acoustic information in a large vocabulary word recognizer", *Proceedings ICASSP'87*, Dallas, TX, 697-700.

Gupta, V.N., Lennig, M., Mermelstein, P., Kenny, P., Seitz, P.F. & O'Shaughnessy, D. (1992): "Use of minimum duration and energy contour for phonemes to improve large vocabulary isolated-word recognition", *Computer Speech and Language* **6**, 345-359.

Haeb-Umbach, R. & Ney, H. (1992): "Linear discriminate analysis for improved large vocabulary continuous speech recognition", *Proceedings ICASSP'92,* San Francisco, CA, I.13-I.16.

Haeb-Umbach, R., Geller, D. & Ney, H. (1993): "Improvements in connected digit recognition using linear discriminate analysis and mixture densities", *Proceedings ICASSP'93,* Minneapolis, Minnesota, II.239-II.242.

Haeb-Umbach, R. & Ney, H. (1994): "Improvements in beam search for 10,000-word continuous-speech recognition", *IEEE Trans. Speech Audio Processing* **2**(2), 353-356.

Hansen, J.H.L. & Bou-Ghazale, S.E. (1995): "Robust speech recognition training via duration and spectral-based stress token generation", *IEEE Trans. Speech Audio Processing* **3**(5), 415-421.

Heuven, V.J. van & Pols, L.C.W. (Eds.) (1993): *Analysis and synthesis of speech. Strategic research towards high-quality text-to-speech generation*, Mouton de Gruyter, Berlin.

Hochberg, M.M. & Silverman, H.F. (1993): "Constraining the duration variance in HMM-based connected-speech recognition", *Proceedings EUROSPEECH '93,* Berlin, Germany, 323-326.

Hochberg, M.M., Cook, G.D., Renals, S.J., Robinson, A.J. & Schechtman, R.S. (1995): "The 1994 ABBOT hybrid connectionist-HMM large-vocabulary recognition system", *Proceedings. ARPA Spoken Language Systems Technology Workshop*, Austin, TX, 170-175.

Huang, X.D., Ariki, Y. & Jack, M.A. (1990): *Hidden Markov models for speech recognition,* Edinburgh university press.

Huang, X.D. & Jack, M.A. (1989): "Semi-continuous hidden Markov models for speech signals", *Computer Speech and Language* **3**, 239-251.

Hwang, M.-Y., Hon, H.-W. & Lee, K.-F. (1989): "Modeling between-word coarticulation in continuous speech recognition", *Proceedings EUROSPEECH'89,* Paris, France, 5-8.

Jankowski, C.R.Jr., Vo, H.-D.H. & Lippmann, R.P. (1995): "A comparison of signal processing front ends for automatic word recognition", *IEEE Trans. Speech Audio Processing* **3**(4), 286-293.

Jelinek, F. (1976): "Continuous speech recognition by statistical methods", *Proceedings IEEE* **64**(4), 532-556.

Jelinek, F. (1996): "Five speculations (and a divertimento) on the themes of H. Boulard, H. Hermansky and N. Morgan", *Speech Comm.* **18**, 242-246.

Jones, M. & Woodland, P.C. (1993): "Using Relative Duration in Large Vocabulary Speech Recognition", *Proceedings EUROSPEECH'93*, Berlin, Germany, 311-314.

Jones, M. & Woodland, P.C. (1994): "Modelling syllable characteristics to improve a large vocabulary continuous speech recogniser", *Proceedings ICSLP'94*, Yokohama, Japan, 2171-2174.

Jouvet, D., Bartkova, K. & Stouff, A. (1994): "Structure of allophonic models and reliable estimation of the contextual parameters", *Proceedings ICSLP'94*, Yokohama, Japan, 283-286.

Juang, B.-H. & Rabiner, L. R. (1990): "The segmental *K*-means algorithm for estimating parameters of hidden Markov models", *IEEE Trans. Acoust., Speech, Signal Processing* **38**(9), 1639-1641.

