PHONETIC EVENT-BASED WHOLE-WORD MODELING APPROACHES FOR SPEECH RECOGNITION

by

Keith Kintzley

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

February, 2014

Abstract

Speech is composed of basic speech sounds called phonemes, and these sub-word units are the foundation of most speech recognition systems. While detailed acoustic models of phones (and phone sequences) are common, most recognizers model words themselves as a simple concatenation of phonemes and do not closely model the temporal relationships between phonemes within words. Human speech production is constrained by the movement of speech articulators, and there is abundant evidence to indicate that human speech recognition is inextricably linked to the temporal patterns of speech sounds. Structures such as the hidden Markov model (HMM) have proved extremely useful and effective because they offer a convenient framework for combining acoustic modeling of phones with powerful probabilistic language models. However, this convenience masks deficiencies in temporal modeling. Additionally, robust recognition requires complex automatic speech recognition (ASR) systems and entails non-trivial computational costs.

As an alternative, we extend previous work on the point process model (PPM) for keyword spotting, an approach to speech recognition expressly based on

whole-word modeling of the temporal relations of phonetic events. In our research, we have investigated and advanced a number of major components of this system. First, we have considered alternate methods of determining phonetic events from phone posteriorgrams. We have introduced several parametric approaches to modeling intra-word phonetic timing distributions which allow us to cope with data sparsity issues. We have substantially improved algorithms used to compute keyword detections, capitalizing on the sparse nature of the phonetic input which permits the system to be scaled to large data sets. We have considered enhanced CART-based modeling of phonetic timing distributions based on related text-to-speech synthesis work. Lastly, we have developed a point process based spoken term detection system and applied it to the conversational telephone speech task of the 2006 NIST Spoken Term Detection evaluation. We demonstrate the PPM system to be competitive with state-of-the-art phonetic search systems while requiring significantly fewer computational resources.

Thesis Committee

Prof. Mounya Elhilali, Prof. Aren Jansen (Reader) and Prof. Hynek Hermansky (Reader and Advisor)

Acknowledgments

First, I would like to thank my advisor, Prof. Hynek Hermansky, whose patience, mentorship, and constant encouragement made this work possible. Over the past four and a half years, there were several moments at which this goal seemed impossibly far away. Thank you, Hynek, for keeping me going. Equally, I will be forever grateful to Aren Jansen for his role as my co-advisor. Most of the research contained in this document has its genesis in a model that Aren pioneered in his own doctoral work. Thank you for sharing your enthusiasm, insight and frequent words of encouragement. Also, I am indebted to you for your careful editing of our papers and your patience in being available for last-minute assistance.

Within the Johns Hopkins Department of Electrical & Computer Engineering, I would like to thank Prof. Mounya Elhilali for a thorough introduction to speech signal processing and for graciously serving on several of my committees during the course of my doctoral program. Additionally, I would like to thank Prof. Sanjeev Khudanpur for his encouragement to undertake this journey, his leadership in the Center for Language and Speech Processing (CLSP), and for being a sounding board

for ideas.

I was extremely fortunate to have been blessed with some wonderful friends in CLSP. I could never have been successful in this endeavor without the help of Samuel Thomas, Sivaram Garimella, Sriram Ganapathy, and Carolina Parada. Besides teaching me all the nuts and bolts of training models and running experiments and tutoring me in coursework, they were a wellspring of good advice and always willing to drop what they were doing to lend a hand. More recently I've been lucky to work with Vijay Peddinti and Harrish Mallidi who also have provided abundant assistance. Beyond Hynek's group of students, this experience would not have been the same without the friendship of Mike Carlin, Anni Irvine, Scott Novotoney, Courtney Napoles and Jonathan Weese.

Beyond Johns Hopkins, I need to acknowledge the ongoing support of several individuals in the Department of Electrical & Computer Engineering at the U.S. Naval Academy. I am certain that CAPT Bob Voigt had a big part in my being accepted into the Permanent Military Professor Program. Additionally, I would like to thank our department chair, Prof. Rob Ives, for his support and accommodation through the years.

I need to acknowledge my incredible parents. To my mother who is no longer with us, I thank you for surrounding me with love and encouragement. As well, for my father, I have not forgotten the seven years of night school that you put yourself through earning your degree in electrical engineering in order to better provide for us.

ACKNOWLEDGMENTS

Lastly and most importantly, I would never have made it without my wonderful wife Janna and our beautiful daughters Greta, Isabel and Claire, who have sacrificed so much over the last four and a half years. Thank you for perservering through times of discouragement, for enduring the many the nights and weekends I wasn't home, and for your many prayers and for your unfailing love.

Dedication

To my wife Janna and our daughters Greta, Isabel and Claire.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

All of the research presented in this work follows a central theme of addressing the technological challenges necessary to extend the point process model for keyword search from a prototype research system into a viable word recognition technology. In this chapter we outline the theoretical motivations for the point process model approach and contrast it with existing technologies. Much of the inspiration for this model is drawn from research in human speech perception and the evolving understanding of the importance of temporal aspects of the speech signal. We review several related works. Finally, the chapter concludes with an overview of the contributions of this dissertation that are presented in subsequent chapters.

1.2 Conventional speech recognition technology

Since its introduction in the 1970s, the hidden Markov model (HMM) has been firmly established as the principle architecture upon which all large vocabulary continuous speech recognition (LVCSR) systems are based, so it is worthwhile to briefly reflect on its emergence. Originally presented by both Jelinek [1] and Baker [2], the HMM provides a statistical framework with which to combine multiple sources of knowledge, namely acoustic-phonetic, lexical, and syntactic, for the recognition of a speech signal. Jelinek, already a renowned information theorist, had previously done pioneering work developing efficient decoding algorithms for convolutional codes [3]. In the context of digital communications, the fundamental problem is given a message corrupted by noise, find the most likely sequence of symbols that was produced by the sender given the structure imposed by the encoder. A convolutional code is essentially a finite state machine, its output is truly Markov, and hence the optimal decoder is a maximum-likelihood sequence estimator such as provided by the Viterbi algorithm [4]. The fundamental achievement of Jelinek was to cast the speech recognition problem in this "noisy channel" framework.

To adapt speech recognition to the mathematical structure of an HMM, the acoustic speech signal is first converted into feature vectors, typically computed every 10 milliseconds. The two dominant feature vector representations, Mel-frequency cepstral coefficients (MFCCs) [5] and perceptual linear predictive (PLP) [6], both provide a spectral representation of the speech signal informed by knowledge of the human auditory system. While the acoustic signal and corresponding sequence of feature vectors are observed variables, the unseen or "hidden" portion of the model is the set of phonetic states. The acoustic model

of each phone permits the calculation of the likelihood of a particular feature vector having been produced by a given phone. Using a lexicon to map words to their corresponding phonetic sequences enables the estimation of the acoustic likelihood of a given word. In order to determine the sequence of words, the acoustic likelihoods of words are combined with prior probabilities of word sequences provided by a separate language model. Thus, the speech decoding problem is one of finding the sequence of words which maximizes posterior likelihood.

A series of research projects sponsored by the U.S. Department of Defense's Advanced Research Projects Agency in the mid-1980s helped propel HMM-based approaches into the mainstream through regular competitions and the development of several speech databases [7]. In the years since, substantial growth in the amount of training data has enabled increasingly sophisticated acoustic and language models. As well, the diminishing cost of storage and memory and the exponential growth in processing power have allowed extremely computationally intensive algorithms to become practical. However, significant improvements in the last decade have tapered and the gap in performance between automated systems and human listeners persists. A widely cited study by Lippmann [8] quantified the difference in performance between humans and then state-of-the-art recognizers on several domains and noise conditions. The gulf in performance expanded with the complexity of the task. Relatively simple tasks consisted of isolated digit recognition (0.72% error for machine vs. 0.009% error for humans) and isolated spoken letters (5% error for machine vs. 1.6% error for humans). Domains of increasing difficulty consisted of read text from the Wall Street Journal (7.2% error for machine vs. 0.9% error for humans)

and ultimately spontaneous conversational speech as found in the Switchboard corpus (43% error for machine vs. 4% error for humans). In the time since the original publication of [8], automatic recognition has made considerable advances and current state-of-the-art word error rate on Switchboard corpus has come down to approximately 16% [9]. While the previous performance numbers reflect clean speech, machine recognition, in contrast to human performance, degrades precipitously with relatively small additions of noise.

Another aspect of HMM-based recognition highlights the crucial role of the language model. In [8], a comparison was presented using the Resource Management dataset in which the regular language model was replaced by a "null grammar" (i.e., all words are assigned equal probability), and it was observed that word error rate jumped from 3.6% to 17%. Comparable experiments with human recognition of nonsense sentences yielded an average word error rate of 2.0%, nearly an order of magnitude lower. Human and machine listeners take advantage of context, but automatic recognition systems are substantially more dependent on predictable grammar for correct recognition. This partly explains the apparent difficulty that machine recognition exhibits on spontaneous conversational speech relative to read text. Clearly, the human listener is significantly more adept at recognizing words solely based on low-level acoustic-phonetic information.

Without question, the key factor in the improvement of automatic speech recognition systems stems from the vast growth in available training data over the previous two decades [10]. It has been estimated that two to three orders of magnitude more data currently exist as compared to the mid-1990s [11]. As with most data-driven approaches, decreasing word error rates reliably follow from increases in the available training material.

However, high-quality labeled training corpora such as those published by the Linguistic Data Consortium represent multimillion dollar investments. Since reliable labeled training data is expensive, much effort has been expended to bootstrap acoustic models using cheap and abundant unlabeled data (see [12, 13]). However, even when large amounts of training data is available, in order to benefit it is necessary to increase model complexity which requires more processing time to estimate a larger number of model parameters. This increase in the parameter space is best illustrated by the proliferation of deep neural nets in acoustic modeling [14]. Even with dramatic growth in data, it is unlikely that existing speech models will ever be able to close the gap between human and machine performance. An analysis in [15] observed that the reduction in word error rate is linear with the logarithm of the total quantity of training material, and by extrapolating results from [13], it predicted that in excess of 600,000 hours of acoustic training data would be required to achieve word error rates approaching 0% on a broadcast news recognition task. In contrast, it was observed that by age 10, a human has heard roughly 10,000 hours of speech.

The HMM architecture has been tremendously successful principally because it provides a means of combing two sources of information: high-level semantic knowledge from the language model and low-level acoustic phonetic evidence from the speech signal. While a convenient structure for merging these two vital sources of information, Jelinek points out that,"These models will have no more than a mathematical reality. No claims whatever can conceivably be made about their relation to humans' actual speech production or recognition." [16]. Although the HMM framework has been incredibly productive, the difficulty in closing the gap with human speech recognition should motivate us to look to

evidence from human speech perception for inspiration.

1.3 Event-based speech recognition

A number of proposals for alternative approaches to speech recognition have arisen from the field of phonetics, the study of speech production, acoustics and perception. Human speech is the product of a small number articulators positioned in a discrete number of possible configurations. The set of speech sounds known as phones are sufficient to specify words in a language and are naturally organized by articulator configuration in terms of place, manner and voicing. However, broad phonemic categories mask tremendous variability in the acoustic realization of these sounds by different speakers in various contexts. In the study of phonology, binary-valued "features" corresponding to articulator properties were introduced by Jakobson [17] and enable the characterization of phonetic variation by general rules.

A notable alternative approach to HMM-based recognition constructed from linguistic features is found in the framework for lexical access based on acoustic landmarks presented by Stevens [18]. In this model, the analog acoustic input drives parallel streams of detectors which identify "acoustic landmarks" described as distinct acoustic features such as "peaks, valleys, and discontinuities in particular frequency ranges" that mark "centers or regions in the signal where acoustic parameters are examined." Landmarks initially identify higher-level features termed "articulator-free" features (e.g., [sonorant], [continuant], [strident]) which are subsequently used to resolve specific "articulator-bound" features in the vicinity of landmarks. Stevens proposed specific feature modules to integrate acoustic

cues and landmark times with phonetic and prosodic context in order to estimate feature values (+ or −). This approach is distinguished from the segmentation of the acoustic signal into phones as performed in HMM models; here, the landmarks are first identified then used to determine segments which form sequences of segmental units. However, multiple segmental units run in parallel rather than existing as a single partition of the signal. The ultimate goal of this system was to handle acoustic variability in the production of speech by seeking to identify the underlying articulatory states and movements, a significantly more constrained space than the time-frequency plane. He proposed a model in which the listener's lexicon consists of bundles of distinct binary-valued features and recognition is accomplished by the listener matching patterns to items in the lexicon.

The landmark framework is fundamentally different from the HMM-based recognizer paradigm. First, Stevens defines a "knowledge-based" approach that extracts acoustic correlates of linguistic features, in contrast to a statistically-based system which learns relations from speech feature vectors with limited supervision. Secondly, while the HMM operates on a single stream of feature vectors, landmarks are derived from many different cues in the original acoustic signal which can be viewed as parallel streams. Advocates of a landmark-based approach would argue that speaker independence naturally proceeds from defining acoustic parameters in relative terms [19]. In contrast, HMM-based recognizers rely on feature design, feature transformations, and parameter adaptation to achieve speaker independence. Finally, the two approaches differ fundamentally in how the speech signal is analyzed: HMM systems proceed frame-by-frame, typically in 10ms steps, applying identical effort to each frame. A landmark-based approach proceeds in an asynchronous

manner and performs analysis in regions local to acoustic events.

There does not exist a complete ASR system which fully realizes the landmark-based framework outlined in [18], but portions have been implemented in several works. In [19], a landmark-based approach is presented to recognize semivowels by first locating sonorant regions and then relevant acoustic events (energy dip, F_2 dip, F_2 peak, F_3 dip, F_3 peak). Classification proceeds by extracting acoustic properties at acoustic events and applying explicit rules using a fuzzy logic framework to identify semivowels. A further iteration of this approach is found in the event-based system presented in [20] where 13 acoustic parameters, correlates of phonetic manner features, are combined using a bank of five support vector machines (SVM) to produce to a broad-class segmentation of the speech input. In a later version described in [21, 22], the authors combine knowledge-based acoustic parameters using a probabilistic phonetic feature hierarchy. The MIT SUMMIT speech recognition system [23, 24] is perhaps the closest embodiment of a landmark-based recognition system envisioned in [18]. The SUMMIT system described in [24] identifies phone boundaries using landmarks and employs a phone-based dictionary to recognize words. An attempt to construct landmark-based acoustic models for ASR using articulatory features is found in [25]. In this work, SVM-based detectors were trained to identify distinctive features and landmarks which were then combined to produce word scores for the task of HMM lattice rescoring.

Another unique approach to keyword search based on the temporal relations between phonemes is found in [26, 27]. Both systems begin with a phone posteriorgram representation, the posterior probability distribution across the phone set estimated for each

frame (10 ms) of speech. In [26], a phone posterior trajectory, the posterior probability of a single phone as a function of time, is smoothed using a phone-specific matched filter, and the local maxima of smoothed trajectories are extracted to create "phoneme-spaced" posteriors. Words are then detected by finding segments in which the intervals between successive phonemes of a keyword fall within minimum and maximum durations. Additionally, the entire duration of the detected segment corresponding to a keyword must meet minimum and maximum limits on word duration. For each keyword, in this case spoken digits, phoneme interval and word duration limits are estimated from keyword examples. In [27], an entirely different approach to identifying keywords is employed based upon the idea of a matched filter for temporal trajectories of word posteriors. For each keyword, a two-output MLP is trained using long spans (1010 ms) of 29-phone posterior vectors (2929-dimensional input), and the word posterior output is subsequently convolved with a keyword matched filter. The peaks of the filtered trajectories are marked as detections. While the underlying method of pattern matching is entirely different from the point process model considered in the present work, there are striking similarities in terms of resolving phones to discreet points in time and in matching the temporal structure of whole words.

The first work to combine acoustic landmark-based recognition with a point process statistical modeling approach is seen in [28]. Drawing inspiration from related work in the neuroscience community, the paper evaluated a family of point process approaches including a hidden Markov model of the point process representation, explicit time-mark model, and several variants of Poisson process models. The preceding approaches were applied to an obstruent phone recognition task, and the inhomogeneous unmarked Poisson

process model was observed to outperform the other approaches. Additionally, it offered improved robustness as detector reliability decreased and operated on a very sparse representation. These results motivated the subsequent development of the point process model for keyword spotting introduced in [29].

1.4 Insights from human speech perception

While the landmark-based approaches previously introduced define a framework for recognition stemming from human speech production, the persistent gap in performance between human and machine recognition suggests that we should look to human speech perception as a source of inspiration. Indeed, some portions of the ASR pipeline have long benefited by incorporating knowledge of human auditory processes. The dominant acoustic feature vector representations, Mel-frequecy cepstral coefficients (MFCC) [30] and perceptual linear prediction (PLP) coefficients [6], both employ a warping of the frequency spectrum based upon evidence of critical bandwidths in human hearing. PLP features employ several additional transforms derived from knowledge of human speech perception. In addition to warping the frequency axis according to the Bark scale which approximates the shape of auditory filters [5], pre-emphasis consistent with a simulated equal-loudness curve is applied to account for nonuniform sensitivity of hearing [31], and finally, cubic-root amplitude compression is used to approximate the power law of hearing, the non-linear relation between intensity and perceived loudness [32].

While these are examples of low-level attributes of human auditory perception, other high-level cues known to be extremely relevant to perception are not reflected in cur-

rent systems. Abundant evidence highlights the critical role of temporal relations in human speech perception and language acquisition. Given the short duration of analysis windows commonly used in computing speech feature vectors, spectral information obviously plays a dominant role. However, it is not clear spectral information is equally critical for human perception. Experiments conducted in [33] tested the effect of replacing frequency information in speech with band-limited white noise while retaining the amplitude envelope of several frequency bands. Despite a severely degraded signal devoid of frequency information which might evidence formant structure and voicing, temporal cues alone proved sufficient to permit 90% word recognition accuracy. It is well known that the human listener can tolerate significant distortions such as peak clipping [34] and band-reject filtering to remove mid-frequency speech energy [35] and still retain high recognition accuracies. While human speech perception is robust to extensive alteration of spectral data, the most severe degradation in speech intelligibility occurs from corruption of temporal information contained in modulation spectrum between 2 to 10 Hz [36]. The low frequency modulations, commensurate with the rate of change of the vocal track, reflect the syllabic and phonetic temporal structure of speech [37].

Interestingly, there is abundant evidence stressing the importance of temporal information in the acquisition of language. Children with otherwise normal IQ and no hearing impairment who demonstrate selective language impairment commonly exhibit a basic temporal processing impairment [38]. These children are unable to perform basic two-tone sequencing and serial memory tasks when stimulus is presented in rapid succession (tens of milliseconds). It is widely believed that these children's failure to develop a phonological

inventory stems from their inability to properly segment speech which follows from a failure of temporal processing. In related work, it has been found that dyslexic children exhibit difficultly in the the detection of amplitude modulations at rates of 2-10 Hz, and this poor rhythm detection leads to impaired syllabic and prosodic perception [39].

Despite the evidence indicating the importance of the temporal aspects of human speech perception, it is well known that HMMs do a poor job of modeling segmental duration. The actual distribution of phoneme duration is well approximated by the two-parameter gamma distribution [40,41]. However, the Markov assumption in standard HMM configurations naturally gives rise to state occupancy duration which follows a geometric distribution [42]. Examples of augmenting the HMM structure to faithfully reflect phone duration can be found in hidden semi-Markov models [42], the explicit duration HMM [43], the continuously variable duration HMM [40], and the expanded state HMM [44]. While these extensions of the basic HMM structure more accurately model state duration distributions, the improvement comes at the price of increasing the number of parameters and additional topological complexity and thus far has yielded only small improvements in speech recognition accuracies.

1.5 Point process modeling of speech

Given strong evidence suggesting a central role for the temporal structure of sound in human speech perception, the focus of the research presented in this work will be extending the development of the point process model (PPM) for keyword search, a recently proposed whole-word modeling approach originally presented in [29]. As studies

have demonstrated the importance of temporal information in speech intelligibility and language acquisition, the PPM system expressly seeks to model the relative timing of phonetic events within words. This model arises from the same motivation as Stevens' landmark framework [18], namely that speech is the product of the highly-coupled movement of articulators and is robustly encoded by characteristic patterns of landmarks in time. While clearly an event-based approach, the structure of the point process model diverges fundamentally from other event-based recognition implementations in [19,21,24]. The probabilistic framework underlying the PPM approach is a Poisson counting process in which words are distinguished by inhomogeneous Poisson rate parameters that give rise to the characteristic pattern of phonetic events within the word. In addition to its probabilistic structure, the PPM approach also differs in its definition of events. Landmarks in Stevens' system corresponded to low-level, distinctive acoustic changes, whereas the point process model envisions higher-level perceptual events, specifically the occurrence of phonemes. The use of phonetic events results in an extremely sparse representation which has clear advantages for a speech recognition system in offering the potential for extremely fast processing. The development of the point process model will be reviewed in detail in Chapter 2.

1.6 Contributions and thesis organization

The original presentation of the point process model for keyword search in [29] detailed the theoretical development of this novel approach to keyword search and demonstrated its feasibility in experiments on the TIMIT and BURadio news corpus. These simple experiments yielded encouraging results but also highlighted a limitation common to most

whole-word approaches, namely the requirement for large numbers of word examples to build accurate models. Additionally, despite being based on an extremely sparse representation, the direct implementation of the detection function did not take advantage of this fact and proceeded in a slower, frame-by-frame manner. The objective of the research efforts contained in this dissertation was to develop the point process model into a competitive keyword search technology and to evaluate it relative to conventional phonetic approaches on a standard benchmark evaluation. The specific contributions in support of this goal are enumerated below.

1.6.1 Optimizing phonetic event selection

Central to point process framework is the concept of discrete phonetic events. In the original work [29], phonetic events were defined as the local maxima of the posterior trajectory above a threshold of $\delta = 0.5$ (a posterior trajectory refers to the posterior probability of a phone as function of time). In this work we address the fundamental question of how best to define phonetic events and how to describe a minimal set of events. We demonstrate the sufficiency of a representation in which each instance of a phone is described by a single phonetic event. We accomplish this through the use of phonetic matched filters, develop a metric for evaluating event selection threshold δ, and evaluate PPM keyword search performance using these filtered events (Chapter 3).