Juang, B.-H. & Rabiner, L. R. (1992): "Issues in Using Hidden Markov Models for Speech Recognition", in: Furui, S. and Soudhi, M. M. (eds.), *Advances in Speech Signal Processing*, Marcel Dekker, Inc. New York, 509-553.

Juang, B.-H. & Katagiri, S. (1992): "Discriminative training", *J. Acoust. Soc. Jpn. (E)* **13**(6), 333-339.

Juang, B.-H., Chou, W. & Lee, C.-H. (1995): "Statistical and discriminative methods for speech recognition", in: Rubio Ayuso, A.J. & Lopez Soler, J.M. (eds): *NATO ASI Series, Vol. F147, Speech recognition and coding: New advances and trends,* Springer Verlag, Berlin, 41-55.

Kamp, Y. (1991): *An introduction to the Baum and EM algorithms for maximum likelihood estimation,* Rapport no. 830 of Institute for Perception Research, Eindhoven.

Katoh, M. & Kodha, M. (1994): "A study of Viterbi best-first search for isolated word recognition using duration-controlled HMM", *Proceedings ICSLP'94*, Yokohama, Japan, 263-266.

Keating, P.A., Byrd, D., Flemming, E. & Todaka, Y. (1994): "Phonetic analysis of word and segment variation using the TIMIT corpus of American English", *Speech Comm.* **14**, 131-142.

Kenny, P., Parthasarathy, S., Gupta, V.N., Lennig, M., Mermelstein, P. & O'Shaughnessy, D. (1991): "Energy, duration and Markov models", *Proceedings EUROSPEECH'91,* Genova, Italy, 655-658.

Kenny, P., Hollan, R., Gupta, V.N., Lennig, M., Mermelstein, P. & O'Shaughnessy, D. (1993): "A*-admissible heuristics for rapid lexical access", *IEEE Trans. Speech Audio Processing* **1**(1), 49-58.

Klatt, D.H. (1976): "Linguistic uses of segmental duration in English: acoustic and perceptual evidence", *J. Acoust. Soc. Am.* **59**(5), 1208-1221.

Lamel, L.F. & Gauvain, J.-L. (1993a): "Identifying non-linguistic speech features", *Proceedings EUROSPEECH'93,* Berlin, Germany, 23-30.

Lamel, L.F. & Gauvain, J.-L. (1993b): "High performance speaker-independent phone recognition using CDHMM", *Proceedings EUROSPEECH'93,* Berlin, Germany, 121-124.

Le Cerf, P., van Compernolle, D. & van Diest, M. (1992): "Reduction techniques for frames and frame dimensions in automatic speech recognition", *Proceedings EURASIP'92,* Brussels, Belgium, 371-374.

Lee, C.-H. & Gauvain, J.-L. (1995): "Adaptive learning in acoustic and language modeling", in: Rubio Ayuso, A.J. & Lopez Soler, J.M. (eds): *NATO ASI Series, Vol. F147, Speech recognition and coding: New advances and trends,* Springer Verlag, Berlin, 14-31.

Lee, K.-F. (1989): *Automatic speech recognition: the development of the Sphinx system*, Kluwer Academic Publishers, Boston.

Lee, K.-F. & Hon, H.-W. (1989): "Speaker-independent phone recognition using Hidden Markov Models", *IEEE Trans. Acoust., Speech, Signal Processing* **37**, 1641-1648.

Lee, K.-F. & Mahajan, S. (1990): "Corrective and reinforcement learning for speaker-independent continuous speech recognition", *Computer Speech and Language* **4**, 231-245.

Leggetter, C.J. & Woodland, P.C. (1994): "Speaker adaptation of continuous density HMMs using multivariate linear regression", *Proceedings ICSLP'94*, Yokohama, Japan, 451-454.