1.6.2 Bayesian approaches to whole-word acoustic modeling

Data sparsity is a common problem in systems that estimate whole-word models from data. To investigate alternatives to the original word models based on maximum likeli-

hood parameter estimates, we consider several alternative parametric modeling approaches. Inspired by the finding that phonetic events are well described by the Gaussian distribution, we develop a Bayesian approach to whole-word modeling which exploits prior knowledge of a word's phonetic form to overcome the limitation of insufficient training examples. Additionally, the model compensates for common phone confusions and pronunciation variation. We demonstrate significant gains in keyword search performance when word training examples are limited (Chapter 4).

1.6.3 Improving whole-word models without word examples

The MAP estimated whole-word models introduced in Chapter 4 achieved significantly better estimates of inhomogeneous rate parameters despite the use of a rather simplistic prior model. In this chapter we explore gains possible from using more sophisticated models of phone duration. Drawing upon previous work in text-to-speech synthesis, we develop a procedure for estimating phonetic timing distribution using CART analysis to model context dependence in segmental duration and a Monte Carlo approach to estimating phone-timing distributions (Chapter 5).

1.6.4 Speeding up PPM decoding

A recognition system based on a sparse set of discrete phonetic events should provide a substantial computational advantage over those which operate on dense, frame-by-frame representations. We reengineer the evaluation of the keyword detection function to capitalize on its discrete nature. We evaluate approaches to efficiently approximating the PPM detection function using an upper bound. By employing this bound, we reduce the

complexity of decoding from being linear in the number of frames to linear in the number of events. We demonstrate decoding speeds a factor of 50 times faster than previous decoding methods (Chapter 6).

1.6.5 Spoken term detection on conversational telephone speech

Previous experimental keyword search evaluations have only considered read text such as TIMIT and the Wall Street Journal corpora. Conversational telephone speech presents an appreciably more challenging task because of its spontaneous nature and pronunciation variation. In order to assess the PPM approach relative to known benchmarks, we tested it on the 2006 NIST Spoken Term Detection (STD) evaluation. This assessment required the development of several techniques including score normalization, handling multi-term queries and the modeling of words not present in training. In addition to achieving performance competitive with other phonetic keyword search systems, the PPM index construction time and size were better than any keyword search system entered in the NIST evaluation (Chapter 7).

Chapter 2

The Point Process Model for Keyword Search

Consistent with the importance of the temporal structure of the speech signal, the point process model provides a probabilistic framework for word recognition based on temporal patterns of discrete phonetic events. In this chapter we detail the original development of the point process model which will be used throughout this work.

2.1 Background

As noted in Chapter 1, the first example of an event-based speech recognition constructed on top of a point process representation appeared in [28]. In that work, the superior performance of an obstruent phone recognition system based on an inhomogeneous Poisson process modeling approach laid the groundwork for the point process model for keyword search presented in [29]. The framework detailed in [29] is the starting point for

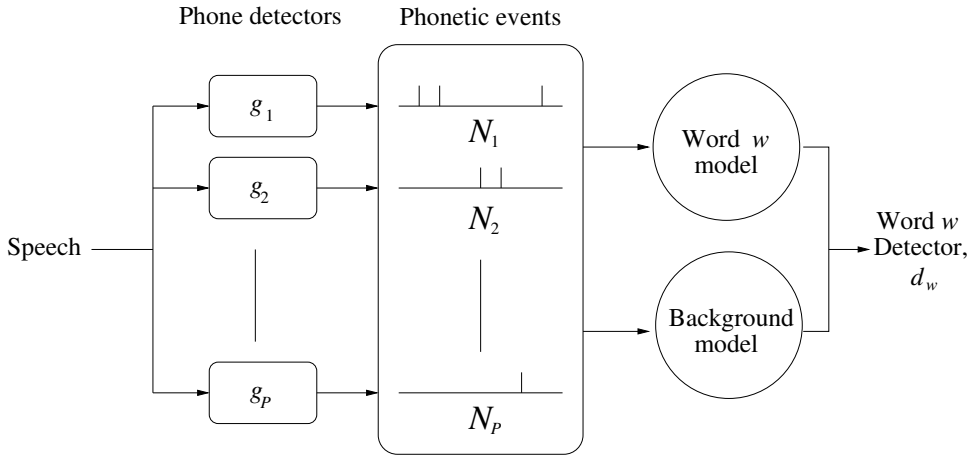

Figure 2.1: System diagram of the point process model for keyword search.

the research in this dissertation, and in this chapter we review its development.

The general structure of the point process model for keyword search is depicted in Figure 2.1. First, the acoustic speech signal input is converted into discrete phonetic events. We assume the existence of a set of phone detectors g_p for each phone p in the phone set $\mathcal{P}$ whose output is converted to discrete points in time to evidence the occurrence of a particular phone. The specific detector implementations and methods of generating events will be addressed in Chapter 3. Candidate occurrences of a keyword are identified from the detection function defined as the ratio of the likelihood of a set phonetic events under a keyword model relative to its likelihood under a background model. Formally, given a keyword w and a set of observed phonetic events $O(t)$ beginning at time t, the detection function $d_w(t)$ is given by

$$d_w(t) = \log \left[\frac{P(O(t)|\theta_w)}{P(O(t)|\theta_{bg})} \right], \tag{2.1}$$

where θ_w corresponds to the keyword-specific model parameters and θ_{bg} corresponds to background model parameters. This detection function is simply a log likelihood ratio

evaluated at time t which takes large values when it is likely that keyword w occurred. For each phone $p \in \mathcal{P}$, we define $N_p = \{t_1, \ldots, t_{n_p}\}$, the set of points in time at which phone p occurs relative to time t. The observation $O(t) = \{N_p\}_{p \in \mathcal{P}}$ is thus the collection of these sets of points. Assuming for the moment a fixed keyword duration T, we will now specify the form of the models which yield estimates of $P(O(t)|T, \theta_w)$ and $P(O(t)|T, \theta_{bg})$.

2.2 Poisson process models

A Poisson counting process is a discrete random process in which the time between arrivals of events are independent, identically distributed exponential random variables. If $\eta(t)$ is defined as the total count of events in the interval $(0, t]$, then the probability of k arrivals in the interval t_a to t_b is given by the Poisson probability mass function,

$$P\{\eta(t_b) - \eta(t_a) = k\} = \frac{(\lambda\tau)^k}{k!} e^{-\lambda\tau}, \tag{2.2}$$

where $\tau = t_b - t_a$. Because the exponential distribution is *memoryless* (i.e., the probability of an arrival after time t is independent of the time elapsed from the previous arrival), it follows that the Poisson counting process has *independent increments*, meaning that the set of n random variables $\{\eta(t_1), \eta(t_2) - \eta(t_1), \ldots, \eta(t_n) - \eta(t_{n-1})\}$ are jointly independent for all $t_1 < t_2 < \cdots < t_n$ and for all $n \geq 1$. Consequently, it can be readily shown that the likelihood of the set of events N_p in the interval $(0, T]$ is given by

$$P(N_p) = (\lambda_p)^{n_p} e^{-\lambda_p T}, \tag{2.3}$$

where λ_p is the Poisson rate parameter corresponding to phone p and $n_p = |\{t_i \in N_p | t_i \in (0, T]\}|$. The constant or *homogeneous* rate parameter λ_p is not a function of time and

corresponds to the average or background arrival rate of the phonetic events for phone p independent of any particular word. Under the simplifying assumption that the Poisson process for each phone p is independent of other phones, we can then express the likelihood of the entire collection of events $O(t)$ under the background model given T as

$$P(O(t)|T, \theta_{bg}) = \prod_{p \in \mathcal{P}} (\lambda_p)^{n_p} e^{-\lambda_p T}.$$

To calculate λ_p, if we have N example segments each of duration T and observe a total of K events corresponding to phone p, then the maximum likelihood estimate of λ_p is given by

$$\lambda_p^* = \underset{\lambda}{\operatorname{argmax}} K \log \lambda - \lambda NT = \frac{K}{NT}. \tag{2.4}$$

The background model θ_{bg} consists of the set of homogenous rate parameters $\{\lambda_p\}_{p \in \mathcal{P}}$.

Now we consider a Poisson model describing the generation of phonetic events within a word. For the inhomogeneous Poisson process, the rate parameter $\lambda_p(t)$ is not constant but is instead a function of time. In the context of a word, $\lambda_p(t)$ takes large values within segments of the word at which phone p is likely to occur. While $\lambda_p(t)$ is assumed to be a continuous function of time, we will consider approximating it as a piecewise constant function over D uniformly spaced divisions in $(0, T]$, with the inhomogeneous rate parameters for phone p denoted $\lambda_{p,d}$ for $d = 1, \ldots, D$. We make a corresponding subdivision in our collection of observations N_p into D partitions specified as

$$N_{p,d} \equiv \{t_i \in N_p | t_i \in ((d-1)\Delta T, d\Delta T], i = 1, \cdots, n_{p,d}\},$$

where $\Delta T = T/D$. Just as in Equation (2.3), the likelihood of the event set $N_{p,d}$ for the dth segment is given by

$$P(N_{p,d}) = (\lambda_{p,d})^{n_{p,d}} e^{-\lambda_{p,d} \Delta T}, \tag{2.5}$$

and applying the independent increments property, the likelihood of the event set N_p under the inhomogeneous model is

$$P(N_p) = \prod_{d=1}^{D} P(N_{p,d}) = \prod_{d=1}^{D} (\lambda_{p,d})^{n_{p,d}} e^{-\lambda_{p,d}\Delta T}.$$

In a similar manner to Equation (2.4), the maximum likelihood estimates of $\lambda_{p,d}$ can be calculated

$$\lambda^*_{p,d} = \frac{K_{p,d}}{N\Delta T},$$

where $K_{p,d}$ denotes to the total count of events corresponding to phone p in the dth segment over N keyword examples. The set of inhomogeneous rate parameters $\{\lambda_{p,d}\}_{p\in\mathcal{P},d=1,\ldots,D}$ is referred to as the word model θ_w. An example word model for the keyword "greasy" is depicted in Figure 2.2. Thus, the likelihood of the entire collection of points $O(t)$ under the word model can be expressed

$$P(O(t)|T,\theta_w) = \prod_{p\in\mathcal{P}} \prod_{d=1}^{D} (\lambda_{p,d})^{n_{p,d}} e^{-\lambda_{p,d}\Delta T}. \tag{2.6}$$

2.3 Point process model detection function

To this point we have assumed a fixed keyword duration T, so we will now describe how keyword duration is incorporated into the model. Underlying our entire approach is the assertion that words are distinguished by a characteristic pattern of phonetic events in time. We now make a further simplifying assumption that this representative pattern is independent of actual keyword duration. In other words, multiple observations of the same keyword scaled to the interval $(0, 1]$ will result in the same pattern and thus can be modeled by the same set of inhomogeneous rate parameters. To incorporate this, we define a new

set of points with respect to a normalized time scale as $N'_p = \{t'_i | t'_i = t_i/T,\ \forall t_i \in N_{p,d}\}$ with $O'(t) = \{N'_p\}_{p\in\mathcal{P}}$. After a change of variables, the probability in Equation (2.6) with $O'(t)$ becomes

$$P(O'(t)|T,\theta_w) = \prod_{p\in\mathcal{P}}\prod_{d=1}^{D}(T\lambda_{p,d})^{n_{p,d}}e^{-\lambda_{p,d}/D}.$$

Our estimates of $P(O'(t)|T,\theta_w)$ and $P(O(t)|T,\theta_{bg})$ are conditioned on the latent variable T, therefore, we may compute the detection function in Equation (2.1) on an unknown utterance by integrating over T in

$$d_w(t) = \log\left[\int_0^\infty \frac{P(O'(t)|T,\theta_w)P(T|\theta_w)}{T^{|O(t)|}P(O(t)|T,\theta_{bg})}dT\right].$$

In practice this integral is approximated by a summation over a discrete set $\mathcal{T}$ of candidate durations spaced at even intervals. We estimate $P(T|\theta_w)$ for each $T \in \mathcal{T}$ based upon keyword examples from training. After finding the parameters for θ_w, θ_{bg} and $P(T|\theta_w)$, we can calculate $d_w(t)$ given an observation $O(t)$. A keyword detection occurs whenever $d_w(t)$ exceeds threshold δ_w which may be determined from development data.

2.4 Keyword search performance metrics

Putative keyword detections are marked by a detection time and a detection score (i.e., the magnitude of the detection function at the time of detection). The reliability of detections varies directly with detection score. The keyword search performance metric adopted throughout most of this work is average figure of merit (FOM), the mean detection rate given $1, 2, \ldots, 10$ false alarms per keyword per hour as the detection threshold is varied [45]. FOM provides a summary of the performance at the higher precision portion of the receiver operating characteristic curve.

In more recent years, an alternative performance metric called term-weighted value (TWV) has become the standard in the spoken term detection (STD) community. As defined in [73], the term-weighted value at a detection threshold θ is given by

$$\mathrm{TWV}(\theta) = 1 - \underset{term}{\mathrm{average}}\{\mathrm{P}_{\mathrm{miss}}(term, \theta) + \beta\, \mathrm{P}_{\mathrm{FA}}(term, \theta)\},$$

and represents the average over all query terms. The factor β is a penalty for false alarms, and while the maximum possible TWV is 1.0, negative values of TWV are possible for high false alarm rates. In the 2006 NIST STD evaluation, a value of $\beta = 999.9$ was specified. In addition to providing a list of detections and associated detection scores, participants in that STD evaluation were also required to specify a binary "YES/NO" decision corresponding to a specific decision threshold θ. The TWV computed at this specific θ was defined as the actual term-weighted value (ATWV), and this metric will be used in the evaluation presented in Chapter 7.

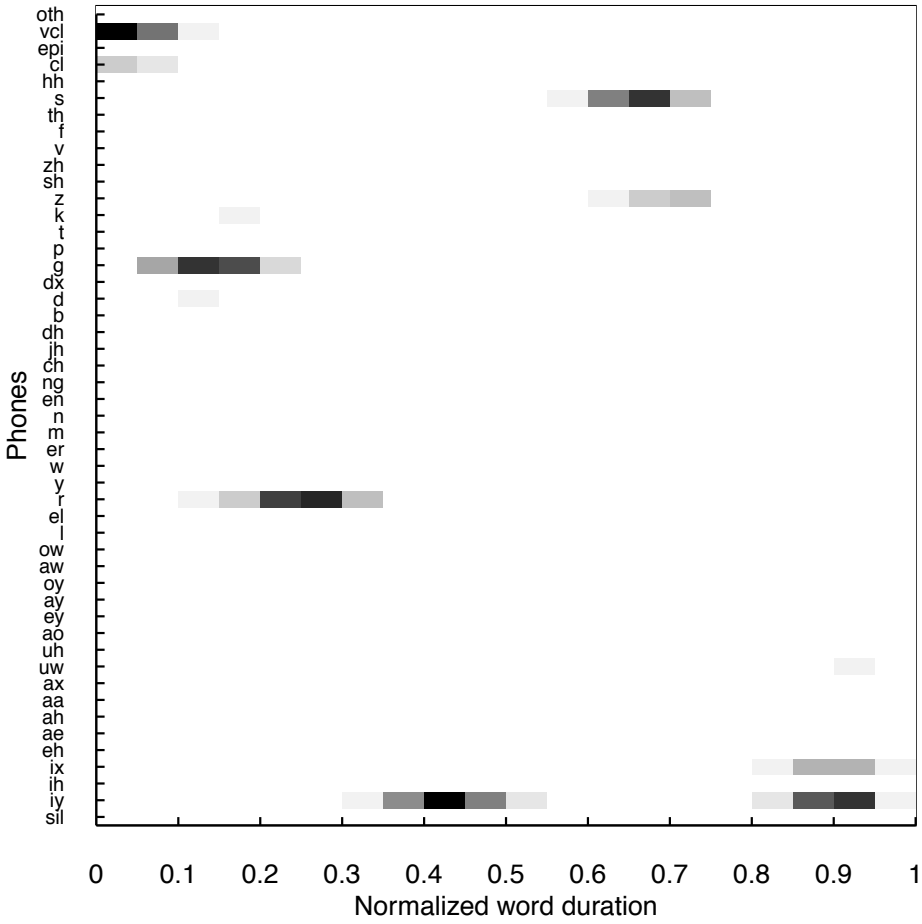

Figure 2.2: Word model θ_w for the keyword "greasy" based on 462 keyword examples in the TIMIT corpus. Segments correspond to inhomogeneous rate parameters for $D = 20$ subdivisions of the normalized word duration, and dark segments correspond to high values of $\lambda_{p,d}$

.

Chapter 3

Optimizing Phonetic Event Selection

In this chapter we develop and assess methods for defining the phonetic events, the most basic element of our representation of the acoustic speech signal within the Poisson process modeling framework. Building from previous work, we evaluate phonetic events derived by convolving phone posterior trajectories with phonetic matched filters and present a mutual information based metric for optimizing event selection thresholds. Importantly, we demonstrate that the representation of a phoneme using a single event yields improved keyword search performance and a significantly sparser representation of speech. The techniques presented in this chapter are used in all subsequent work.

3.1 Phonetic events and the point process model

The event-based approaches reviewed in Chapter 1 diverge fundamentally from common HMM systems in that speech is represented by an asynchronous collection of distinct points in time rather than a continuous sequence of feature vectors. Most of the event-based methods previously described had a common root in the Stevens' landmark framework detailed in [18]. In the context of Stevens' work, events or landmarks are defined as "peaks, valleys, and discontinuities in particular frequency ranges." A common attribute of the preceding systems is that feature streams correspond to traditional, linguistically-based articulatory features which can serve as a basis set for identifying phonemes.

In the point process implementation presented in this work, we start with a phonetic representation of speech instead of low-level articulatory features. There are several reasons for beginning at a phonetic, or more precisely, a phonemic representation. Principle among our motivations is the hypothesis that the structure of words is encoded in the temporal sequence of phonemes. Another more practical reason to begin with a phonetic representation is the abundance of labelled phonetic data necessary for the training of phone detectors. In many of the previously presented event-based systems that utilize articulatory features, the availability of corpora which include acoustic feature labeling was extremely limited. Additionally, many corpora lacking explicit phonetic labels can nonetheless be used via phonetic forced alignments derived from large vocabulary speech recognition systems. Finally, a phonetic representation allows us to construct models for unseen words based upon their dictionary (phonetic) form.

3.2 Phone detectors

Defining phonetic events first requires some means of detecting the presence of phonemes in a speech signal. In general, given the set of all phones $\mathcal{P}$, one can construct independent detector functions g_p for each $p \in \mathcal{P}$ which take as input speech feature vectors in $\mathbb{R}^D$ and generate a real-valued output which takes large values when phone p is present. Furthermore, detectors can be constrained to produce values in $[0, 1]$ such that for a given acoustic feature vector $x \in \mathbb{R}^D$, $g_p(x) = \Pr(p|x)$. Thus, at each time t, the collection of $g_p(x)$ for all $p \in \mathcal{P}$ represents posterior probability distributions across the phone set. The vector time series of posterior distributions across the phone set as a function of time is commonly referred to as a *phone posteriorgram*. Further, we refer to the posterior probability of a single phone as a function of time as a *posterior trajectory*. Both are illustrated for a sample sentence from the TIMIT corpus in Figure 3.1.

A posteriorgram is a general representation for any collection of mutually exclusive detection functions which estimate posterior probability and is independent of a specific machine learning implementation. In this chapter, we consider two forms of acoustic models for estimating phone posterior probabilities typical of speech recognition systems. For the first approach, phone detectors are based on Gaussian mixtures models (GMM), the standard acoustic modeling architecture common to the majority of speech recognition systems since the advent of HMMs. The GMM acoustic models employed for the experiments in this chapter use standard 39-dimensional MFCC features for each speech frame with 8 mixture components (full covariance) to estimate $\Pr(x|p)$ for each frame where $x \in \mathbb{R}^{39}$ and $p \in \mathcal{P}$. From $\Pr(x|p)$ and the prior phone probabilities $\Pr(p)$, the posterior probabilities $P(p|x)$

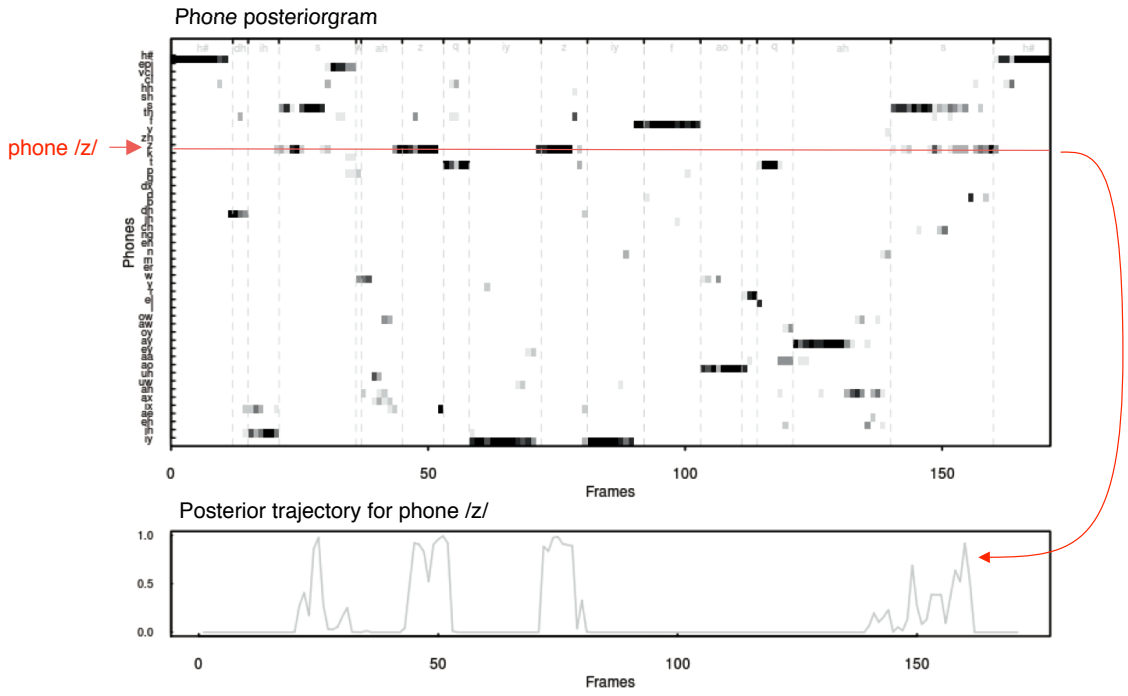

Figure 3.1: Phone posteriorgram for the TIMIT sentence "This was easy for us." For the plot of $\Pr(p|x)$ for each frame of the utterance, the darker shades depict higher probability. A posterior trajectory illustrated for the phone /z/ is defined as the posterior probability of a single phone as a function of time.

are computed using Bayes' rule. Further details of this GMM implementation are found in [29].