Levinson, S.E. (1985): "Structural methods in automatic speech recognition", *Proceedings of IEEE* **73**(11), 1625-1650.

Levinson, S.E. (1986): "Continuously variable duration hidden Markov models for automatic speech recognition", *Computer Speech and Language* **1**, 29-45.

Lippmann, R.P. (1987): "An introduction to computing with neural nets", *IEEE ASSP Magazine,* April 1987, 4-22.

Ljolje, A. (1994a): "High accuracy phone recognition using context clustering and quasi-triphone models", *Computer Speech and Language* **8**, 129-151.

Ljolje, A. (1994b): "The importance of cepstral parameter correlation in speech recognition", *Computer Speech and Language* **8**, 223-232.

Ljolje, A. (1995): "Calculation of delta coefficients from MFCC coefficients", persoanl communication.

Ljolje, A. & Riley, M.D. (1991), "Automatic segmentation and labeling of speech", *Proceedings ICASSP 91*, Toronto, Canada, 473-476.

Ljolje, A., Riley, M., Hindle D. & Pereira, F. (1995): "The AT&T 60,000 word speech-to-text system", *Proceedings ARPA Spoken Language Systems Technology Workshop*, Austin, TX, 162-165.

Lloyd, E. (1980): *Handbook of applied mathematics Vol. 2: Probability*, John Wiley & Sons, Ltd.

Magrin-Chagnolleau, I., Bonastre, J.-F. & Bimbot, F. (1995): "Effect of utterance duration and phonetic content on speaker identification using second-order statistical methods", *Proceedings EUROSPEECH'95*, Madrid, Spain, 337-340.

Mantysalo, J., Torkkola, K. & Kohonen, T. (1994): "Mapping context dependent acoustic information into context independent form by LVQ", *Speech Comm.* **14**, 119-130.

Marcus, J.N. & Zue, V.W. (1991): "A variable duration acoustic segment HMM for hard-to-recognise words and phrases", *Proceedings ICASSP '91*, Toronto, Canada, 281-284.

Markowitz, J. (1995): "Talking to machines", *Byte*, 97-104.

Merhav, N. & Ephraim, Y. (1991a): "Maximum likelihood hidden Markov modeling using a dominant sequence of states", *IEEE Trans. Signal Processing* **39**(9), 2111-2115.

Merhav, N. & Ephraim, Y. (1991b): "Hidden Markov modeling using a dominant state sequence with application to speech recognition", *Computer Speech and Language* **5**, 327-339.

Merhav, N. & Lee, C.-H. (1993): "A minimax classification approach with application to robust speech recognition", *IEEE Trans. Speech Audio Processing* **1**(1), 90-100.

Merialdo, B. (1993): "On the locality of the forward-backward algorithm", *IEEE Trans. Speech Audio Processing* **1**(2), 255-257.

Minematsu, N. & Hirose, K. (1994): "Speech recognition using HMM with decreased intra-group variation in the temporal structure", *Proceedings ICSLP'94*, Yokohama, Japan, 187-190.

Mitchell C., Harper, M. & Jamieson, L. (1995) "On the complexity of explicit duration HMM's", *IEEE Trans. Speech Audio Processing* **3**(3), 213-217.

Monkowski, M.D., Picheny, M.A. & Rao, P.S. (1995): "Context dependent phonetic duration models for decoding conversational speech", *Proceedings ICASSP 95*, Detroit, Michigan, 528-531.

Moore, R.K. (1993): "Whither a theory of speech pattern processing?", *Proceedings EUROSPEECH'93*, Berlin, Germany, 43-47.

Moore, R.K. (1994): "Twenty things we still don't know about speech", in: Niemann, H., de Mori, R. & Harrieder, G. (eds.), *Progress and prospects of speech research and technology, Proceedings CRIM/FORWISS workshop, Munchen,* Sept. 1994, 9-17.