While GMM-based acoustic modeling has predominated for more than three decades, in the last few years discriminative acoustic models in the form of multilayer perceptrons (or neural networks) have surpassed GMMs and currently define state-of-the-art performance in phonetic recognition [14]. As an alternative to GMM-based posterior data, we now briefly describe a discriminative modeling approach to the estimation of phone posteriorgrams using multilayer perceptrons (MLP). In the simplest form, the MLP acts as a function which maps $x \in \mathbb{R}^D$ corresponding to a D-dimensional feature vector for one frame of speech to a distribution of probabilities over the phone set $\mathcal{P}$. In a 3-layer MLP, an input layer of

D nodes is followed by a hidden layer whose dimensionality is much larger than the input layer. The hidden layer nodes are connected to an output layer nodes where each node corresponds to a phone class. Network weights are learned by choosing a cost function relating input feature and output target values, and then determining the set of network weights to minimize cost, typically by gradient descent with recursive back propagation of error.

For multiclass classification, it has been shown that multilayer perceptrons can be trained to provide good estimates of Bayesian *a posteriori* class probabilities [46]. To learn such a mapping, we begin with a set of training examples consisting of acoustic feature vectors for frames of speech and the corresponding phonetic class labels. Phone class labels are encoded as "one-high" binary targets; the target value of the output node corresponding to the true phone class is 1 and remaining nodes all take value 0. To ensure the output node values correspond to a proper probability distribution (i.e., all output values lie in $[0, 1]$ and the sum of all output nodes is unity), we construct the network using the softmax function for the output units [47]. The squared-error cost function is appropriate for many tasks, however, with binary output targets the use of the *cross-entropy* cost function has the theoretical justification of minimizing the Kullback-Liebler distance between the estimated output distribution and the true target distribution.

In the previous description of a basic 3-layer perceptron, the dimension of the input layer was D-dimensions, the same dimensionality as the feature vector corresponding to a single frame of speech. The typical frame of speech is derived from 20-30 ms analysis windows sampled every 10 ms. However, this period is significantly shorter than the physical

movement of speech articulators, auditory perception phenomena and typical syllable length which reside in timescales on the order of hundreds of milliseconds [37]. In many previous MLP-based phone recognition experiments, it has been demonstrated that the inclusion of feature vectors for the adjacent 5 frames (i.e., a 9-frame context window) results in significant improvements in accuracy [48].

Another variation of the basic MLP structure that has also demonstrated improved phone accuracy is the sparse multilayer perceptron (SMLP) [49]. The SMLP is composed of a hierarchical structure consisting of two MLPs in tandem. The first MLP uses a 9-frame context at the input layer, a 1000 node hidden layer, and a 3-state phone posterior probability output layer. Instead of an output layer which consists of targets representing just the individual phone classes, the output nodes correspond to a partition of the labels into three sub-phone labels which differentiate the beginning, middle and end as separate classes. For the first MLP the usual cross-entropy cost function includes an additional sparse regularization term to enforce sparsity in the outputs of the first hidden layer. Further, the input of the second MLP includes 23 frames of context from the preceding 3-state phone posterior output of the first MLP, a 3500 unit hidden layer, and an output consisting of 49 phone classes. In has been claimed that the extended temporal context (150-230ms) of the second MLP is valuable in learning patterns of phonetic confusions of the first MLP and the phonotactics of the language [50]. Both GMM and MLP-based systems produce posteriorgram outputs, thus phonetic events can be generated in the same manner.

3.3 Phonetic event selection by local maxima

Given the posteriorgram representation, we now address the question of how to distill this dense, frame-by-frame posterior data into a sparse set of phonetic events. We begin by considering a single phone and its corresponding posterior trajectory as depicted in Figure 3.1. In the original presentation of point process models for keyword search [29], phonetic events were defined as the points in time corresponding to local maxima of the posterior trajectories exceeding a threshold of $\delta = 0.5$. The choice of this threshold value was motivated by the intuition that probability one-half corresponds to the Bayes optimal binary classification decision. A simple illustration of how events are derived from posterior trajectories in this manner is depicted in Figure 3.2a. We will subsequently refer to these as *local maxima* based events. For events defined in this manner, it is common to observe several local maxima occuring in the duration of a particular phone resulting in multiple phonetic events per phone instance. However, the sparsest representation would be characterized by just one event per phone. It should be noted that although these points appear to convey magnitude and timing information, for the point process model, time of arrival is the only relevant statistic. Magnitude information merely determines whether a local maxima is sufficient to be deemed an event. Thus, the phonetic events for the example Figure 3.2a are $\{28, 47, 52, 77, 80, 149, 168, 170\}$, the frames at which the points occur.

We now consider alternative methods of deriving phonetic events. To begin, let us imagine for the moment the existence of ideal phone detectors whose outputs are either 0 or 1 and operate with 100% accuracy, perfectly matching phonetic labels for every frame.

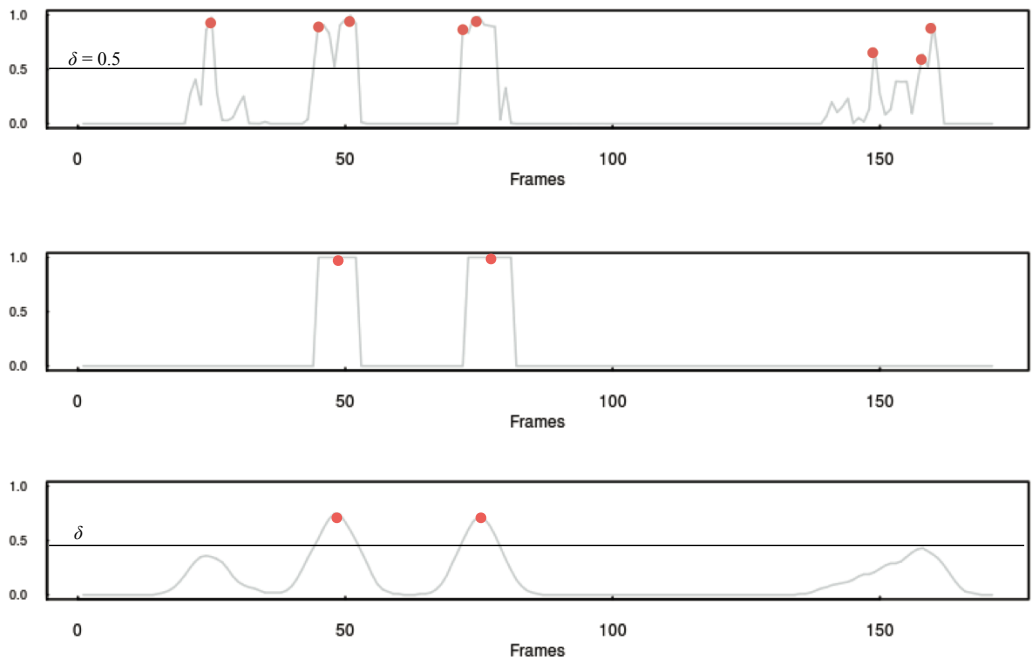

Figure 3.2: Examples of posterior trajectories of phone /iy/. Symbols indicate the set of points which would be marked as events given the threshold δ and correspond to (a) local maxima events, (b) oracle events, and (c) filtered events.

Given such detectors, for each phone trajectory we could define a phonetic event as the midpoint of the phone labels (see Figure 3.2b) yielding precisely one event per label. Any fewer points would imply a loss of phonetic information, so this set represents a lower bound on the number of events that we could hope to obtain. While such ideal detectors do not exist, it is a simple matter to derive the set of events they would produce using phonetically labeled data. Thus, we will refer to these ideal events as *oracle* events. While this represents the sparsest set of events, it is not immediately apparent that the point process keyword search system will perform well with such a limited set of points.

3.4 Phone matched filters

The posterior trajectories obtained from real detectors as shown in Figure 3.2a differ significantly from the ideal binary-valued output shown in 3.2b. However, it is ap-

parent that both trajectories evidence the same underlying phonetic events. If we consider the problem from the perspective of a communications system, the speaker of the utterance is transmitting information as a sequence of distinct phones which is converted into an acoustic signal. In a sense, our phone detector acts as a receiver, outputting, albeit imperfectly, the presence of a particular phone at each frame. Yet, what we really desire is the original underlying phone string which constitutes the message. In a manner similar to [51], if we consider the phone labels to be the clean transmitted signal and posterior output as noise-corrupted received signal, one mechanism for detecting the original symbols would be to apply matched filters. Since phone instances vary in duration, we obviously do not have a fixed waveform from which to design a matched filter, so we consider the average signal profile instead. In [51], filters specific to each phone were obtained by averaging 0.5 second windows of the *actual* posterior trajectory (as in Figure 3.2a) for all occurrences of the phone aligned to the true phone centers determined from the labels. In this work, we derived equivalent filters by instead averaging 0.5 second windows of the *ideal* trajectory (as in Figure 3.2b) extracted directly from phone labels. Figure 3.3 shows the filter shapes resulting for a selection of phones. Given these filters, we then convolve each raw posterior trajectory with its corresponding filter to obtain a smoothed posterior trajectory as shown in Figure 3.2c. We then define *filtered* events as the local maxima of the smoothed trajectory exceeding a threshold δ. Visually, these events align very closely with the oracle events in Figure 3.2b.

The purpose of the matched filter is to act as a smoothing function for the posterior trajectories, integrating probability estimates over a contiguous span of speech frames. It

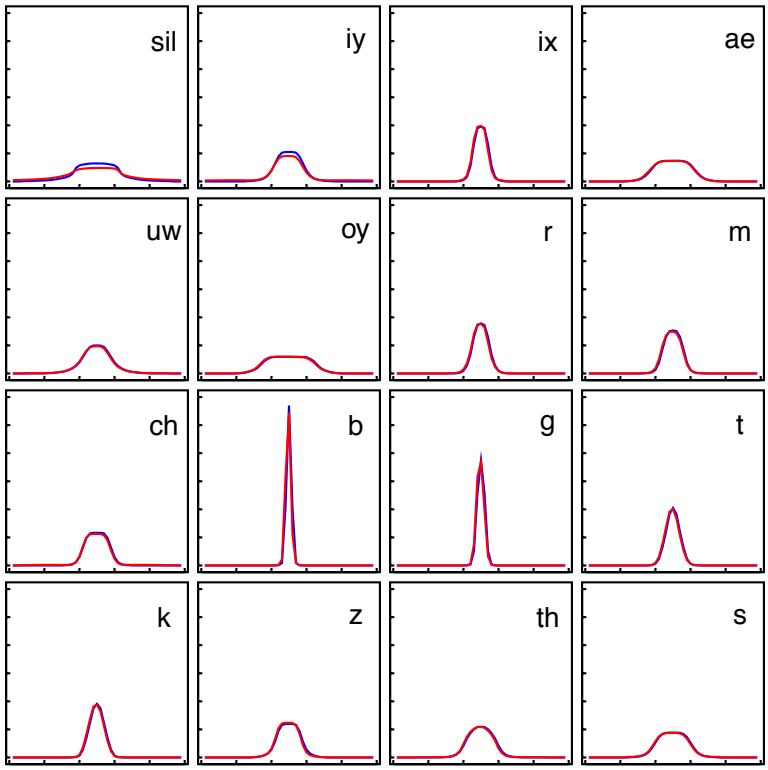

Figure 3.3: Examples of matched filters for a selection of phones. Filter profiles colored blue were derived from oracle posterior trajectories based upon phonetic labeling. Filter profiles colored red were derived from actual posteriorgram trajectories.

is desirable that the resulting filtered output be bounded in the range $[0, 1]$. To ensure this, after the filter parameters are estimated from label data, we normalized the filter coefficients to sum to one. Had this normalization not been performed, phones with long average durations (e.g., vowels, semivowels) would result in filtered local maxima with larger values while the opposite would be true of those phones with short average durations (e.g., plosives). This would have complicated the search for an optimal event threshold δ since it is unlikely that a single threshold would been appropriate for all phones.

3.5 Evaluating phonetic event selection techniques

To compare phonetic event selection techniques, we now propose a metric formulated using the mutual information between phone labels and the resulting phonetic events. We envision the phone detector output and the event selection mechanism in the "noisy channel" framework as depicted in Figure 3.4. Each channel input is a single phone (spanning successive frames) uttered by the speaker as indicated by the phone labeling. The channel output consists of all the phonetic events which occur during the span of the input phone. For the simplest case consider oracle events for which there exists a single phonetic event output for each input phone produced by the speaker. For this ideal channel there is no loss of information, so the mutual information between the input and output distribution is just the entropy of the input.

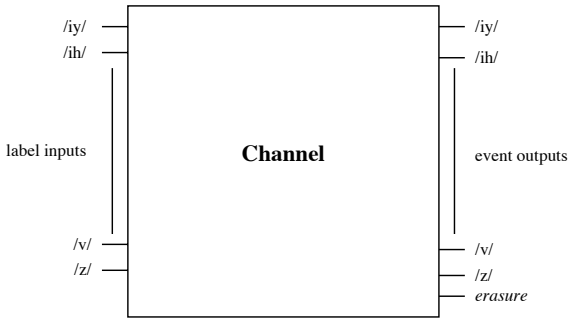

Figure 3.4: "Noisy channel" model illustration of phonetic event detection.

The diagram in Figure 3.5 illustrates the process of estimating the distribution of input and output events. The left side of the figure shows sample phonetic events annotated with phone labels at the top. In a normal communication channel, each symbol input to the channel results in a single, possibly corrupted, symbol being received. For local maxima and filtered events, our communications analogy requires some augmentation since a single

phone label input does not always result in a single phonetic event output. First, it is possible that an input produces no output, so we will augment our set of outputs with an *erasure* (deletion) event as shown for the missing phonetic event at frame 37 in Figure 3.5. It is also possible that a single input produces multiple outputs (insertions). We propose handling this with fractional counts. Consider a count matrix in which the rows are input phones and the columns are output phones with the erasure symbol. Suppose the phone /s/ is uttered resulting in phonetic events /s/,/s/,/s/, and /z/. In row /s/ of the count matrix we would record a count of 3/4 in the column corresponding to output /s/ and 1/4 in column /z/. This matrix corresponds to an estimate of the joint distribution of input and output events, and from it we can compute mutual information.

3.5.1 Choosing an optimal phonetic event threshold

As illustrated in Figure 3.2, the set of phonetic events extracted from a posteriorgram is a function of the threshold δ which dictates whether a posterior trajectory local maxima is recorded as an event. Hence, the fractional count matrix shown in Figure 3.5 is unique to a particular choice of δ. As would be expected, setting a high threshold produces many erasures and results in low mutual information. Alternatively, a very low threshold produces numerous false alarms also resulting in low mutual information. Thus, it was hoped that this measure would allow us to find good thresholds between these extremes in order to maximize mutual information between phone labels and the resulting phonetic events.

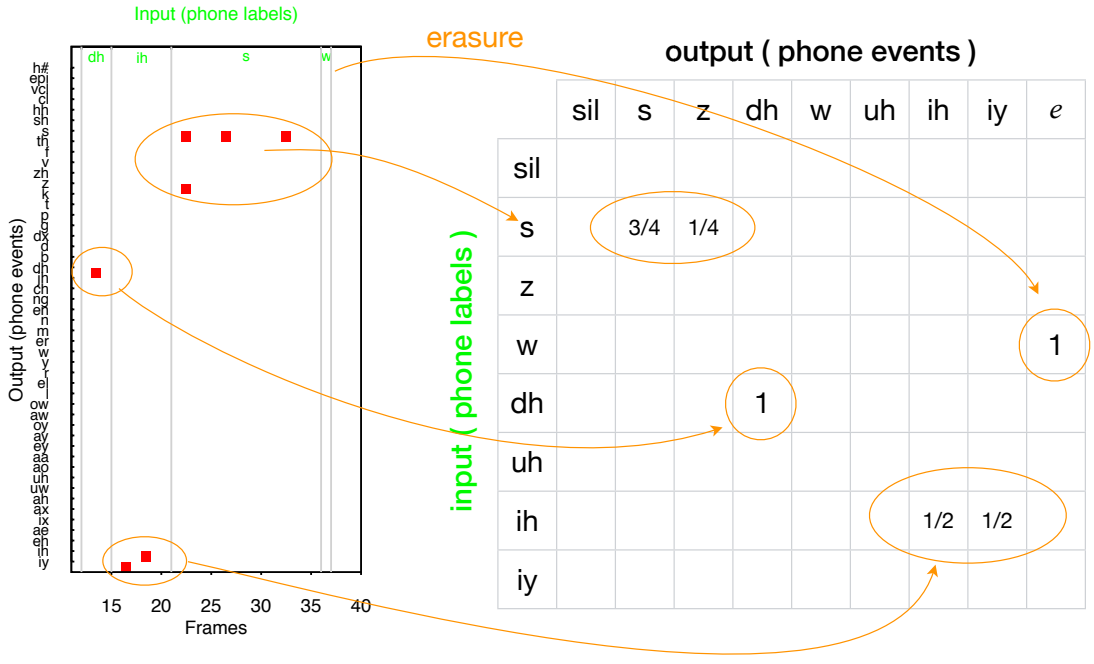

Figure 3.5: An illustration of computing the fractional count matrix used to calculate mutual information between input and output phonetic events.

3.5.2 Optimal single threshold

To explore approaches to identifying an optimal threshold δ, we considered the simplest case, a single common threshold used for all phones. This one dimensional search only required the accumulation of count matrices and computation of mutual information as threshold was swept from 0 to 1. For this evaluation, we began by generating both GMM and SMLP-based phone posteriors for the TIMIT database and then derived local maxima and filtered events over a range of thresholds. To compare these sets of events, we computed

our mutual information metric sweeping over a range of values of δ. The results are plotted in Figure 3.6 and show that mutual information for SMLP events exceeds GMM events, and that filtered events yield a slightly higher mutual information than local maxima events in both GMM and SMLP cases. The filtering operation necessarily reduces the magnitude of the peaks in filtered trajectories which accounts for the difference in location of the peak mutual information. The SMLP posteriors employed here yield state-of-the-art performance in standard TIMIT phone recognition experiments, so it is not surprising that they exhibited higher mutual information than GMM posteriors. Finally, the mutual information of the oracle events at 5.16 bits is exactly the entropy of the input distribution.

3.5.3 Experiments with phone-specific event thresholds

In addition to a single threshold, we also explored optimizing the thresholds of each individual phone or phone class. A grid search of 10 thresholds for 49 phones would require 10^{49} evaluations of mutual information, clearly not a feasible task. A more tractable option is to group phonemes into two classes and to optimize just two thresholds. Based on standard broad class phone assignments, we grouped vowels and semivowels into one class and the everything else in the other. The plot in Figure 3.7 shows the results of a grid search for optimal thresholds for vowels/semivowels (t_1) and other (t_2) for the TIMIT `si/sx` data using a threshold step size of 0.02. We observed that the gain in mutual information is only 0.000396 bits as compared to a single common threshold. Similar evaluations were performed for each of the five broad classes. The largest gain in mutual information was only 0.001581 bits, an insignificant improvement.

While a grid search of 49 phone classes is infeasible, other approaches to the

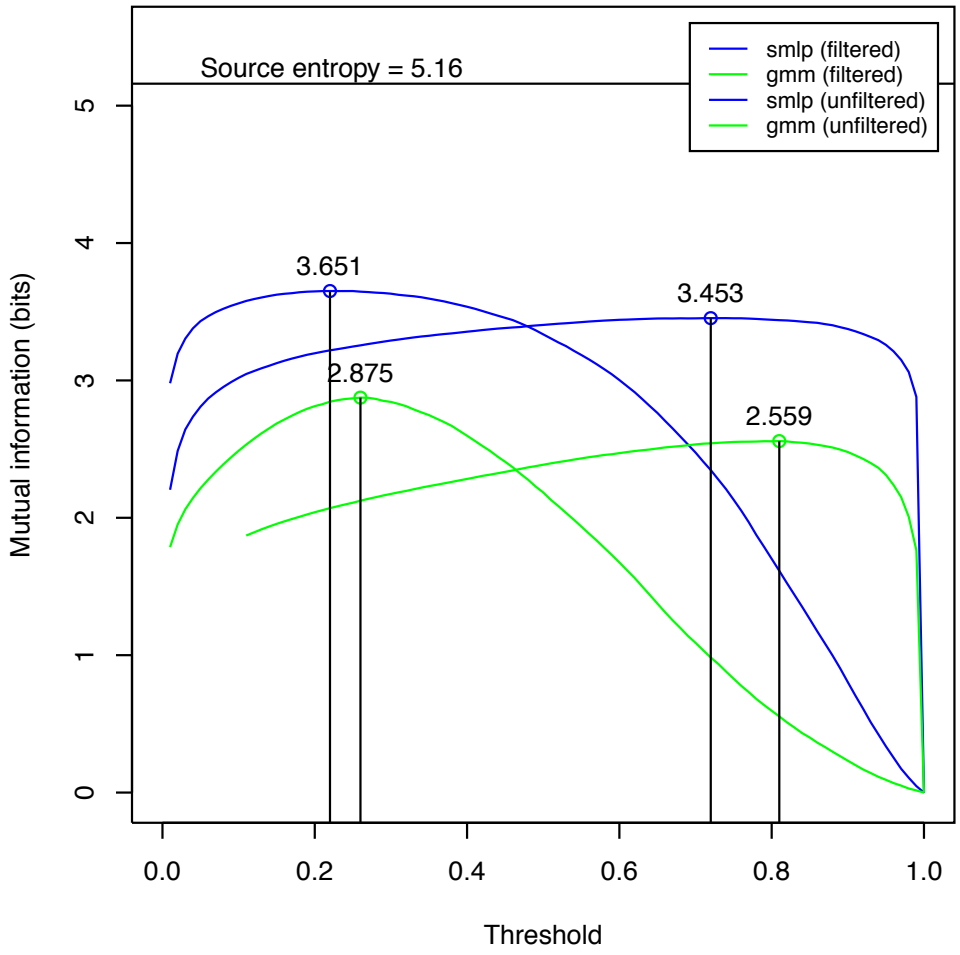

Figure 3.6: Mutual information as a function of threshold for local maxima and filtered events using GMM and SMLP-based posteriorgrams of TIMIT `si/sx` test sentences.

maximization problem exist. Unfortunately, it is not possible to compute the gradient of the mutual information as a function of thresholds of each of the phones. One approach to finding a maximum is coordinate descent. We began by initializing the thresholds of all 49 phones to a single common value, then checked for the maximum change in mutual information as one threshold was changed by $+\Delta t$ and $-\Delta t$ and the other 48 were held constant. This was repeated for all 49 phones. The threshold of the phone for which Δt produced the largest increase in mutual information was changed, and the process was