Moore, R.K. (1996): "Minimal instances in CD duration models", persoanl communication at *ICSLP'96*, Philadelphia, PA.

Morgan, N. & Bourlard, H. (1995): "Continuous speech recognition: An introduction to the hybrid HMM/connectionist approach", *IEEE Signal Proc. Magazine*, May 1995, 25-42.

Ney, H. & Aubert, X. (1994): "A word graph algorithm for large vocabulary, continuous speech recognition", *Proceedings ICSLP'94*, Yokohama, Japan, 1355-1358.

Nicol, N., Euler, S., Falkhausen, M., Reininger, H., Wolf, D. & Zinke, J. (1992): "Improving the robustness of automatic speech recognizers using state duration information", *Proceedings ESCA Workshop on Speech Processing in Adverse Conditions*, Cannes-Mandelieu (France), Nov. 1992, 183-186.

Niyogi, P. & Zue, V.W. (1991): "Correlation analysis of vowels and their application to speech recognition", *Proceedings EUROSPEECH'91*, Genova, Italy, 1253-1256.

Nooteboom, S.G. (1970): *Production and perception of vowel duration*, Ph.D. Thesis, University of Utrecht.

Oerder, M. & Ney, H. (1993): "Word graphs: An efficient interface between continuous-speech recognition and language understanding", *Proceedings ICASSP'93,* Minneapolis, Minnesota, II.119-II-122.

Ohno, S. & Fujisaki, H. (1995): "A method for quantitative analysis of the local speech rate", *Proceedings EUROSPEECH'95*, Madrid, Spain, 421-424.

Okamoto, M. (1969): "Optimality of principal components", in Krishnaiah, P.R. (ed.): *Multivariate analysis-II,* Academic Press, New York, 673-685.

Oppenheim, A.V. & Johnson, D.H. (1972): "Discrete representation of signals", *Proceedings IEEE* **60**, 681-691.

Osaka, Y., Makino, S. & Sone, T. (1994): "Spoken word recognition using phoneme duration information estimated from speaking rate of input speech", *Proceedings ICSLP'94*, Yokohama, Japan, 191-194.

O'Shaughnessy, D. (1987): *Speech communication: Human and machine,* Addison-Wesley Publishing Company, Reading.

Ostendorf, M. & Roukos, S. (1989): "A stochastic segmental model for phoneme-based continuous speech recognition", *IEEE Trans. Acoust., Speech, Signal Processing* **37**(12), 1857-1869.

Papoulis, A. (1990): *Probability & statistics*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Pallett, D.S., Fiscus, J.G., Fisher, W.M., Garofolo, J.S., Lund, B.A., Martin, A. & Przybocki, M.A. (1995): "1994 Benchmark tests for the ARPA Spoken Language Program", *Proceedings ARPA Spoken Language Systems Technology Workshop*, Austin, TX, 5-36.

Pauws, S., Kamp, Y. & Willems, L. (1994): *On the automatic segmentation of transcribed words*, Rapport no. 1023 of Institute for Perception Research, Eindhoven.

Peterson, G.E. & Lehiste, I. (1960): "Duration of syllable nuclei in English", *J. Acoust. Soc. Am*. **32**, 693-703.

Picone, J. (1989): "On modeling duration in context in speech recognition", *Proceedings ICASSP'89,* Glasgow, Scotland, 421-424.

Pitrelli J.F. (1990): "Hierarchical modeling of phoneme duration: Application to speech recognition", Ph.D. thesis, MIT.

Pitrelli J.F. & Zue, V.W. (1989): "A hierarchical model for phoneme duration in American English", *Proceedings EUROSPEECH'89,* Paris, France, 324-327.

Pols, L.C.W. (1977): *Spectral analysis and identification of Dutch vowels in monosyllabic words,* Ph.D. thesis, Free university Amsterdam.

Pols, L.C.W. (1994): "Speech technology systems: performance and evaluation", in: Asher, R.E. & Simpson, J.M.Y. (eds): *The encyclopedia of language and linguistics*, Pergamon Press, Oxford, **vol. 8,** 4289-4296.

Pols, L.C.W., Wang, X. & ten Bosch, L.F.M. (1996): "Modelling of phone duration (using the TIMIT database) and its potential benefit for ASR", *Speech Comm*. **19**, 161-176.

Press, W.H., Flannery, B.P., Teukolsky, S.A. & Vetterling, W.T. (1989): *Numerical recipes in Pascal*, Cambridge Univ. Press, 305-306.

Rabiner, L.R. (1988): "Mathematical foundations of hidden Markov models", in: Niemann, H., Lang, M. & Sagerer, G. (eds): *NATO ASI Series, Vol. F46: Recent advances in speech understanding and dialog systems,* Springer Verlag, Berlin, 183-205.

Rabiner, L.R. (1989): "A tutorial on hidden Markov models and selected application in speech recognition", *Proceedings IEEE* **77**(2), 257-286.

Rabiner, L.R. & Schafer, R.W. (1978): *Digital processing of speech signals*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Rabiner, L.R. & Juang, B.-H. (1986): "An introduction to hidden Markov models", *IEEE ASSP Magazine*, 4-16.

Rabiner, L.R. & Juang, B.-H. (1993): *Fundamentals of speech recognition*, Prentice-Hall, Inc. Englewood Cliffs.

Riley, M.D. (1992): "Tree-based modelling of segmental durations", in: Bailly, G., Benoit, C. & Sawallis, T.R. (eds.): *Talking machines: Theories, models, and designs,* Elsevier Science publishers B.V., 265-273.

Roe, D.B. & Riley, M.D. (1994): "Prediction of word confusabilities for speech recognition", *Proceedings ICSLP'94*, Yokohama, Japan, 227-230.

Rose, R.C., Schroeter, J. & Sondhi, M.M. (1996): "The potential role of speech production models in automatic speech recognition", *J. Acoust. Soc. Am*. **99**(3), 1699-1709 (and "Critique:" by Moore, R.K., 1710-1713).

Russell, M.J. & Moore, R.K. (1985): "Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition", *Proceedings ICASSP'85*, Tampa, Florida, 5-8.

Sakoe, H. (1992): "Dynamic programming-based speech recognition algorithms", in: Furui, S & Soudhi, M.M. (eds.), *Advances in Speech Signal Processing*, Marcel Dekker, Inc. New York, 487-507.

Scharf, L.L. (1991): *Statistical signal processing: Detection, estimation, and time series analysis*, Addison-Wesley Publishing Company, Reading.

Scheffe, H. (1959): *The analysis of variance,* John Wiley & Sons, New York.

Schwartz, R. & Chow, Y.-L. (1990): "The N-best algorithm: An efficient and exact procedure for finding the N most likely sentence hypotheses", *Proceedings ICASSP'90*, Albuquerque, NM, 81-84.

Schwartz, R. & Austin, S. (1991): "A comparison of several approximate algorithms for finding multiple (N-best) sentence hypotheses", *Proceedings ICASSP'91*, Toronto, Canada, 701-704.

Shih, H-H., Young, S.J. & Waegner, N.P. (1995): "An inference approach to gramar construction", *Computer Speech and Language* **9**, 235-256.

Shirai, K., Okawa, S. & Kobayashi, T. (1992): "Phoneme recognition in continuous speech based on mutual information considering phoneme duration and connectivity", *Proceedings ICSLP'92,* Banff, Canada, 1479-1482.

Siegler, M.A. & Stern, R.M. (1995): "on the effects of speech rate in large vocabulary speech recognition systems", *Proceedings ICASSP'95*, Detroit, Michigan, 612-615.

Singer, H. & Sagayama, S. (1993): "Suprasegmental duration control with matrix parsing in continuous speech recognition", *Speech Comm.* **13**, 315-322.