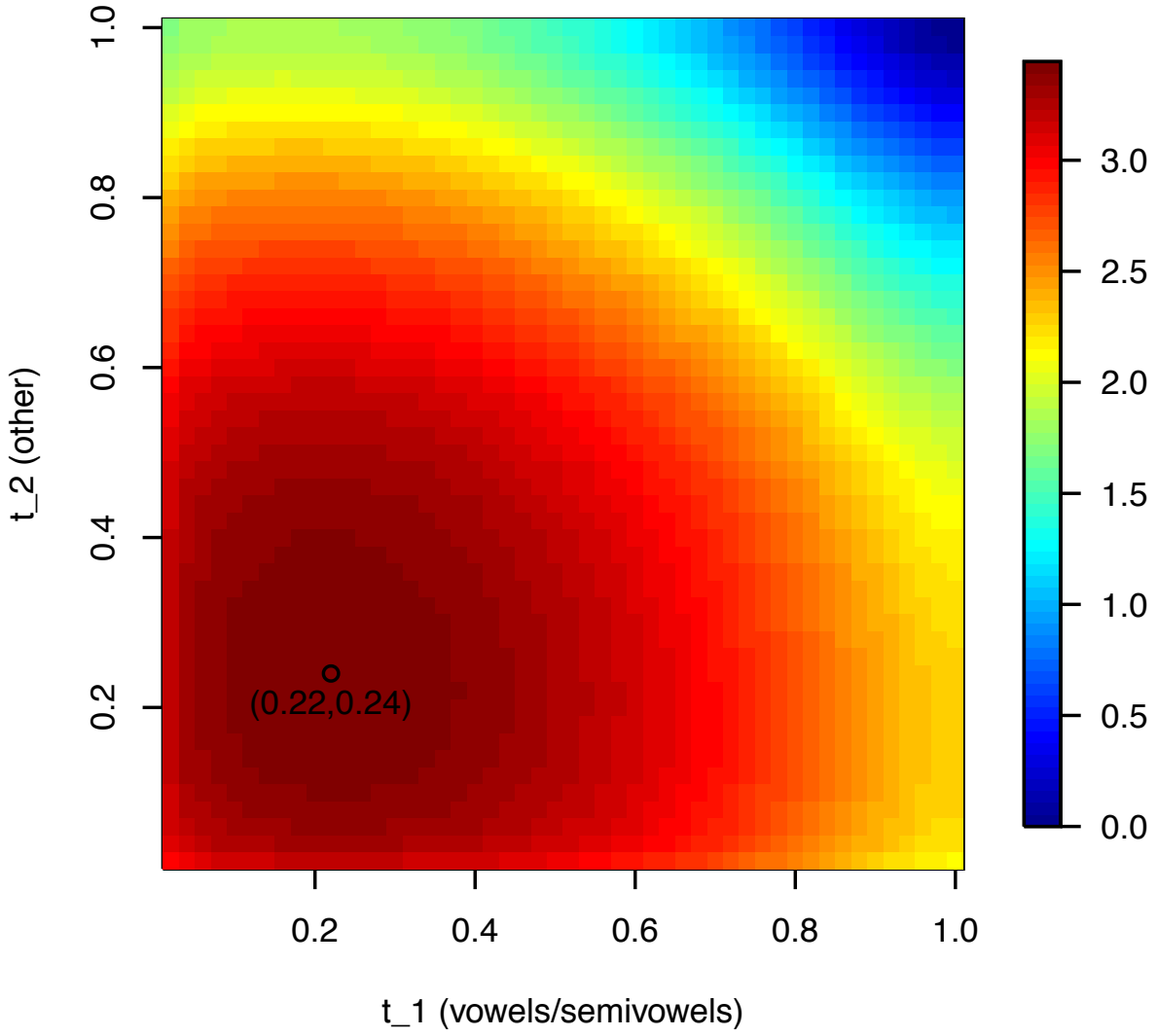

Figure 3.7: Mutual information as a function of two thresholds for filtered local maxima of SMLP-based posteriorgrams of TIMIT `si/sx` test sentences. Threshold t_1 was applied to vowels/semivowels and t_2 was applied to all other phones.

repeated until the incremental improvement in mutual information fell below a threshold $\epsilon = 0.000001$. The approach has no guarantee of finding the optimal solution, but it will necessarily find a better solution than the single common threshold of Figure 3.6. Also,

there is reason to believe based on the preceding plots that the mutual information function is fairly smooth. We considered $\Delta t = 0.10$ and began the search from uniform threshold vectors of $0.1, 0.2, \ldots, 0.9$. Each of these initializations converged to the same solution yielding a mutual information of 4.541056 bits for TIMIT `si/sx` train data, an improvement of 0.006324 bits compared with the single common threshold of 0.24. When this optimal threshold was applied to the TIMIT `si/sx` test data, the resulting mutual information was 3.445719, an improvement of 0.003649 bits over the common threshold of 0.24. Given that optimizing thresholds for each phone provides at most a 0.14% improvement in mutual information on the training data, we concluded that there was little to gain over choosing a single threshold for all phones.

3.6 TIMIT keyword search experiments

In the following set of keyword search experiments, we seek to answer two questions. The first pertains to our proposed metric for selection of event threshold δ. Our criterion permits us to maximize mutual information between phone labels and phonetic events as a function of event threshold, but we have yet to correlate high mutual information with improved keyword search performance. Secondly, all previous work on the point process model did not address the size of the phonetic event set. In this work we have explicitly sought to minimize the number of phonetic events through the application of phonetic matched filters, thus we must investigate possible ramifications of an extremely sparse event set on keyword search performance.

For consistency with previous work, we have replicated the identical set of keyword

search experiments presented in [29] using the TIMIT corpus. The TIMIT corpus consists of 6300 sentences of phonetically-balanced, read speech and manually labeled, time-aligned phonetic transcripts [52]. Each of the 630 speakers recorded 2 identical dialect sentences (`sa` sentences) and 8 sentences drawn from a set of 2340 sentences (`si/sx` sentences). The `si/sx` portion of the corpus is split into training and test sets consisting of 3696 and 1344 sentences, respectively. The original TIMIT labeling differentiated 61 phonetic classes, however, consistent with many other phone recognition experiments, we operated on a reduced set of 48 phones as defined in [53]. For these experiments we used two distinct phone detectors to produce phone posteriorgrams, GMM and SMLP.

The GMM detectors were identical to those described in [29] and used standard 39-dimensional MFCC features based on 25 ms windows sampled every 10 ms. Feature vectors were further processed using cepstral mean subtraction and principal component diagonalization. For each phone $p \in \mathcal{P}$, the parameters of a full-covariance, mixture of 8 component Gaussian model were derived using the expectation maximization algorithm. The GMM models produced estimates of $\Pr(x|p)$, from which the phone detector posterior estimates $g_p(x) = \Pr(p|x)$ are computed using Bayes' rule where the prior phone probabilities $\Pr(p)$ are taken as the fraction of frames labeled p in the training set.

The SMLP-based phone detectors were trained as detailed in [49] using 39-dimensional PLP features estimated over 25 ms windows every 10 ms. Based upon psychophysical properties of hearing, PLP features provide a low dimensional representation of the speech signal with increased speaker independence while retaining linguistically relevant portions of the signal [6]. With SMLP detectors the original 61 TIMIT phonetic classes were mapped to a

reduced set of 49 classes based on the consolidated set of 48 phones listed in [53] with the addition of *oth* (i.e., "other") used as a garbage class.

For comparison, we replicated the toy experiments on TIMIT presented in [29] using local maxima and filtered events for GMM and SMLP posteriors extracted using the thresholds $\delta = 0.22, 0.26, 0.72, 0.81$ as indicated in Figure 3.6. Keyword model parameters and duration statistics for each of the 11 words in the TIMIT `sa1` training sentence were computed using transcriptions. The background model parameters were derived from 3696 `si/sx` type sentences because they were more phonetically balanced. As in the previous work, the test set consisted of 1512 sentences from `sa1`, `si` and `sx` test sentences. After applying the model, keyword detections were declared for local maxima of the keyword detection function $d_w(t)$ above threshold δ_w, and detections within 100 ms of the beginning of the keyword in the transcript were marked as correct. Multiple correct detections of the same keyword were discarded, and all other detections were recorded as false alarms. For the results listed in Table 3.1, we calculated average figure of merit (FOM), the mean detection rate given $1, 2, \ldots, 10$ false alarms per keyword per hour as the threshold δ_w was varied [45]. In another series of tests, we evaluated FOM as the number of training examples used to generate the keyword model was varied. The average FOM performance for the keywords in Table 3.1 plotted as a function of the number of keyword training examples is shown in Figure 3.8.

We observe that the use of filtered events resulted in 23% and 14% relative improvement in average FOM over local maxima for SMLP and GMM, respectively. Examining the mutual information in Figure 3.6, we also note that that peak mutual information is

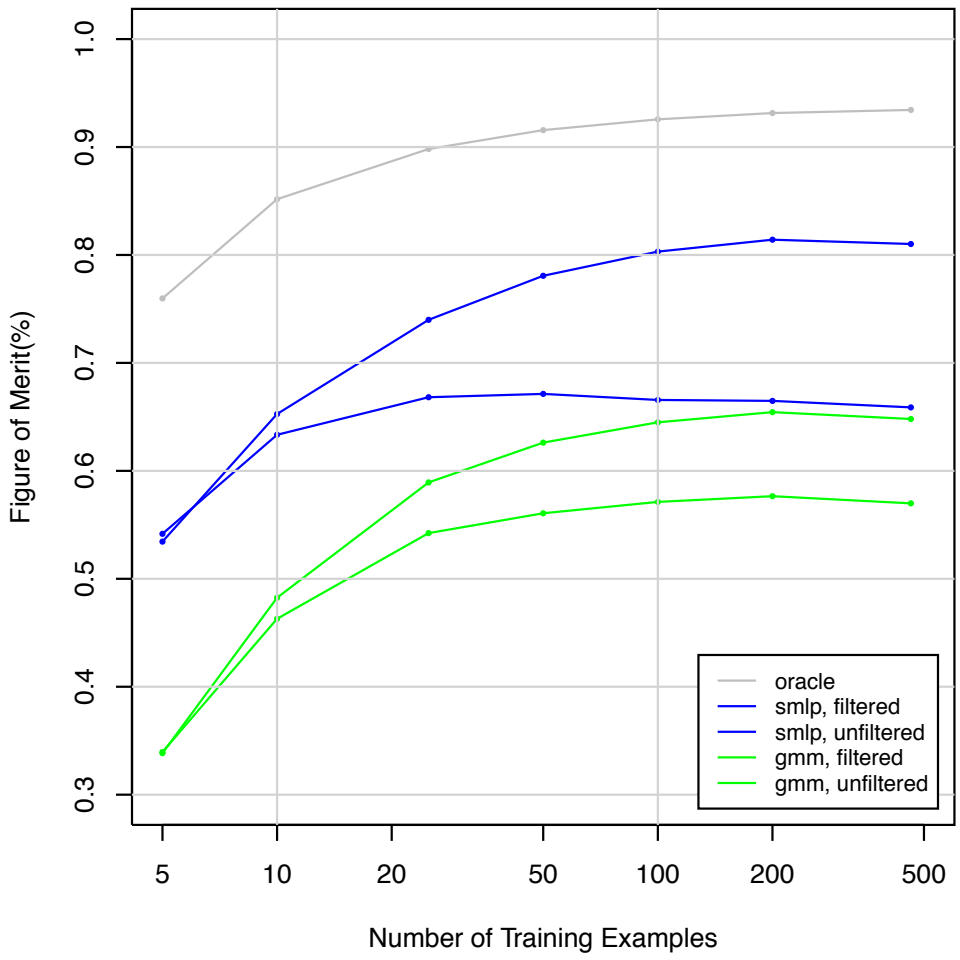

Figure 3.8: Average figure of merit vs. number of examples used in model construction for various TIMIT sa1 keywords using oracle, GMM and SMLP phonetic events.

highly correlated ($\rho = 0.9$) with average FOM in Table 3.1. In some instances better FOM results can be obtained by choosing a threshold *lower* than indicated by our metric. For instance using filtered SMLP events, FOM for "greasy" is 98.0 with $\delta = 0.10$ compared to 96.1 with $\delta = 0.22$. While decreasing threshold increases false alarms, we find that the keyword search model is much more sensitive to missing true events than false alarms.

Table 3.1: Average FOM for various TIMIT `sa1` keywords.

		filtered		local maxima	
keyword	oracle	SMLP	GMM	SMLP	GMM
had	87.2	66.8	54.9	57.1	51.6
dark	98.2	92.2	79.8	67.8	63.6
suit	84.1	67.7	53.9	44.6	38.2
greasy	99.2	96.1	87.6	88.3	89.2
wash	96.4	93.4	85.9	86.3	78.6
water	97.8	77.6	56.9	64.1	40.8
year	91.0	73.3	34.7	52.9	37.1
averages:	93.4	81.0	64.8	65.9	57.0

3.7 Conclusions

In this chapter we have considered methods of extracting phonetic events from phone posteriorgrams. Drawing on related work in [51], we applied phonetic matched filters to smooth posterior trajectories and introduced a mutual information based metric to determine appropriate thresholds for selecting events. It was previously demonstrated in [29] that Poisson process based keyword search models operate with performance comparable to traditional HMM-based keyword filler approaches while using a far sparser representation. In this this chapter we have demonstrated the use of phonetic matched filters to produce an even sparser set of events, reducing the event set by 40%, while simultaneously improving average keyword search performance by 23%. The event selection techniques introduced in this chapter are employed in all subsequent work in this dissertation. Furthermore, the finding that the minimal representation of speech consisting of only a single phonetic event per phone is sufficient for point process model keyword search facilitates parametric modeling presented in Chapter 4 and enhances keyword search speed in Chapter 6.

Chapter 4

Bayesian Approaches to Whole-Word Acoustic Modeling

Previous experiments with point process model keyword search identified a core limitation of the approach, namely the large numbers of keyword examples necessary to accurately estimate word model parameters. Indeed, the intrinsic advantages of whole-word acoustic modeling are frequently offset by the problem of data sparsity. To address this, we present several parametric approaches to estimating intra-word phonetic timing models. We present evidence that the distributions of phonetic event timing are well described by the Gaussian distribution. We explore the construction of models in the absence of keyword examples (dictionary-based models), when keyword examples are abundant (Gaussian mixture models), and also present a Bayesian approach which unifies the two. Applying these techniques in a point process model keyword search framework, we demonstrate a substantial improvement in performance for models constructed from few examples.

4.1 Background

Isolated word recognition systems in the early days of speech recognition were often constructed by modeling entire words. While practical for limited vocabulary size, the advent of large vocabulary systems based on hidden Markov models necessitated the use of subword units to enable the sharing of training examples across contexts and to permit the modeling of unseen words. However, if training examples are available, by maintaining the structure of the word, whole-word models have long been known to offer superior performance to subword-based systems [54].

The synthesis of words from subword units and the resulting geometric state duration distributions are partially responsible for the HMM's well-known deficiency in duration modeling. Additional constructs within the HMM framework such as segment models [55] have been introduced to address these shortcomings at the cost of increased complexity. As HMMs have been shown lacking in their ability to model duration, a large body of research has documented the importance of temporal cues in human speech perception [56].

Maximum a posteriori (MAP) approaches have been applied to HMM parameter estimation for purposes such as parameter smoothing and speaker adaptation [57]. Prior HMM parameter distributions based on context independent phone models can be used in the estimation of context dependent models. Likewise, speaker adaptation can be enhanced by using speaker independent prior models when speaker-specific data is limited. In both these cases, the prior is based on class-independent HMM parameter averages and MAP

estimation enables smoothed estimates of class-specific HMM parameters. Unlike HMM models, we are specifically modeling the timing of phonetic events, and our prior will take a very natural form derived from a word's phonetic composition.

In the point process model framework, whole-word models are characterized by an inhomogeneous Poisson process, and keyword detections are derived from the relative timing of a sparse set of phonetic events. Like other whole-word approaches, data sparsity is a problem. In previous PPM experiments in [29] and in Chapter 3, the Poisson rate parameters have been calculated using maximum likelihood estimation (MLE). As documented in [29], system performance depends on the accurate estimation of Poisson rate parameters which in turn requires large numbers of example keywords. In this chapter we confront the issue of data sparsity by introducing parametric models of phonetic event distributions. Using MAP estimation with simple dictionary-based prior distributions, we overcome the need for large amounts of training data and provide a seamless transition between subword and whole-word frameworks.

4.2 Word models based on maximum likelihood estimates

The defining features of the point process word model are the inhomogeneous Poisson process rate parameters which characterize the generation of phonetic events within a word. As detailed in Chapter 2, these rate parameters can be estimated from keyword examples using a maximum likelihood approach. Given N keyword training examples containing a total of $K_{p,d}$ phonetic events in the dth partition of phone p, the maximum likelihood

estimate of the inhomogeneous Poisson rate parameter $\lambda_{p,d}$ is given by

$$\lambda^*_{p,d} = \frac{K_{p,d}}{N\Delta T}. \tag{4.1}$$

The quality of the estimates of $\lambda_{p,d}$ and hence the quality of the resulting word model is a function of the number of training examples N, and we observe this characteristic in the performance of MLE-based point process models presented in Chapter 3. In Figure 3.8, note the rapid decay in average FOM when the number of keyword training examples falls below approximately 50 words. Interestingly, a comparable decrease in performance in models estimated from oracle phonetic events required only 20 keyword training examples. The plot in Figure 3.8 reflects the performance of keyword models based on a fixed number of training examples averaged over many random draws of training example sets. While the plot presents average performance, it does not reveal another significant shortcoming of the MLE-based approach: model variance. This characteristic was also remarked upon in [29] and it was suggested that parameterized models for the Poisson rate function $\lambda_p(t)$ could reduce the required number of model parameters and "provide more stability as we decrease the number of training examples."

4.3 Empirical distributions of phonetic events

To begin our investigation of suitable parametric models for phonetic events, consider Figure 4.1 which illustrates the distribution of phonetic events relative to normalized word duration derived from 462 examples of the keywords in the TIMIT dataset. Viewing the distribution of phonetic events suggests the possibility of modeling phonetic events using a Gaussian density. To qualify this intuition we present normal quantile-quantile (Q-Q)

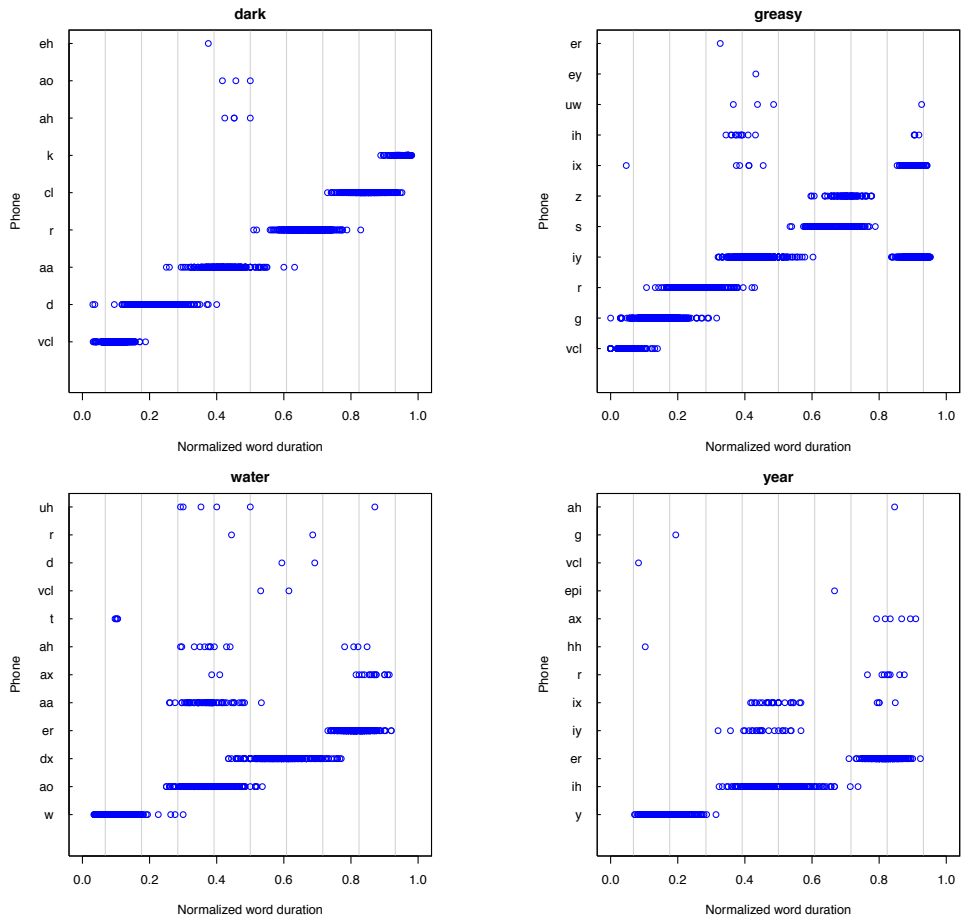

Figure 4.1: Distributions of phonetic events for 462 training examples of TIMIT words (a) dark, (b) greasy, (c) water, (d) year based on oracle phonetic events.

plots in Figure 4.2 for four TIMIT keywords comparing the empirical distribution with a Gaussian distribution. In these plots, empirical quantiles are depicted on the vertical axis and theoretical (Gaussian) quantiles on the horizontal. With the exception of the phone /k/ in "dark," we observed that the distributions of phonetic events are reasonably well modeled by the Gaussian distribution.