Siohan, O., Gong, Y. & Haton, J.-P. (1993): "A Bayesian approach to phone duration adaptation for Lombard speech recognition", *Proceedings EUROSPEECH'93,* Berlin, Germany, 1639-1642.

Sitaram, R.N.V. & Sreenivas, T. (1995): "On incorporating phonemic constraints in hidden Markov models for speech recognition", *Proceedings EUROSPEECH'95*, Madrid, Spain, 775-778.

Suaudeau, N. & Andre-Obrecht, R. (1993): "Sound duration modelling and time-variable speaking rate in a speech recognition system", *Proceedings EUROSPEECH'93,* Berlin, Germany, 307-310.

Sun, D.X. (1996): "Feature dimension reduction using reduced-rank maximum likelihood estimation for hidden Markov models", *Proceedings ICSLP'96,* Philadelphia, PA, 244-247.

Sun, D.X. & Deng, L. (1994): "Nonstationary-state hidden Markov model with state-dependent time warping: Application to speech recognition", *Proceedings ICSLP'94*, Yokohama, Japan, 243-246.

Sun, D.X. & Deng, L. (1995): "Analysis of acoustic-phonetic variations in fluent speech using TIMIT", *Proceedings ICASSP'95*, Detroit, Michigan, 201-204.

Sun, D.X., Deng, L. & Wu, C.F.J. (1994): "State-dependent time warping in the trended hidden Markov model", *Signal Processing* (Elsevier) **39**, 263-275.

Svendsen, T. & Kvale, K. (1990): "Automatic alignment of phonemic labels with continuous speech", *Proceedings ICSLP'90*, Kobe, Japan, 997-1000.

Strik, H., Russel, A., van den Heuvel, H., Cucchiarini, C. & Boves, L. (1996): "Localizing an automatic inquiry system for public transport information", *Proceedings ICSLP'96,* Philadelphia, PA, 853-856.

Ten Bosch, L.F.M. (1991): "On relations between phone models, segment duration, and the Padé-expansion", *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam* **15**, 61-77

Umeda, N. (1975): "Vowel duration in American English", *J. Acoust. Soc. Am.* **58**(2), 434-445.

Van Alphen, P. (1992): *HMM-based continuous-speech recognition: Systematic evaluation of various system components,* Ph.D. thesis, University of Amsterdam.

Van Bergem, D. (1995): *Acoustic and lexical vowel reduction*, Ph.D. thesis, IFOTT series "Studies in Language and Language Use" no. 16, University of Amsterdam.

Van Santen, J.P.H. (1992a): "Deriving text-to-speech durations from natural speech", in: Bailly, G., Benoit, C. & Sawallis, T.R. (eds.): *Talking machines: Theories, models, and designs,* Elsevier Science publishers B.V., 275-285.

Van Santen, J.P.H. (1992b): "Contextual Effects on Vowel Duration", *Speech Comm*. **11**(6), 513-546.

Van Santen, J.P.H. (1994): "Assignment of segmental duration in text-to-speech synthesis", *Speech Comm*. **8**, 95-128.

Van Santen, J.P.H. & Olive, J.P. (1990): "The analysis of contextual effects on segmental duration", *Computer Speech and Language* **4**, 359-390.

Van Son, R.J.J.H. & Pols, L.C.W. (1990): "Formant frequencies of Dutch vowels in a text, read at normal and fast rate", *J. Acoust. Soc. Am.* **88**(4), 1683-1693.

Vaseghi, S.V. & Conner, P. (1992): "Hidden Markov models with combined state transition and duration probabilities", *Proceedings EURASIP'92,* Brussels, Belgium, 435-438.

Vorstermans, A, Martens, J.P. & van Coile, B. (1995): "Fast automatic segmentation and labeling: Results on TIMIT and EUROM0", *Proceedings EUROSPEECH'95*, Madrid, Spain, 1397-1400.

Vorstermans, A, Martens, J.P. & van Coile, B. (1996): "Automatic segmentation and labelling of multi-lingual speech data", *Speech Comm*. **19**, 271-293.

Wakita, Y. & Tsuboka, E. (1994): "State duration constraint using syllable duration for speech recognition", *Proceedings ICSLP'94*, Yokohama, Japan, 195-198.

Wang, L.-X. (1979): *Handbook of mathematics*, People's Education Press, Beijing (in Chinese).

Wang, X. (1993): "Modelling duration and other long-term speech features in HMM-based speech recognition", *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam* **17**: 19-32.

Wang, X. (1994): "Durationally constrained training of HMM without explicit state durational pdf", *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam* **18**, 111-130.

Wang, X. (1995): "Durational modelling in HMM-based speech recognition: Towards a justified measure", in: Rubio Ayuso, A.J. & Lopez Soler, J.M. (eds): *NATO ASI Series, Vol. F147, Speech recognition and coding: New advances and trends,* Springer Verlag, Berlin, 128-131.

Wang, X., ten Bosch, L.F.M. & Pols, L.C.W. (1992): "Dimensionality and correlation of observation vectors in HMM-based Speech Recognition", *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam* **16**, 1-26.

Wang, X., ten Bosch, L.F.M. & Pols, L.C.W. (1993): "Impact of Dimensionality and Correlation of Observation Vectors in HMM-Based Speech Recognition", *Proceedings EUROSPEECH'93*, Berlin, Germany, 1583-1586.

Wang, X., Pols, L.C.W. & ten Bosch, L.F.M. (1996a): "Analysis of context-dependent segmental duration for automatic speech recognition", *Proceedings ICSLP'96,* Philadelphia, PA, 1181-1184.

Wang, X., ten Bosch, L.F.M. & Pols, L.C.W. (1996b): "Integration of context-dependent durational knowledge into HMM-based speech recognition", *Proceedings ICSLP'96,* Philadelphia, PA, 1073-1076.

Wellekens, C.J. (1987): "Explicit time correlation in hidden Markov models for speech recognition", *Proceedings ICASSP'87*, Dallas, TX, 384-386.

Woodland, P.C. & Young, S.J. (1993), "The HTK tied-state continuous speech recogniser", *Proceedings EUROSPEECH'93,* Berlin, Germany, 2207-2210.

Woodland, P.C., Leggetter, C.J., Odell, J.J., Valtchev, V. & Young, S.J. (1995): "The development of the 1994 HTK large vocabulary speech recognition system", *Proceedings ARPA Spoken Language Systems Technology Workshop*, Austin, TX, 104-109.

Young, S.J. (1992): *HTK: Hidden Markov model toolkit v1.4 Reference Manual*, Cambridge University, 1992.

Young, S.J., Russel, N.H. & Thornton, J.H.S. (1989): *Token passing: A simple conceptual model for connected speech recognition systems,* Technical report, Cambridge University.

Young, S.J. & Woodland, P.C. (1993): "The use of state tying in continuous speech recognition", *Proceedings EUROSPEECH'93*, Berlin, Germany, 2203-2206.

Young, S.J. & Woodland, P.C. (1994), "State clustering in hidden Markov model-based continuos speech recognition", *Computer Speech and Language* **8**, 369-383.

Young, S.J., Woodland, P.C. & Byrne, W.J. (1994): "Spontaneous speech recognition for the credit card corpus using the HTK toolkit", *IEEE Trans. Speech Audio Processing* **2**(4), 615-621.

Zue, V.W. (1985): "The use of speech knowledge in automatic speech recognition", *Proceedings of the IEEE* **73**(11), 1602-1615.

Zue, V., Seneff, S. & Glass, J. (1990): "Speech database development at MIT: TIMIT and beyond", *Speech Comm.* **9**, 351-356.

Index