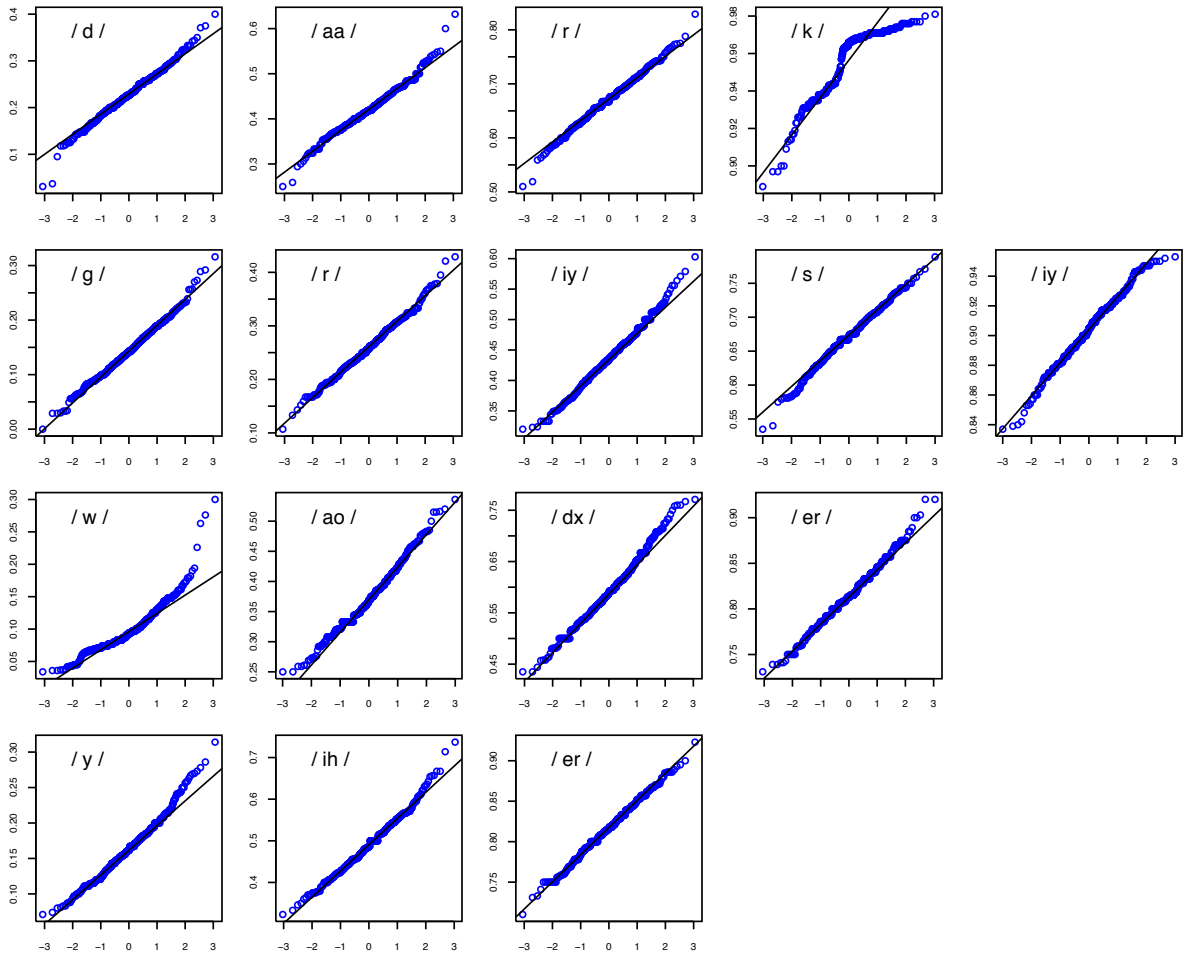

Figure 4.2: Normal Q-Q plots of phonetic timing distributions for TIMIT words (a) dark, (b) greasy, (c) water, (d) year showing approximate normality of timing distributions. Data quantiles are shown on the vertical axis and theoretical quantiles on the horizontal axis.

4.4 Word modeling based on Gaussian mixtures

In light of the normal Q-Q plots in Figure 4.2, an obvious choice to parametrically describe phonetic timing distributions is the Gaussian mixture model (GMM). In the MLE-based word model, each phone $p \in \mathcal{P}$ necessitated the estimation of D inhomogeneous rates parameters $\lambda_{p,d}$. For typical applications, $|\mathcal{P}| = 48$ phone classes and $D = 10$ word divisions which requires the estimation of 480 parameters. Although a significant fraction

these parameters are zero, construction of a representative distribution demands a large number of word examples as apparent in performance plots of MLE-based models found in Figure 3.8. Ideally, the use of Gaussian modeling should reduce the number of parameters to just two parameters per phone in a word's dictionary (phonetic) form. Additionally, the Gaussian density imposes strong constraints on the shape of estimated distributions.

4.4.1 Computing Poisson rate parameters using parametric distributions of phonetic events

We now address the method by which rate parameters are calculated given a parametric distribution of the phonetic events. We will consider the estimation of the inhomogeneous Poisson rate parameter $\lambda_{p,d}$ corresponding to phone p in the dth partition of normalized word duration. Since the following derivation is valid for any phone, we will drop the p subscript for the remainder of this section to simplify the notation. As noted previously, the maximum likelihood estimate λ_d^* is given by

$$\lambda_d^* = \frac{K_d}{N\Delta T}. \tag{4.2}$$

In the expression for λ_d^*, the total count of phonetic events in the interval d for N word examples is given by $K_d = X_d^1 + X_d^2 + \cdots + X_d^N$ where X_d^n is the count of events in interval d for example n. Note that X_d^n are independent, identically distributed Poisson random variables drawn from Poisson($\lambda(t)$), the true but unknown distribution for phone p with the continuous-valued rate parameter $\lambda(t)$. It follows that λ_d^* is also a random variable and its

expected value is given by

$$\begin{aligned} \mathrm{E}[\lambda_d] &= \mathrm{E}\big[\frac{K_d}{N\Delta T}\big] \\ &= \mathrm{E}\big[\sum_{n=1}^{N} \frac{X_d^n}{N\Delta T}\big] \\ &= \frac{1}{N\Delta T}\,\mathrm{E}\big[\sum_{n=1}^{N} X_d^n\big] \\ &= \frac{1}{\Delta T}\,\mathrm{E}\big[X_d^n\big] \\ &= \frac{1}{\Delta T}\int_{t_d}^{t_{d+1}} \lambda(\tau)d\tau \end{aligned} \tag{4.3}$$

where the limits of the dth partition are given by t_d and t_{d+1}. Therefore, for any parametric description of $\lambda(t)$, instead of using observed counts of phonetic events to calculate λ_d^*, we can determine λ_d directly by simply integrating $\lambda(t)$.

4.4.2 Estimation of GMM-based word model parameters

We calculate a Gaussian mixture to model the inhomogeneous Poission rate function $\lambda_p(t)$ for each phone p using phonetic event data extracted from N length-normalized example words. This is an unsupervised process, and we first determine an appropriate number of mixtures by performing k-means clustering of the phonetic events for each phone. The number of clusters k should reflect the notion that each phone instance within a word can be modeled by a single Gaussian (i.e., the phone /iy/ in "greasy" should be modeled using $k = 2$ Gaussians). While our clustering is *not* informed by the word's dictionary form, we can encourage clustering consistent with this idea by allowing the number of clusters k to incrementally grow as long as successive cluster means are separated by roughly 4 standard deviations. With the number of mixtures k determined, we then employ expectation-maximization to obtain the mean, variance and mixture coefficients for

each phone GMM.

While this model does reflect the distribution of events in time, ultimately we will be using it to compute the *expected counts* of phonetic events. We are really modeling the rate parameter functions $\lambda_p(t)$ which do not share the constraints of a true probability density function. For example, the GMM for the phone /iy/ should have a bimodal shape reflecting the two instances of /iy/ in the word's phonetic form. In a generative sense, if we were drawing samples of the word "greasy" using a collection of GMMs corresponding to each phone, we would need a mechanism to ensure that we draw from the GMM for /iy/ twice. Therefore, it was necessary to weight the component distributions relative to the total number of keyword examples from which the phonetic events were drawn. Thus, if we observe n_p phonetic events from n_w keyword examples, applying a scale factor of n_p/n_w to the GMM for phone p allows us to correctly compute expected counts. To illustrate this point, the GMM-based model for the word "greasy" shown in Figure 4.3 was estimated from 462 training examples. From these keyword instances, we observed 793 phonetic events for the phone /iy/ which clustered into two groups of size 451 and 342 centered at 0.43 and 0.90, respectively (see Figure 4.3). Thus, the mixture of two Gaussians for phone /iy/ are weighted relative to the 462 examples resulting in mixture weights of 0.976 and 0.740.

GMM-based models for selected TIMIT keywords are shown in Figure 4.3. Note that the models capture both speaker pronunciation variation and phone detector confusions. The models depicted in this figure are based on all 462 training examples. For a fixed number of word examples, when multiple models are estimated using different samples, the variation across models will increase as sample size decreases. However, given that

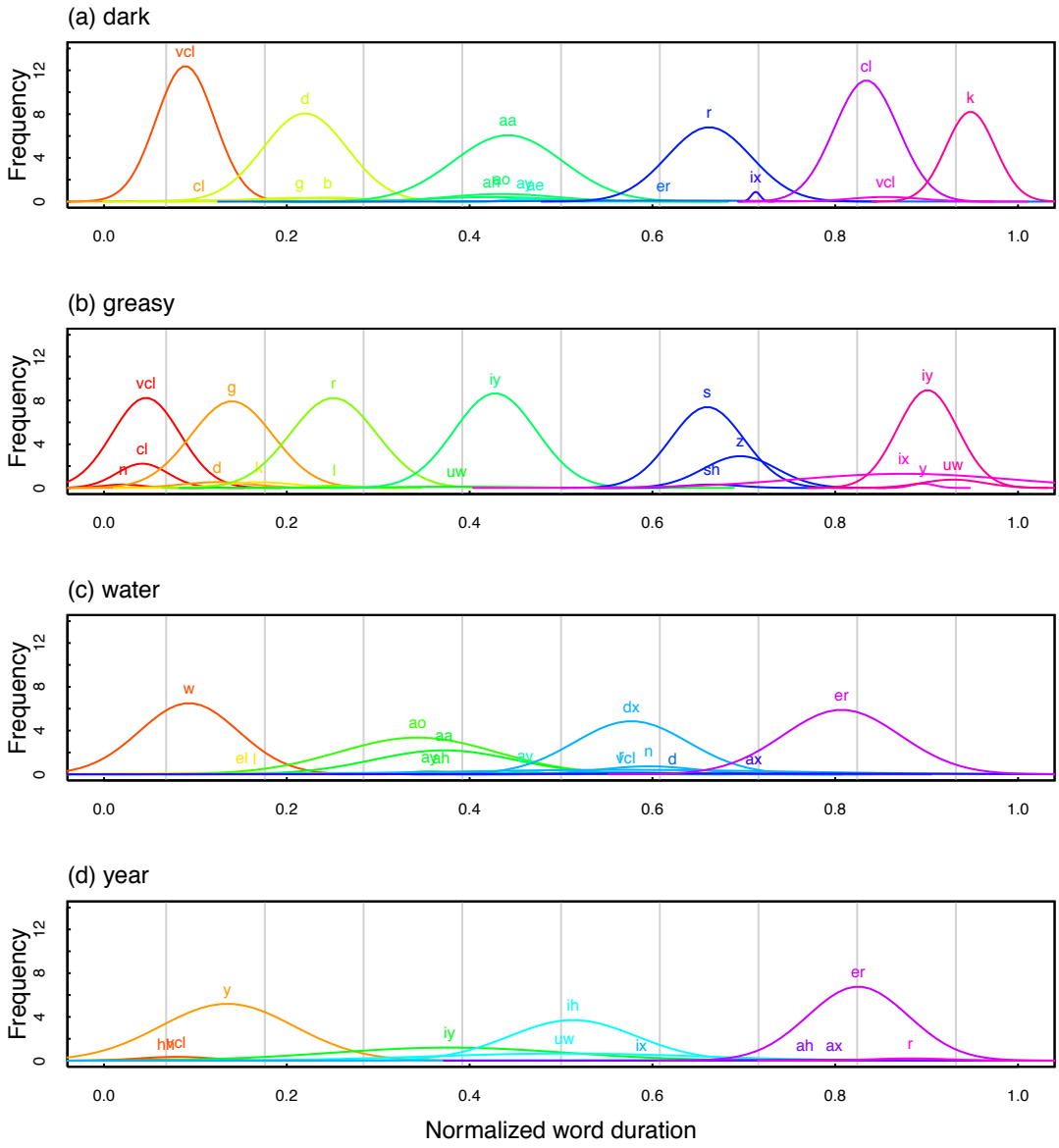

Figure 4.3: GMM-based phone timing distributions for TIMIT keywords (a) dark, (b) greasy, (c) water, (d) year, each estimated using 462 keyword examples.

each Gaussian requires the estimation of only 2 parameters, we would expect GMM-based models to exhibit less variation than is observed when estimating 10 parameters per phone as is typically the case with MLE estimation of rate parameters.

4.5 Dictionary-based models

MLE and GMM-based models are both derived entirely from keyword examples with no prior assumptions about the phonetic composition of the word. Such models result in good keyword search performance when training examples are plentiful, but they suffer when only a few examples are available and fail completely for words with no training examples. In the absence of any actual keyword examples, we can intuit much of the structure of a word's phone distribution solely from the word's dictionary (phonetic) form. Without any word examples, we can construct a naive *dictionary* model by assigning a single Gaussian to each phone in the dictionary form with equally spaced means μ and a fixed standard deviation σ. Such a models for selected TIMIT keywords are depicted in Figure 4.4 with $\sigma = 0.05$.

4.5.1 Dictionary-based models incorporating phone confusions

Comparing the GMM-based models in Figure 4.3 and the dictionary models in Figure 4.4, an obvious shortcoming of the dictionary model is its inability to accommodate pronunciation variation and likely phone confusions. Variation which arises from different speaker productions could be incorporated using weighted combinations of alternate dictionary forms. Lacking this information, a very simple alternative is to apply phone

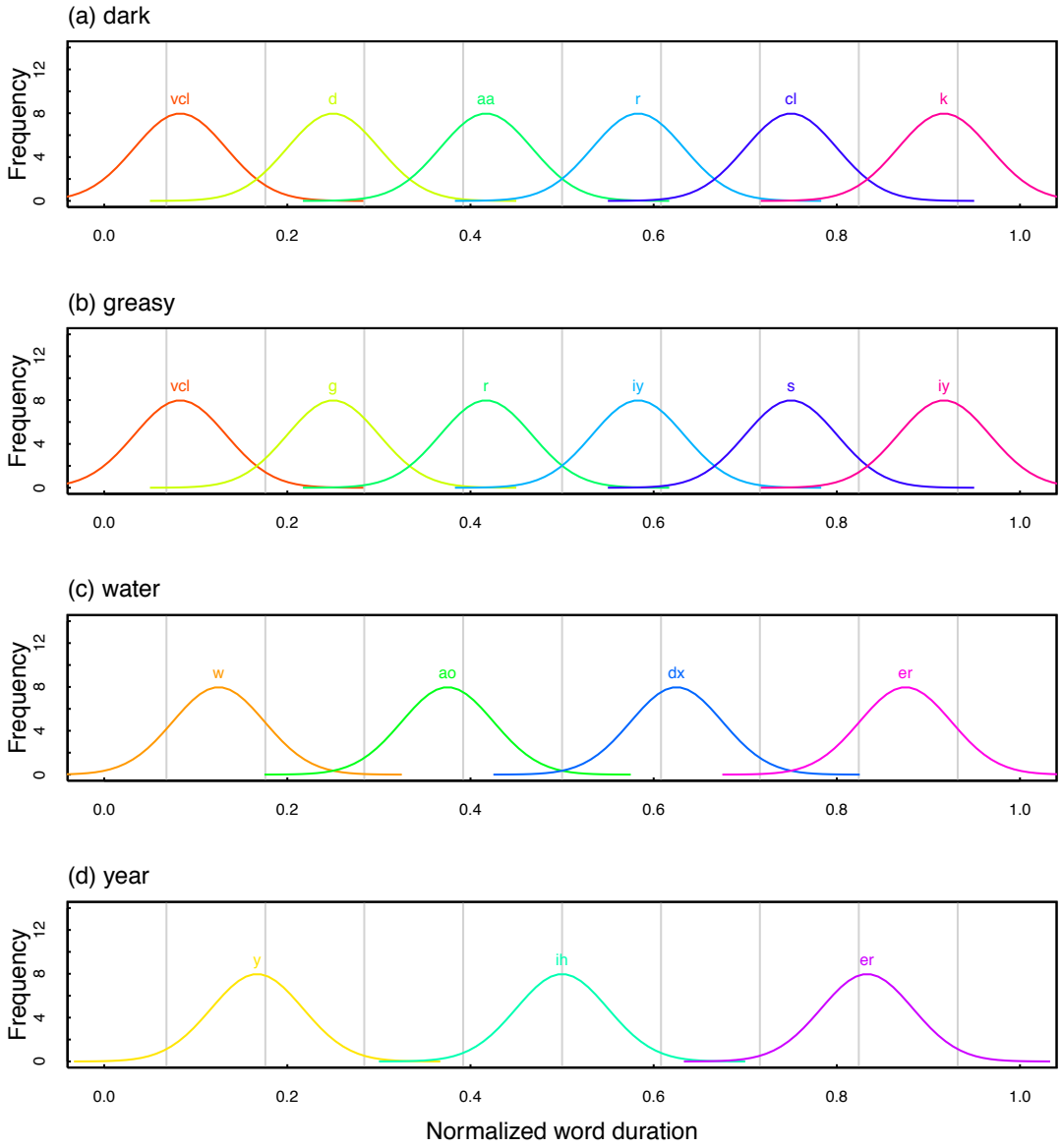

Figure 4.4: Simple dictionary-based phone timing distributions for TIMIT keywords (a) dark, (b) greasy, (c) water, (d) year. Each phone distribution has a fixed standard deviation $\sigma = 0.05$.

confusion matrix data associated with the phone detectors. If rows of the confusion matrix correspond to actual phone classes and columns correspond to predicted phone classes, then each matrix element $C_{ij} = \Pr(p_j|p_i)$. An example confusion matrix is shown in Figure 4.5.

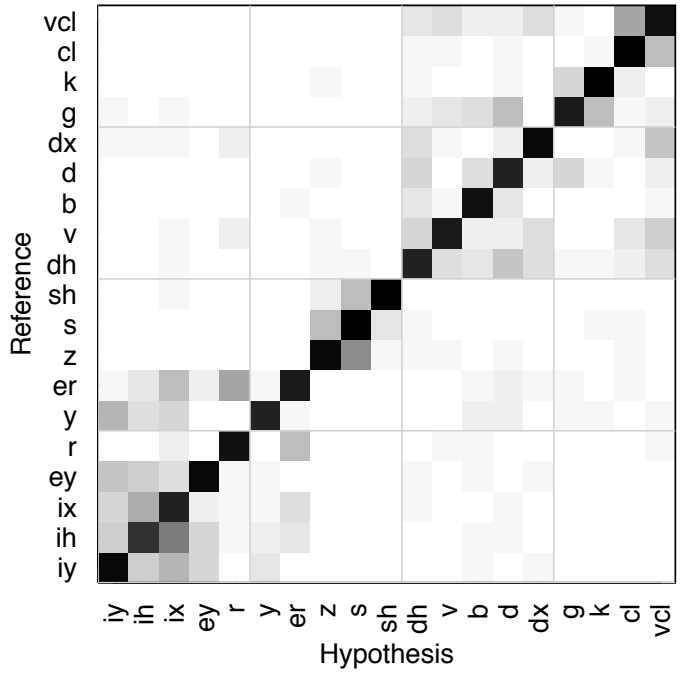

Figure 4.5: Example phonetic event confusion confusion matrix based on filtered SMLP posteriorgrams with $\delta = 0.22$. Matrix elements $C_{ij} = \Pr(p_j|p_i)$ where darker color represent higher probability.

There are a few possible ways to estimate $C_{ij} = \Pr(p_j|p_i)$. We could compute it directly given per frame likelihoods from phone posteriorgrams and the corresponding phonetic labels. However, using frame-by-frame probabilities does not reflect the discrete phonetic events of this model. Another option which is consistent with phonetic events is to use the fractional count matrix presented Section 3.5 and depicted in Figure 3.5. The fractional counts are used as proxies for estimating each $\Pr(p_j|p_i)$.

Next, we introduce the likely confusions into our dictionary model in the following manner. For each phone appearing in a word's dictionary form, the mixture coefficient is re-weighted by the confusion matrix diagonal element $C_{ii} = \Pr(p_i|p_i)$. For phones which are confusable (i.e., $\Pr(p_j|p_i) > 0$ for $i \neq j$), we introduce new Gaussian with with mixture weight C_{ij} but the same mean μ_i and standard deviation σ. The resulting models are depicted in Figure 4.6.

4.6 Bayesian modeling of phonetic event distributions

We have presented model construction at two ends of the spectrum: a model assembled without examples (dictionary-based) and an efficient parametric model built entirely from data (GMM). The significant shortcoming of the example-based approaches stems from the absence of constraints guiding the estimation of the parameters despite the fact phonetic events within words are strongly governed by the word's phonetic form. While dictionary-based models are a poor approximation of the true underlying phonetic event distributions, they can serve as reasonable, informative *prior distributions* for estimating the true parameters. These facts strongly suggest the use of a Bayesian approach to Poisson rate parameter estimation which permits the creation of reasonable models when few word examples are available and provides a mechanism for adapting to additional training data.

Before describing our Bayesian approach in detail, consider first the two limiting cases as depicted in Figure 4.7 for the TIMIT keyword "greasy." The distributions depicted in Figure 4.7a are based upon the word's dictionary form and likely phone confusions and this represents our prior knowledge absent actual word examples. The distributions shown

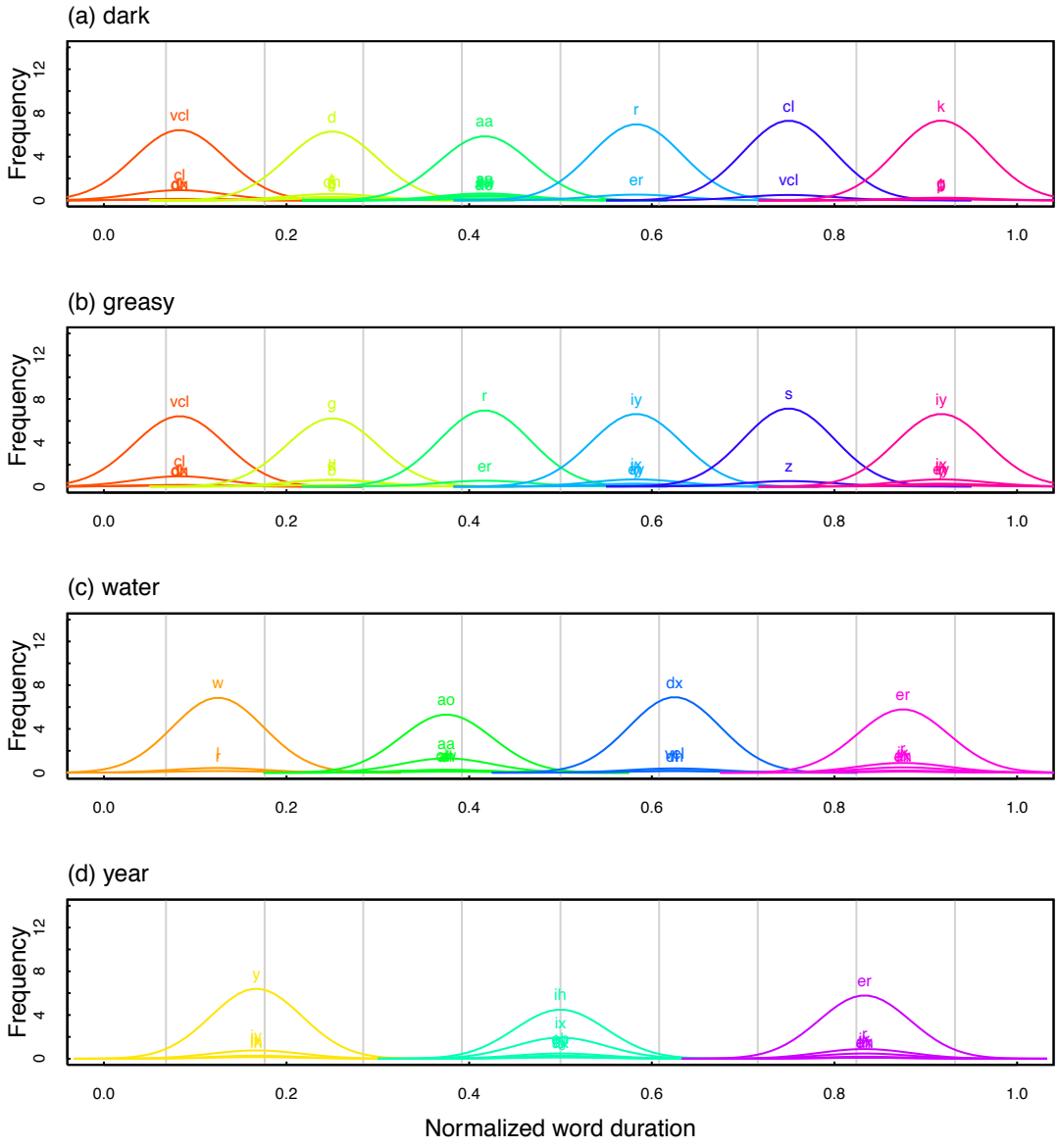

Figure 4.6: Dictionary-based phone timing distributions incorporating phonetic confusions for TIMIT keywords (a) dark, (b) greasy, (c) water, (d) year. Each phone distribution has a fixed standard deviation σ =0.05.

in 4.7b are based on GMM models estimate using 462 keyword examples. The contrast between these extremes illustrates two requirement for our Bayesian approach. First, we must model the change the location and scale of the individual phone distributions as illustrated by the change in the mean μ and variance σ^2 of the phone /g/ in Figure 4.7. Second, the model should adapt the mixture weights to the phonetic variation observed in training examples as can be seen in the phone /z/.

4.6.1 Bayesian estimation of unknown mean and variance

As presented in Section 4.3, phonetic event timing distributions can be reasonably well described by Gaussian distributions. Therefore, the basis of our approach is standard Bayesian inference for the Gaussian distribution where both the mean μ and precision $\lambda \triangleq 1/\sigma^2$ are unknown. As derived in [58, 59], the conjugate prior is given by the normal-gamma distribution:

$$\begin{aligned}
\mathrm{NG}(\mu, \lambda | \mu_0, \kappa_0, \alpha_0, \beta_0) &\triangleq \mathcal{N}(\mu | (\kappa_0 \lambda)^{-1}) \, \mathrm{Gam}(\lambda | \alpha_0, \beta_0) \\
&= \frac{1}{Z_{NG}(\mu_0, \kappa_0, \alpha_0, \beta_0)} \lambda^{\frac{1}{2}} \exp\Big(-\frac{\kappa_0 \lambda}{2} (\mu - \mu_0)^2\Big) \lambda^{\alpha_0 - 1} \exp^{-\lambda \beta_0} \\
&= \frac{1}{Z_{NG}(\mu_0, \kappa_0, \alpha_0, \beta_0)} \lambda^{\alpha_0 - \frac{1}{2}} \exp\Big(-\frac{\lambda}{2} \Big[\kappa_0 (\mu - \mu_0)^2 + 2\beta_0\Big]\Big)
\end{aligned}$$

where the normalizing factor Z_{NG} is defined as

$$Z_{NG}(\mu_0, \kappa_0, \alpha_0, \beta_0) = \frac{\Gamma(\alpha_0)}{\beta_0^{\alpha_0}} \left(\frac{2\pi}{\kappa_0}\right)^{\frac{1}{2}}$$

The normal-gamma prior is specified four hyperparameters μ_0, κ_0, α_0, and rate β_0 which describe the distributions of μ and λ. The prior marginal distribution of precision λ is a

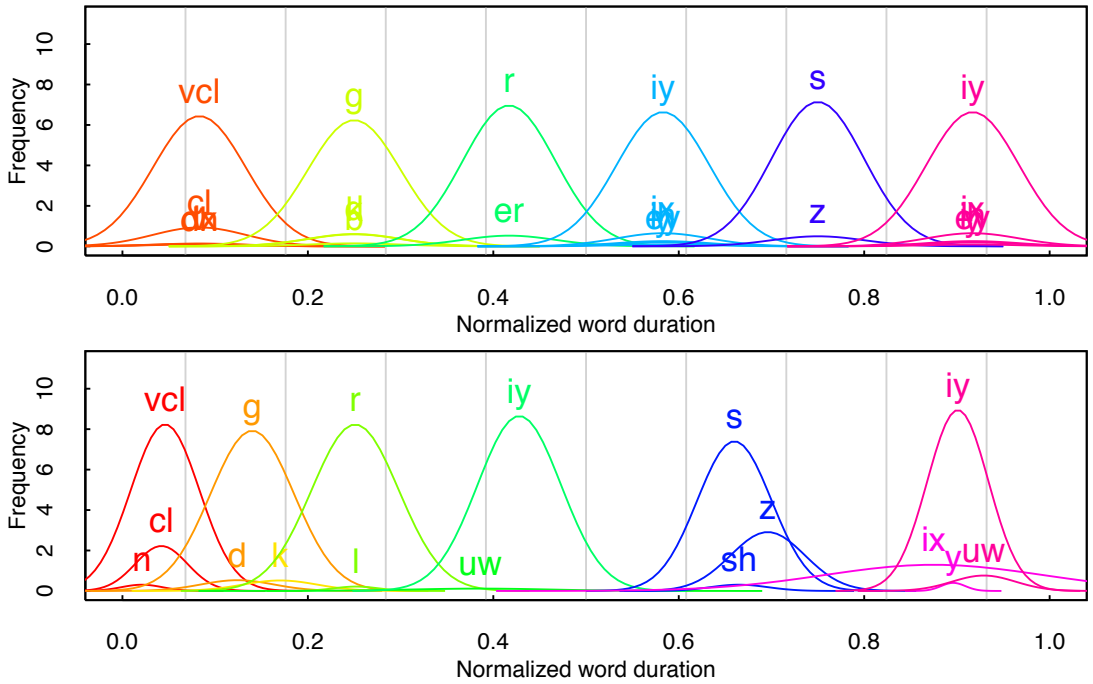

Figure 4.7: Limiting cases for Bayesian estimation of phonetic timing distributions for TIMIT keyword "greasy." (a) Prior model based on dictionary form with likely phone confusions, (b) GMM model estimated from 462 word examples.

$\mathrm{Gam}(\lambda|\alpha_0, \beta_0)$ distribution and its mean and variance are given by

$$\mathrm{E}[\lambda] = \frac{\alpha_0}{\beta_0} \tag{4.4}$$

$$\mathrm{Var}[\lambda] = \frac{\alpha_0}{\beta_0^2}, \tag{4.5}$$

reflecting our prior uncertainty about the variance of a phone distribution. The prior marginal distribution of μ is a Student's t-distribution, $T_{2\alpha_0}(\mu|\mu_0, \beta_0/(\alpha_0\kappa_0))$, for which mean and variance are given by

$$\mathrm{E}[\mu] = \mu_0 \tag{4.6}$$

$$\mathrm{Var}[\mu] = \frac{\beta_0}{\kappa_0(\alpha_0 - 1)}. \tag{4.7}$$

As shown in [58,59], after observing data $D = (x_1, x_2, \ldots, x_n)$ with data likelihood $p(D|\mu, \lambda)$, the posterior distribution for μ and λ has the form

$$\begin{aligned} p(\mu, \lambda|D) &\propto NG(\mu, \lambda|\mu_0, \kappa_0, \alpha_0, \beta_0)\, p(D|\mu, \lambda) \\ &\propto \lambda^{\frac{1}{2}} \lambda^{\alpha_0 + \frac{n}{2} - 1} \exp(-\beta_0\lambda) \exp.\Big(-\frac{\lambda}{2}\left[\kappa_0(\mu - \mu_0)^2 + \sum\nolimits_i (x_i - \mu)^2\right]\Big) \\ &\propto NG(\mu, \lambda|\mu_n, \kappa_n, \alpha_n, \beta_n) \end{aligned} \tag{4.8}$$

The posterior hyperparameters $\mu_n, \kappa_n, \alpha_n, \beta_n$ can be expressed as

$$\mu_n = \frac{\kappa_0\mu_0 + n\bar{x}}{\kappa_0 + n} \tag{4.9}$$

$$\kappa_n = \kappa_0 + n \tag{4.10}$$

$$\alpha_n = \alpha_0 + n/2 \tag{4.11}$$

$$\beta_n = \beta_0 + \frac{1}{2}\sum_{i=1}^{n}(x_i - \bar{x})^2 + \frac{\kappa_0 n(\bar{x} - \mu_0)^2}{2(\kappa_0 + n)} \tag{4.12}$$

Initial values of the hyperparameters are determined from our prior estimates of the mean and variance of μ and λ. Since the precision λ is gamma distributed, α_0 and β_0 are determined using the relations in Equations (4.4) and (4.5). The random variable μ is t-distributed with $2\alpha_0$ degrees of freedom, location parameter μ_0, and precision parameter $\kappa_0\alpha_0/\beta_0$. Thus, with α_0, β_0 and Var$[\mu]$, Equation (4.7) allows us to compute κ_0. Given the form of the posterior distribution of μ and λ in Equation (4.8), once we have observed data $D = (x_1, x_2, \ldots, x_n)$, we can directly compute the posterior hyperparameters as shown by Equations (4.9)–(4.12). Finally, using the hyperparameters of posterior, we can easily extract maximum a posteriori (MAP) estimates for μ and λ.

A graphical model representation of the generation of phonetic event timing is depicted in Figure 4.8. The observed variable t represents the phonetic event time which is drawn from the $\mathcal{N}(t|\mu, 1/\lambda)$ distribution where μ and λ are unobserved random variables.

T ɔution with

as rd would be

g

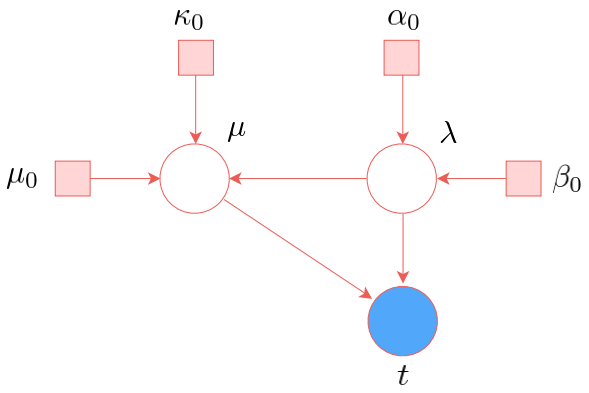

Figure 4.8: Graphical model representation depicting the process of generating a phonetic event time $t \sim \mathcal{N}(t|\mu, 1/\lambda)$ where parameter μ and λ are drawn a normal-gamma prior distribution with associated hyperparameters.

4.6.2 Selection of normal-gamma hyperparameters

The hyperparameters α_0, β_0, κ_0, and μ_0, reflect our confidence in prior estimates of μ and λ and determine how rapidly the posterior distribution adapts to the introduction of new data observations. In this section we discuss our approach to hyperparameter initialization.

The most straightforward of the four hyperparameters is μ_0, the prior expected value of μ. We have chosen the prior mean μ_0 just as described in Section 4.5 on dictionary-based models using the word's canonical dictionary form. In the dictionary model we fixed the standard deviation of each Gaussian to $\sigma = 0.05$ which was chosen after considering a range values: 0.025, 0.05, and 0.10. Setting $\sigma = 0.05$ implies that for precision λ, the expected value $\mathrm{E}[\lambda] = 400$. The gamma distribution governing the random variable λ is fully specified by two hyperparameters, α_0 and β_0, which are related to moments of λ in Equation (4.4) and (4.5). In order to reduce the number of free parameters, we coupled α_0 and β_0 by introducing the variable ρ and setting $\mathrm{Var}[\lambda] = \rho\,\mathrm{E}[\lambda]$. Increasing ρ corresponds to an increase in the prior uncertainty over the value of precision. After considering a range of values (0.1, 0.5, and 0.9), we found the estimation of λ to be fairly insensitive to the choice of ρ and subsequently used $\rho = 0.5$ for all experiments. Thus, for a fixed valued of ρ, α_0 and β_0 are determined by specifying $\mathrm{E}[\lambda]$.

The final hyperparameter to be considered, κ_0, corresponds to the *equivalent sample size* of the prior. As indicated by Equation (4.7), κ_0 is inversely proportional to the variance of μ, so rather than choosing a prior value for $\mathrm{Var}[\mu]$ it is more intuitive to set $\kappa_0 = 1$. This is akin to saying that our prior carries the same weight as a single observation.

This notion is also consistent with posterior parameter κ_n in Equation (4.10).

4.6.3 Example of the MAP estimation process

To illustrate the steps previously described, we present the following example of computing the MAP estimate of the distribution corresponding to the phone /g/ in the TIMIT keyword "greasy." Using the simple dictionary model depicted in Figure 4.4, we observe the prior mean $\mu_0 = 0.25$ and standard deviation $\sigma = 0.05$. For this value of σ, the expected value of precision $\mathrm{E}[\lambda] = 400$ and with $\rho = 0.5$, it follows that $\alpha_0 = 4.0$ and $\beta_0 = 0.01$. Together with $\mu_0 = 0.25$ and $\kappa_0 = 1$, these hyperparameters yield the conjugate prior distribution plotted in Figure 4.10. After having observed 16 examples of the keyword and extracted the phonetic events shown in Figure 4.9, we can calculate the posterior updated hyperparameters using Equations (4.9)–(4.12). From this data we obtain $\alpha_n = 12.0$, $\beta_n = 0.031$, $\mu_n = 0.131$ and $\kappa_n = 17$ where $n = 16$ examples, and the resulting posterior distribution is shown in Figure 4.10. From this posterior distribution, it follows that the MAP estimates $\mu' = \mu_n = 0.131$ and $\lambda' = (\alpha_n - \frac{1}{2})/\beta_n = 370$.

4.6.4 Bayesian estimation of mixture coefficients

We have provided a Bayesian approach to deriving estimates of phone distribution mean and variance, but another critical element is the estimation of mixture weights in

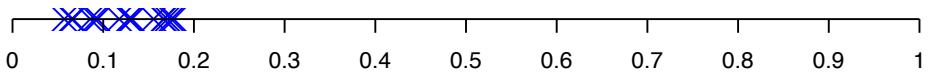

Figure 4.9: Phonetic event data observations for phone /g/ based on 16 examples of keyword "greasy."

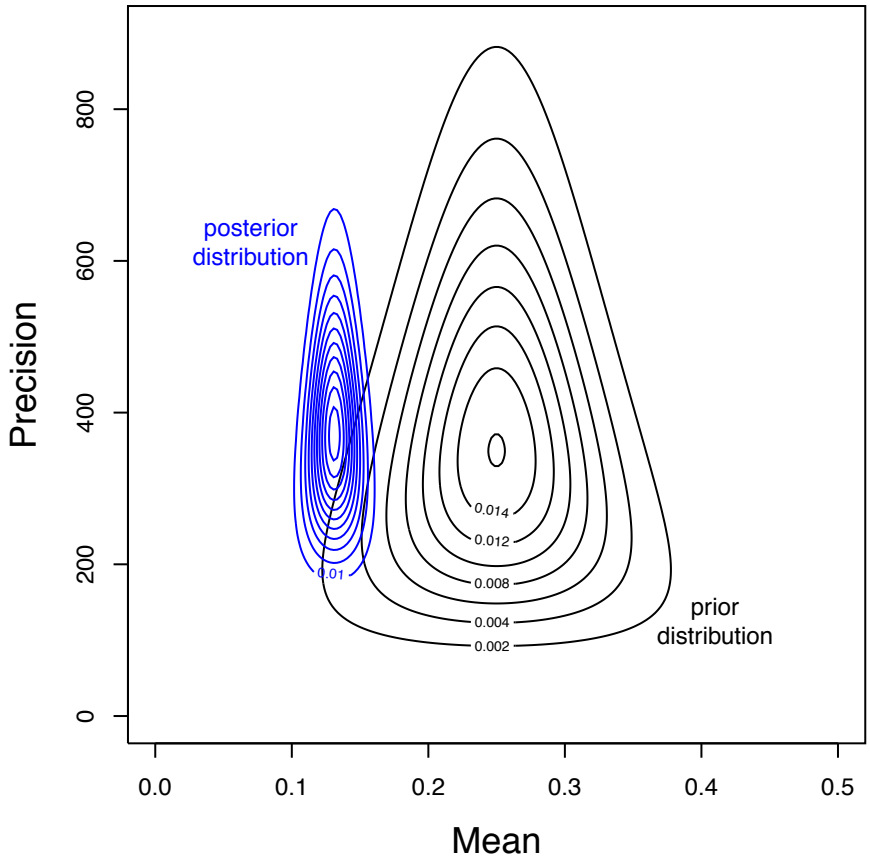

Figure 4.10: Normal-gamma prior and posterior distributions for the example of phone /g/ in "greasy." The conjugate prior distribution is specified by $\alpha_0 = 4.0$, $\beta_0 = 0.01$, $\mu_0 = 0.25$ and $\kappa_0 = 1$. After observing phonetic events from 16 keyword examples in Figure 4.9, the updated hyperparameters are $\alpha_n = 12.0$, $\beta_n = 0.031$, $\mu_n = 0.131$ and $\kappa_n = 17$ resulting in the normal-gamma posterior distribution shown.

order to account for pronunciation variation and phone detector errors. As a prior estimate of the mixture coefficient, we apply the same idea used in the dictionary models pictured in Figure 4.6 where each distribution is weighted by the corresponding phone confusion matrix element. As the number of keyword examples increases, our model should asymptotically approach the mixture weights of GMM-based models.

We can explain the production of phonetic events using the following generative story: a phonetic event is the result of two independent random variables, the first being a

Bernoulli random variable which dictates whether the event is actually observed. The second is a Gaussian random variable which specifies the event's time of occurrence within a word. In our Bayesian treatment, the parameter π associated with the Bernoulli random variable *is itself a random quantity* with conjugate prior distribution $\text{Beta}(\pi|a,b)$ as illustrated in Figure 4.11. Under this distribution, $\text{E}[\pi] = a/(a+b)$ and the sum $a+b$ constitutes the *effec*

we s

choic

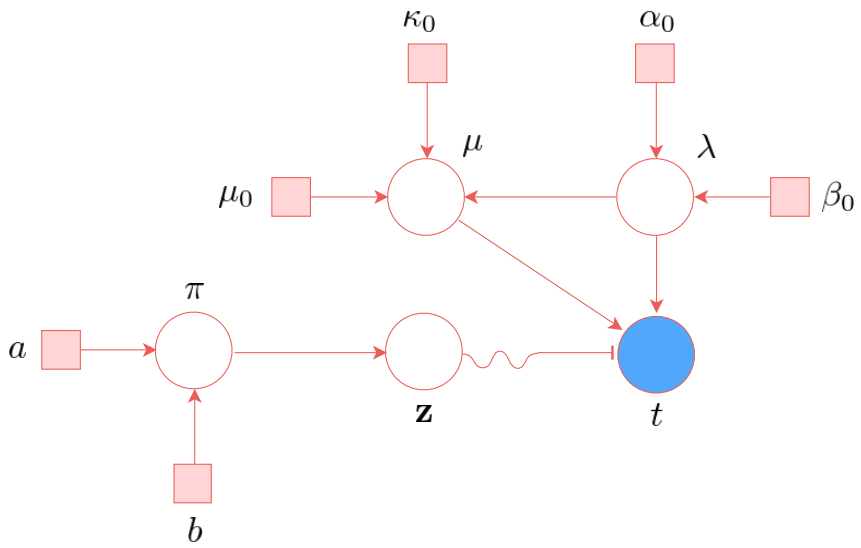

Figure 4.11: Graphical model representation depicting the process of generating a phonetic event from two independent processes. Event time t is generated as previously described in Section 4.6.1. Event occurrence is governed by random variable $z \sim \text{Bernoulli}(\pi)$ where π is drawn from conjugate prior $\text{Beta}(a,b)$ distribution.

We will now describe the posterior update. If we have a total of n keyword examples in which a phonetic event for the phone we are modeling is present in m cases and absent in $l = n - m$ cases, then the posterior distribution of π will be $\text{Beta}(\pi|a+m, b+l)$. Thus, we will take the mean of the posterior distribution $(a+m)/(a+b+n)$ as the posterior

updated value of the parameter π'. However, other cases are more complex. Words in which a phone is repeated (e.g., /iy/ in "greasy") require a distinct π and prior distribution for each instance of the phone in the word. Next, after having observed word examples, we must decide which phonetic events are associated with which distributions. To handle this situation, we performed k-means clustering and assigned phonetic event examples according to their cluster index.

4.6.5 Bayesian model example for "greasy"

An illustration of how the Bayesian model for "greasy" evolves as we increase the number of keyword examples is shown in Figure 4.12. For the case $n = 0$ examples, the Bayesian model is identical to the prior dictionary model with phonetic confusions in Figure 4.6b. Likewise at $n = 462$ examples, the Bayesian model largely mirrors the GMM-based model in Figure 4.3 with a few minor differences such as the handling of the phone /ix/. As the number of examples increases, the mean of each Gaussian shifts rapidly towards its limiting value and the variance contracts. Additionally, we observe the development of pronunciation variation (/s/ with /z/ and final /iy/ with /ix/).

4.7 Experiments

In this chapter we have proposed several approaches to modeling phonetic event distributions, and in this section we apply these techniques to constructing point process models for keyword search, specifically in the estimation of inhomogeneous Poisson rate parameters. In previous work on point process models, MLE rate parameter estimates were

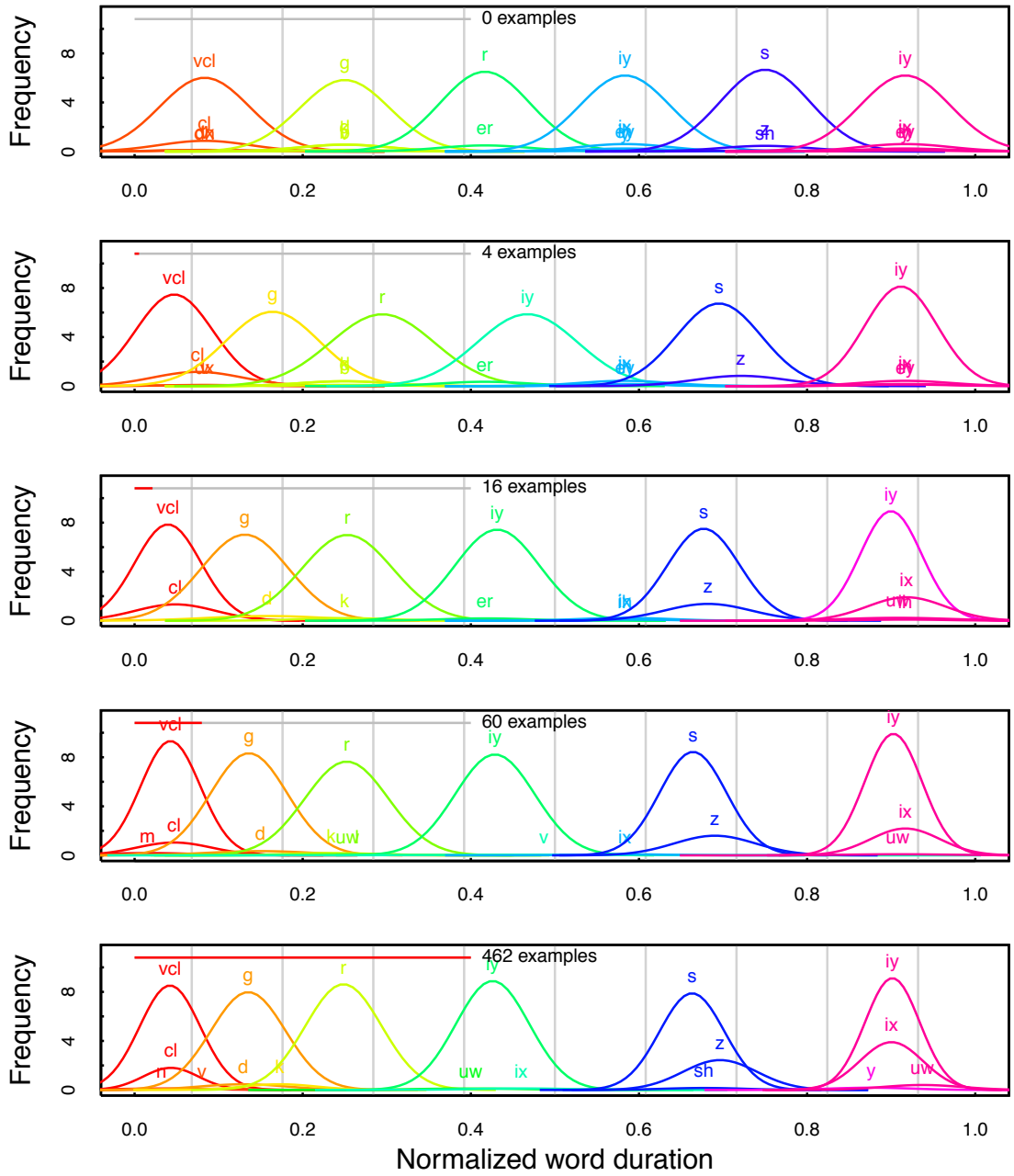

Figure 4.12: Bayesian estimated phone timing models for the keyword "greasy" constructed using various numbers of examples.

derived from the counts of events in each word subdivision. As detailed in Section 4.4.1, given any model of the phonetic event distribution, we can simply replace the "hard" counts with *expected counts* under the model. The specific aim of our work on parametric modeling

was to significantly improve keyword search performance when keyword examples are limited. We conducted experiments using TIMIT as in Chapter 3 and we also present results on the Wall Street Journal (WSJ) corpus.

4.7.1 TIMIT experiments

Keyword search experiments were first conducted on the TIMIT database in the same manner previously detailed in Section 3.6. PLP acoustic features were transformed into phone posteriorgrams using a sparse multilayer perceptron based system from [49]. Posteriorgrams were then converted into phonetic events by applying phonetic matched filters to the posterior trajectories and selecting local maxima above a threshold $\delta = 0.22$. Given previous results of Section 3.6, we did not consider posterior data derived from GMM-based phone detectors, nor did we consider "local maxima" phonetic events (see Section 3.3).

The TIMIT corpus provides numerous examples of selected keywords which allowed us to evaluate model construction as a function of sample size. In Figure 4.13 we show keyword search performance measured using average figure of merit (FOM) as a function of the number of training examples used in model construction. The plot for each keyword includes results from each of the parametric modeling approaches as well as the original MLE-based model. Our test keywords were necessarily limited to the few sufficiently long TIMIT `sa1/sa2` words. The dictionary-based model with phone confusions does not depend on any keyword examples, thus its performance is constant. The results depicted for GMM, MLE and Bayesian models represent the mean performance of many models based on random draws of examples.

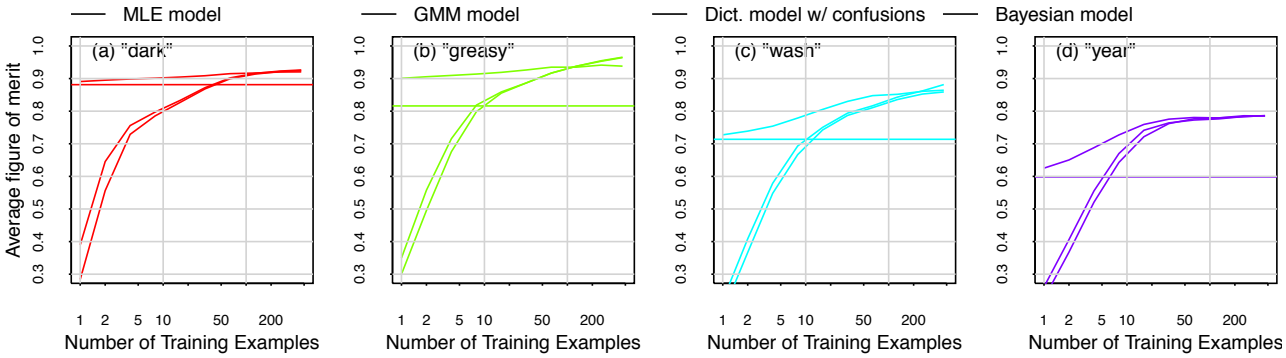

Figure 4.13: Average figure of merit vs. number of examples used in model construction for various TIMIT sa1 keywords.

4.7.2 WSJ experiments

The TIMIT dataset consists of roughly 8 hours of speech. In order to explore point process model performance on a larger scale with more keywords, we conducted a series of keyword search experiments using the Wall Street Journal (WSJ0 and WSJ1) datasets. The WSJ corpora consist of readings from newspaper articles recorded with high quality microphones [60]. Phonetic alignments for WSJ0/1 totaling 35270 utterances and 76 hours of speech (SI-284 data) were provided by the SCARF team from the 2010 CLSP Summer Workshop at Johns Hopkins University [61].

The audio data was processed into PLP features and then transformed into a phone posteriorgram representation using a hierarchical MLP [50]. The first MLP stage consisted of 351 input units (9 context frames of 39-dimensional features), followed by a 5000 unit hidden layer and then an output layer of 126 targets (42 three-state phone classes). The second stage MLP included 23 context frames, a 3000 unit hidden layer and a 42 unit output layer producing posterior probabilities for the phone classes. MLP training on the 35270

utterances employed a 6-fold cross-validation procedure. We then extracted phonetic events from the posteriorgram data using phonetic matched filters as described in Section 3.4 with a threshold of $\delta = 0.24$. For keyword search experiments, we partitioned the data into `fold1`, `fold2`, `dev`, and `test` folds consisting of 23, 23, 15.3, and 14.5 hours of speech, respectively.

Due to the very limited size of the TIMIT dataset, the number of keywords with sufficient training examples to estimate PPM word models consisted of just the words present in `sa1` and `sa2` utterances which were recorded by all 630 speakers. The considerably larger WSJ data set permitted a much more diverse set of search terms. In preparation for keyword search experiments, we first assembled a list of 1521 keywords from the WSJ corpus which satisfied the following criteria: 1) words contained a minimum of 4 phones, 2) minimum average word duration was 200 ms, and 3) words occurred at least 10 times in both `fold1` and `fold2`. For each of these terms we evaluated models constructed using keyword example counts of 1, 2, 4, 8, 16, 32, 64, 128, and 256, assuming that sufficient examples existed. For each example count size, we constructed multiple model instances using distinct random draws of keyword examples, and the FOM at each example count size represented the average over these model instances. Training was performed using keyword example data from one data fold, and then keyword search was evaluated on the opposite fold. We assessed the performance of three PPM keyword model types: MLE-based, dictionary model with phone confusions, and Bayesian models. A few representative plots of FOM versus the number of keyword examples are shown for four WSJ keywords in Figure 4.14. A summary of the relative performance of the three model types (MLE,

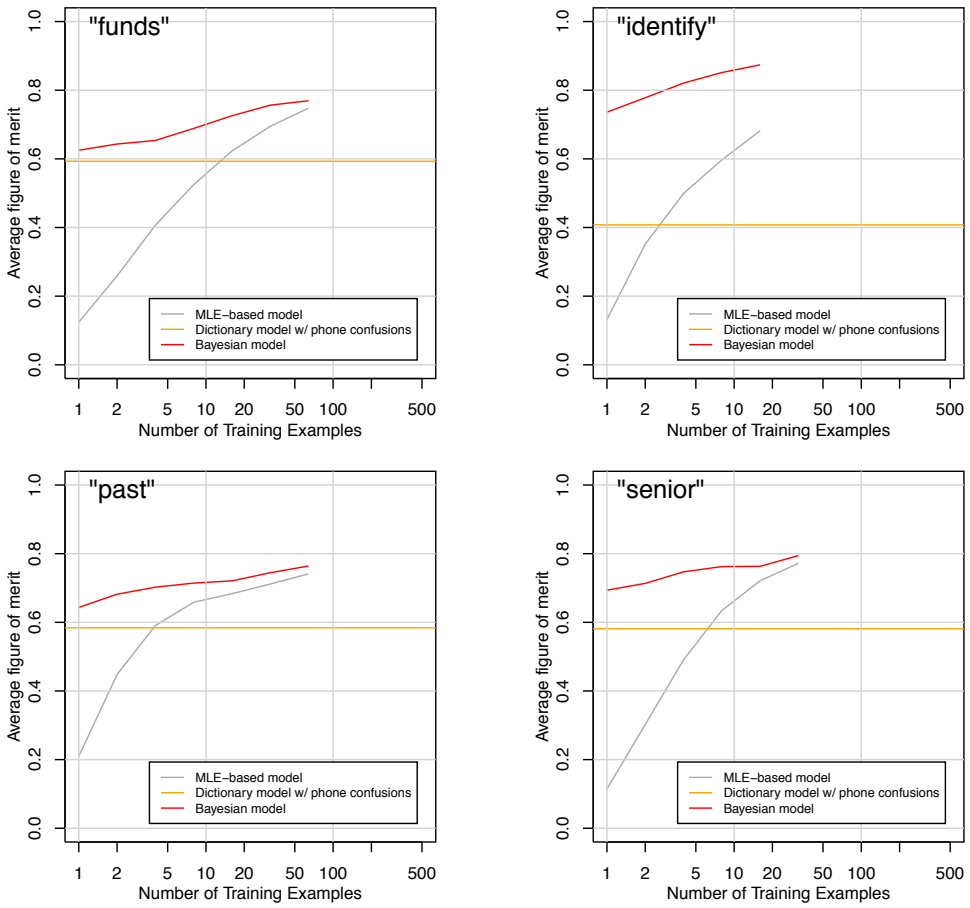

Figure 4.14: Average FOM vs. number of training examples used in model construction MLE, dictionary, and Bayesian PPM models for WSJ keywords: *funds*, *identify*, *past*, and *senior*.

dictionary, and Bayesian) averaged over all 1521 WSJ keywords is presented in Table 4.1.

4.8 Discussion

In comparing the various modeling approaches in experiments on TIMIT, we first note that the dictionary-based model dramatically outperforms both the GMM and MLE models in the low example count regime. Clearly, when there are fewer than 10 examples,

Table 4.1: Average percentage improvement in FOM of Bayesian models relative to MLE and dictionary models as a function of the number of keyword examples on WSJ data.

Number of keyword examples	Bayesian models relative to MLE models	Bayesian models relative to dictionary models
1	1449.4%	6.5%
2	566.3%	10.0%
4	222.0%	12.9%
8	97.9%	15.3%
16	47.2%	16.2%
32	21.7%	25.3%
64	9.8%	30.0%
128	4.3%	5.0%
256	1.7%	4.3%

insufficient data exists to estimate the distributions. However, when training data is abundant, the MLE and GMM models provide as much as 20% absolute increase in performance relative to dictionary-based models since they more accurately describe the word's phonetic events. Between the GMM and MLE models, we note that the GMM model provides a small improvement, approximately 6%, over the original MLE model for small example counts, likely because the GMM model has fewer parameters to estimate. Given the relatively small difference in performance on between GMM and MLE models on TIMIT, further experiments with GMM-based word models were not conducted on WSJ data.

Using the Bayesian model which incorporates both the canonical dictionary form and evidence from keyword examples, we achieve strong performance in all regimes. Particularly notable, Bayesian word models provide significant gains in keyword search performance when few keyword examples are available. Specifically, in TIMIT experiments we have achieved a 55% relative increase in keyword search performance over MLE models with 10 or fewer keyword examples. Considering a much larger set of search terms on the

WSJ corpus, we attained a 97.9% relative increase in performance for models constructed from 8 or fewer keyword examples.

Interestingly, these gains from MAP estimation occur despite using an extremely simple prior model. In Chapter 5, we will consider the gains possible by constructing more sophisticated prior models of phonetic timing. It should also be noted that the performance of the Bayesian and dictionary models is identical when the number of training examples is zero, but the logarithmic scale of the horizontal axes in Figures 4.13 and 4.14 did not permit this point to be shown. In a minor change from previous experiments in Chapter 3 and [29], we have incorporated a parametric model of word duration. In the experiments reported in this chapter, keyword duration is modeled using a gamma distribution whose parameters are estimated using maximum likelihood. For the keywords considered in TIMIT experiments, having 462 word examples permitted using the empirical distribution as a word duration model. For many WSJ keywords for which relatively few examples are available, a parametric model was required. We will further elaborate on word duration models in Chapter 5.

As previously mentioned, each point in the plots in Figures 4.13 and 4.14 represent the average performance of many models constructed from random draws of keyword examples. Although we have not included error bars, it is worth noting that the performance of MLE-based models exhibits very large variance. Bayesian models, on the other hand, are heavily constrained by prior distributions at small sample sizes and this results in dramatically smaller variance in performance. In fact, for the TIMIT experiments, we observed that variance in average FOM was *reduced by a factor of 14* for cases of few ($\leq$8)

training examples.

The estimation of Bayesian models does require decisions about prior distributions for the mean μ and precision λ which arise in the choice of hyperparameters. In this work we have made no effort to optimize these values. For prior estimates of precision λ, we tried $\mathrm{E}[\lambda] = 100$, 400, and 2500. Likewise for the parameter ρ relating the mean and variance of the gamma distribution for precision λ, we ran trials with $\rho = 0.25$, 0.50 and 0.75. For all cases, we observed little difference in the models or the resulting keyword search performance. The one parameter which we would expect to have the most dramatic effect is our estimate of average phone timing, μ_0. This could be improved by designing a more sophisticated approach to estimating prior phone timing distributions and will be considered in Chapter 5.

4.9 Conclusions

In previous applications of the point process model for keyword search, the chief limitation was the large numbers of keyword examples (>50) required to construct representative keyword models. Though simple models based solely on a word's dictionary form offer reasonable performance, they are incapable of benefiting when examples are available. In this chapter we have demonstrated that the use of Bayesian estimation techniques provides a principled method of combining both prior knowledge of phonetic composition and timing information from keyword training examples. Furthermore, we have shown the evolution in model distributions as the number of keyword examples grows ultimately results in an optimal interpolation of the performance gap.

Chapter 5

Improving Whole-Word Models Without Word Examples

In Chapter 4 we substantially improved the construction of whole-word acoustic models by using MAP estimation with a simple prior which included no information about the individual durations of constituent phones. The problem of modeling segmental duration has long been studied in the text-to-speech (TTS) community. We draw upon this work to develop a classification and regression tree (CART) approach for constructing prior models of phonetic timing which considers factors such as syllable stress, syllable position, adjacent phone class and voicing. This improved prior model closes 33% of the gap in keyword search performance between highly supervised whole-word models and those estimated without any examples.

5.1 Approaches to improving phonetic timing models

As previously reviewed, a large body of evidence suggests the preeminent importance of temporal properties of the speech signal in human speech perception. Likewise, temporal structure also plays a crucial role in producing natural sounding synthetic speech. Early attempts to predict the systematic changes in the duration of phonetic segments involved defining a set of hand-designed rules based on contextual factors such as adjacent segment identity, within-word position, syllable stress, among others [62]. A more statistically grounded approach that offers greater ability to model the interaction between factors is found in the sum-of-products model presented in [63]. CART-based modeling is another widely used approach to predicting segmental duration that provides automatic selection of relevant features, accommodates both categorical and continuous features, and produces easily interpretable rules [64].

In Chapter 4, we introduced a Bayesian approach to the estimation of whole-word acoustic models to overcome the problem of data sparsity. Phonetic timing is modeled using a Gaussian distribution, each Gaussian in this model requires the estimation of a mean and variance (or precision). MAP estimation of these parameters presumes the existence of reasonable prior distributions. In the initial presentation of MAP-estimated whole-word acoustic models, a very basic prior distribution was assembled from equally-spaced Gaussian means with uniform variance. This simple model suffices as an initialization point when combined with training examples. However, we would like to improve the estimation of word models for cases when no examples exist, and this naive prior ignores obvious differences in phone duration. As suggested, the problem of constructing a reasonable prior model of

phonetic timing is very closely related to that of computing segmental duration for TTS synthesis. In the following sections, we consider three approaches to defining phonetic timing distributions in the *absence* of any keyword examples. We begin by reviewing the simple dictionary model and then introduce two enhanced models.

5.1.1 Simple dictionary prior model

As presented in Section 4.5, if no training examples of a keyword are available, it is possible to construct a naive model of phonetic timing using the keyword's dictionary pronunciation. Given a normalized word duration of 1.0, we simply assign one Gaussian to each phone in the dictionary form using equally spaced means μ and a fixed standard deviation σ. An example of such a model with $\sigma = 0.05$ for the word "often" is depicted in Figure 5.5a. Despite its simplicity, we have shown this to be a practical method of assembling a prior for subsequent MAP estimation.

We have also found that introducing phonetic variation is a critical element in obtaining reasonable keyword search performance with such models. The use of alternate pronunciations could account for different speaker productions. However, another significant source of differences in observed phonetic events is caused by errors which occur in our phone posteriorgrams. A reasonable means of accounting for both errors and variation is to factor in phone confusion matrix data associated with the phone detectors. In the confusion matrix, each element C_{ij} represents $\Pr(p_j|p_i)$ where the rows correspond to actual phone classes (p_i) and columns correspond to predicted phone classes (p_j). Here, we have obtained a phone confusion matrix from the count matrix employed in phonetic event selection described in Section 3.5. To incorporate likely confusions into our dictionary

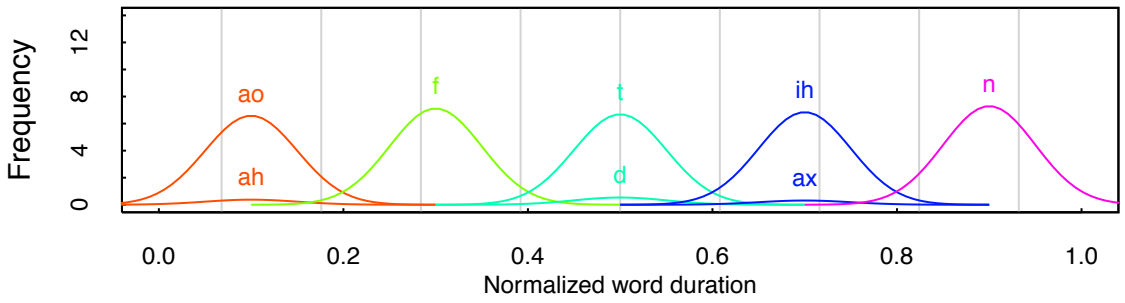

Figure 5.1: Simple dictionary phone timing model for the word "often."

model, we replace the single Gaussian for phone p_i from a word's dictionary form with multiple Gaussians for the confusable phones p_j each weighted by C_{ij} but sharing a common μ and σ as illustrated in Figure 5.1.

5.1.2 Monte Carlo prior using average phone durations

In the simple dictionary model, assigning Gaussian means at equal intervals corresponds to an assumption that all phones are identical in duration. The fixed standard deviation $\sigma = 0.05$ was chosen empirically to produce satisfactory keyword search performance over many keywords. To develop a more realistic model which accounts for phone duration, we first introduce the following expression for the relative timing of phonetic events. Given a word with baseform pronunciation $p_1, p_2, \ldots, p_N$, where each p_i is drawn from the set of all phones $\mathcal{P}$, we define D_i as a random variable representing the duration of the p_i. We can then define R_i as the midpoint of p_i (after word duration normalization), which is given by

$$R_i = \frac{\sum_{j=1}^{i-1} D_j + 0.5 D_i}{D_1 + D_2 + \cdots + D_N}. \tag{5.1}$$

These variables are depicted in Figure 5.2 for the example word "capital."

CHA
EXA

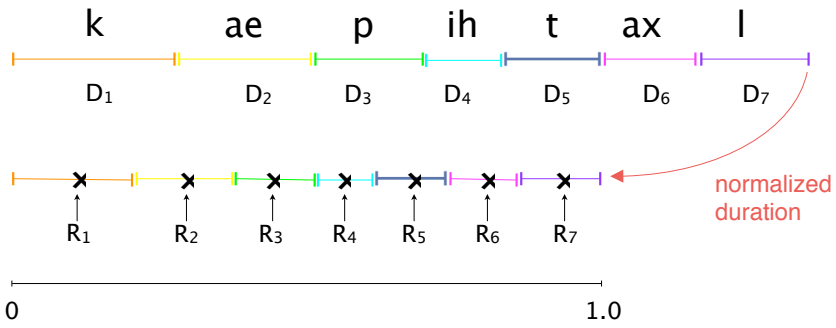

Figure 5.2: Illustration of how the midpoints (R_i) of phone segments within normalized word duration are calculated from constituent phone durations (D_i) for the example word "capital", /k,ae,p,ih,t,ax,l/.

As a starting point we assume that the distribution of the phone duration D_i is derived from the duration statistics of phone p_i realized across all words in the corpus and that D_i is independent of the other phones in the word. A convenient distribution for modeling phone duration is the two-parameter gamma distribution [40]. Studies have shown that the gamma distribution provides a high-quality fit to empirical phone and word duration distributions [41]. Figure 5.3 shows the empirical distributions of selected phones and the corresponding gamma distributions fit to the data extracted from the WSJ corpus.

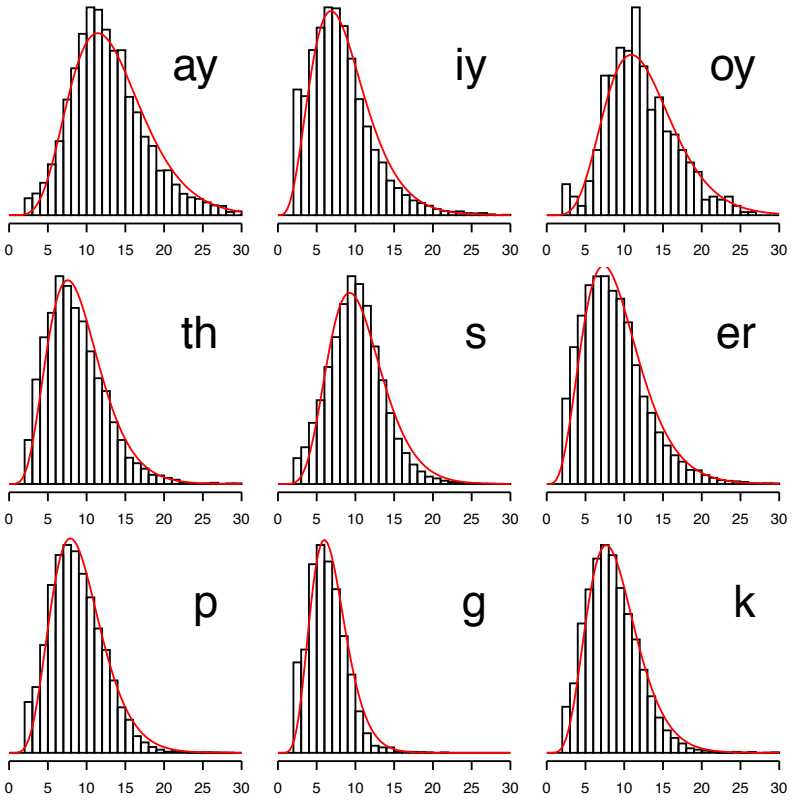

Figure 5.3: Empirical distribution of phone duration in frames for selected phones derived from the WSJ corpus data. The continuous distribution overlaid in red is MLE estimate of the gamma distribution fit to this data.

The random variable R_i is a function of N independent, gamma-distributed random variables and there is no simple closed-form solution for its distribution. Fortunately, it is sufficient for our purposes to estimate just the mean and variance of R_i. These quantities are easily obtained from a Monte Carlo simulation as follows: (i) compute gamma parameters (α, λ) to fit all of the phones based upon examples across the entire corpus; (ii) for a particular word, independently generate N sample phone durations corresponding to each D_i distribution; (iii) from the N duration samples D_i, compute the corresponding N values of R_i; (iv) repeat over many (10,000) iterations and compute sample mean and variance for each R_i; (v) construct a model from N Gaussian distributions using the mean and variances of each R_i.

An example of a model computed using this approach is shown for the word "often" depicted in Figure 5.5b. Unlike the simple dictionary model shown in Figure 5.1, we observe that the positions of the means better respect the average phone durations. Additionally, we find that the variances of the distributions are smaller for phones nearest the word beginning and ending, and larger for phones in the middle of the word. This is a natural byproduct of normalizing word durations and is evident in Equation (5.1) and can also be seen in models estimated from many keyword examples (Figure 5.5d).

5.1.3 Monte Carlo prior using CART-based phone durations

While the incorporation of average phone duration clearly improves the fidelity of the model compared with the simple dictionary version, it is well known that segmental duration is a function of many factors such as phonetic context, stress, syllable position. The text-to-speech community has developed several approaches to model segmental duration

and here we adopt the method based on classification and regression trees (CART). In order to perform CART training, we begin with a pool of example phone durations and an associated set of linguistically relevant features for each example. In speech synthesis, the prediction of duration is performed for an entire utterance, but our prediction can only consider a word in isolation. Therefore, it is not possible to consider some commonly used features such as utterance and phrase position. To compile a training set, we extracted phone durations from our corpus and generated a feature vector for each sample. The features associated with each phone and each word position were derived from the syllable and stress markings provided by the CMU dictionary [65]. We used the following set of features:

- `wsl` - word syllable count (7 levels)
- `swinit` - syllable word initial (boolean)
- `swfinal` - syllable word final (boolean)
- `sp` - syllable position (3 levels: onset, nucleus, coda)
- `stress` - stressed syllable (boolean)
- `prevoice` - previous phone voiced (boolean)
- `postvoice` - next phone voiced (boolean)
- `prev_bc` - previous phone broad class (5 categories)
- `post_bc` - next phone broad class (5 categories)

For each of the phones, we constructed a regression tree using the package `tree` in the statistical software package `R`. An example tree for the phone /ih/ is shown in Figure 5.4. In our example for the word "often" (/ao,f,t,ih,n/), the phone /ih/ which is preceded by a

stop consonant, is not in a stressed syllable, is followed by a nasal and is not word initial, the decision tree shown would predict an expected duration D_{ih} of 39.4 ms compared with the population average of 53.1 ms.

Unlike speech synthesis where it is sufficient to predict just a duration, we need to predict the distribution of the phone /ih/ in its context. We accomplish this by using the decision tree to cluster training examples and then estimate gamma distribution parameters (α, λ) at each node of the tree. The root node contains all examples, and its distribution represents the entire population independent of context. Each question in the tree partitions the examples into two subsets from which we compute corresponding gamma distribution parameters. Continuing to split our examples at each tree node allows us to compute distribution parameters for each context as shown in Figure 5.4. For cases in which the decision tree question is not applicable, the tree returns the parameters at the node where the question fails.

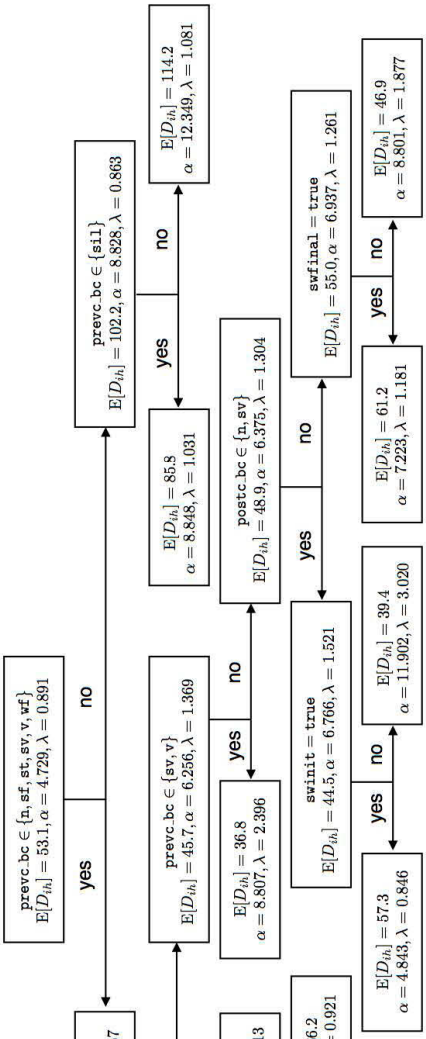

Figure 5.4: CART tree for predicting duration of the phone /ih/. Each node shows the decision tree question, mean duration E[D_{ih}], and gamma distribution parameters (α, λ) for all training examples at that node.

Having estimated the context-dependent gamma distribution parameters, the construction of a word model follows in a similar manner to the Monte Carlo model based on average phone durations in the previous section. However, instead of drawing samples for D_i from the distribution of the entire population, we instead use each phone's context determined by a word's dictionary form to identify the context-dependent distribution parameters contained in the decision tree.

An example of a model computed using this approach is shown for the word "often" depicted in Figure 5.5c. For reference, the model in Figure 5.5d is generated using MAP estimation with many training examples of the keyword. Note that in the progression of models from simple to more complex, the locations of the distributions better reflect the models derived from keyword data. To quantify the effect of improved timing models, we computed the root mean squared error (rmse) between the mean values of R_i under these three models and the positions determined from keyword examples. We found that the Monte Carlo average model provided a 16.7% reduction in rmse relative to the simple dictionary model, and the Monte Carlo CART model yielded a 21.3% relative reduction.

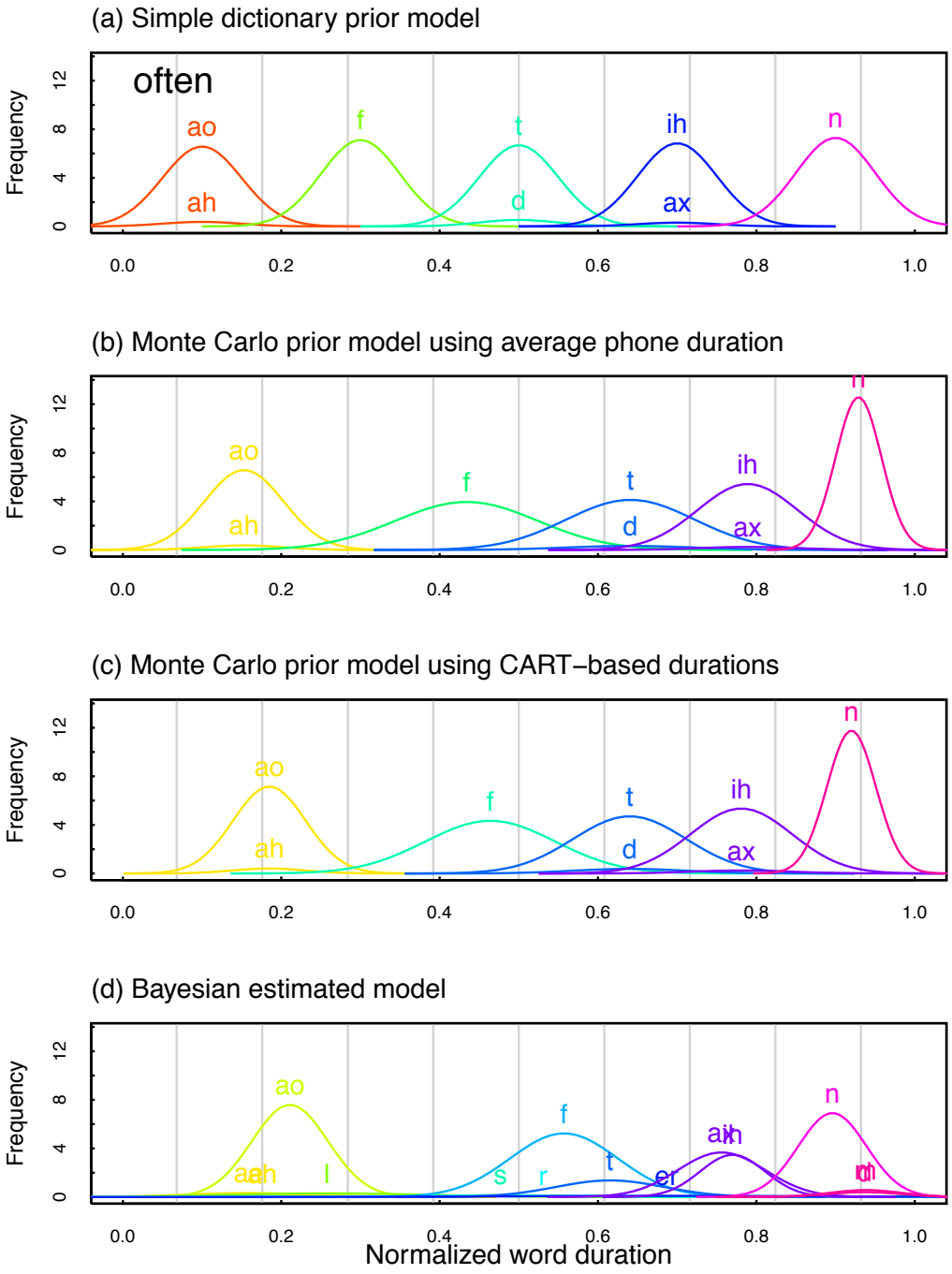

Figure 5.5: Example of phone timing models for the word "often."

5.2 Experiments

To measure the impact of more precise timing models, we conducted a series of keyword search experiments using the Wall Street Journal (WSJ0 and WSJ1) datasets. The training portions of this corpus were partitioned into two folds of 23 hours of speech. The audio data was processed into perceptual linear prediction (PLP) features and then transformed into a phone posteriorgram representation using a hierarchical MLP with 9 context frames [66]. From posteriorgram data, we then extracted phonetic events using phonetic matched filters as described in [67] with a threshold of $\delta = 0.24$. In order to evaluate models over a wide variety of words, we assembled a list of 1521 keywords from the WSJ corpus with minimum average duration of 200 ms, a minimum of 4 phones, and which occurred at least 10 times in each data fold.

For each keyword and each data fold, we created 4 types of keyword models: 1) simple dictionary model, 2) Monte Carlo estimated model based on average phone duration statistics, 3) Monte Carlo estimated model using CART-based phone duration statistics, and 4) Bayesian estimated models as described in Chapter 4. Of these four phone timing model types, the first three were constructed without using keyword examples and only relied on duration statistics of their constituent phones. On the other hand, the Bayesian model used all available keyword examples. All training and model parameter estimation (phone duration statistics, CART estimation, etc.) was performed on one data fold and evaluation was performed on the other, unseen data fold. Performance reported represents an average per keyword over both folds.

We evaluated keyword detection performance using average figure of merit (FOM).

A summary of results of keyword search experiments is shown in Figure 5.6. The average performance of the various models is enumerated in Table 5.1. In addition to the average over all 1521 keywords, the table also shows the average over the subset of keywords which ranked below the bottom 10th percentile with the simple dictionary model.

We observe from Table 5.1 that improvements in estimating the prior models of phone timing distributions do result in improvements in FOM, on average. However, it would appear most of the gain in prior model performance is obtained from the Monte Carlo estimation of relative timing. The additional gain achieved through the inclusion of CART duration modeling was limited. There were several keywords for which the CART model provided improvements compared to the Monte Carlo average model, as evidenced

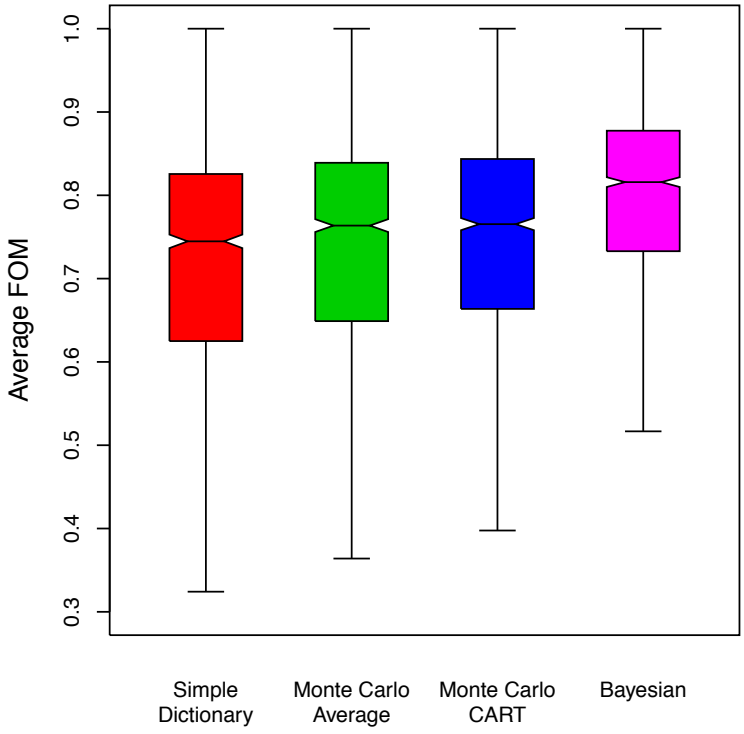

Figure 5.6: Boxplots depicting average figure of merit for 1521 WSJ keywords for each model type.

Table 5.1: Comparison of figure of merit based on 1) average over all 1521 keywords, and 2) average over subset of keywords which scored in the lowest 10th percentile using the simple dictionary model.

model type	mean FOM all words	mean FOM dict lowest 10%
Simple dictionary	0.704	0.322
Monte Carlo (average)	0.725	0.385
Monte Carlo (CART)	0.733	0.413
Bayesian	0.791	0.605

by the significant increase in the minimal FOM value. However, we were unable to identify a systematic keyword property that accounted for these occurrences. While generating a more sophisticated prior model for phonetic timing was a logical place to look for improved performance, we observe other factors in Bayesian models which account for their superior performance. Chiefly, Bayesian models more accurately represent phonetic variation observed in keyword examples. This suggests that further improvement might come by adding alternate pronunciations in our dictionary. Additionally, more investigation may reveal systematic errors in phone posteriorgram estimates which may be predictable from context instead of using on phone confusion matrix data.

5.3 Conclusions

In our previous work on MAP estimation of whole-word acoustic models in Chapter 4 and [68], we demonstrated that a Bayesian approach to estimating phone timing models provided significant gains in keyword search performance in the case that few keyword examples are available. In Chapter 4, the prior model of phone timing used in MAP estimation was based on the simple dictionary model. The motivation for this work was to

assess the gains possible by considering more sophisticated prior models. By incorporating a Monte Carlo approach to estimating phone-timing distributions, we were able to obtain a 4.2% relative improvement in average FOM compared to using a simple dictionary model. While modest in absolute terms, this gain represents 33% of the difference in performance between simple dictionary and MAP-estimated models.

Chapter 6

Speeding up PPM Decoding

In this chapter we consider the evaluation of the point process model detection function. Normally, the calculation proceeds incrementally, frame-by-frame. However, in decomposing the likelihood ratio which comprises the detection function, we observe that the effect of each phonetic event is a simple additive contribution. Each word can be represented by a "score matrix" and the contribution of a phonetic event is a time-reversed "score vector." In place of a frame-by-frame evaluation, we can instead proceed event-by-event. This view enables huge speedups; runtime no longer depends on the frame rate and is instead linear in the number of events. We apply this intuition to redesign the runtime engine behind the point process model for keyword search. In this chapter, we demonstrate impressive real-time speedups (500,000x faster than real-time) with minimal loss in search accuracy.

6.1 Background

State-of-the-art spoken term detection (STD) systems based on ASR lattice search offer very strong performance, but it comes at a high cost. Assuming the existence of an appropriate ASR pipeline, there is a non-trivial computational overhead associated with running full recognition, and after having processed the speech, we are confronted with the issue of storing and searching the resulting lattices. In order to provide reasonable access speed, it is typical to employ an inverted index the size of which can easily be on par with the word lattice itself. A recently published state-of-the-art STD system for Turkish Broadcast News in [69] using a finite-state transducer based index reports an average search time of 4 ms per query on 163 hours of audio (nearly 150,000,000x real-time). However, achieving this search speed requires an index more than twice the size of the corresponding word lattice. Handling out-of-vocabulary (OOV) terms poses a further challenge. By definition these terms will not be present in ASR word lattices, therefore many STD systems fall back on searching potentially larger phonetic lattices in order to handle these queries.

In addition to the size of the index, there is also significant processing overhead involved with index construction. The figures reported in [70] on the IBM system constructed for the NIST 2006 STD evaluation provides some insight. This system recorded an average query time of 0.0041 sec per hour of speech (878,000x real-time) and an index size of 0.327 MB per hour of speech. However, these numbers do not account for the index construction time (including audio processing, word/phonetic lattice generation and index creation) of 7.56 hours of processing per hour of speech (8 times *slower* than real-time). Furthermore, if we are dealing with a constant stream of audio, we need to consider the ease with which an

index can be augmented with new data. Consider the task of spoken term detection applied to the intake of a typical day at YouTube. At the time of writing, YouTube reports that its users upload 100 hours of video on average every single minute (or equivalently, 16.4 years of content per day). Considering the volume of this stream, perhaps a less complex solution has merit for certain scenarios.

The dynamic match lattice search STD system presented in [71] is not based on an LVCSR system. It offers open vocabulary search on phonetic lattices and reports an average search speed of 2 sec per hour of speech (1,800x real-time). Certainly, older HMM-based keyword search systems, which predate ASR lattice-based approaches, have reasonable results with relatively low complexity, but search speed is lacking. The basic HMM-based system described in [71] was implemented in [72], which reported a search speed of roughly 33 times real-time.

With these issues in mind, we begin by arguing the advantages of a sparse representation and outline a novel detection framework. After reviewing the evaluation of the detection function in PPM keyword search, we develop an efficient upper bound consistent with our proposed approach. Finally, we validate these ideas in keyword search experiments.

6.2 Characteristics of an efficient keyword search system

An ideal keyword search system is one which offers compact representation, provides fast query times, and can easily incorporate new data. Before addressing approaches to keyword search, we should pause to consider the representation of the signal upon which our decoder must operate. For a signal T frames in duration and an HMM with N states,

the signal representation consists of the observation likelihoods at every state and every frame, $O(NT)$ real-valued quantities. It is on top of this representation which the Viterbi algorithm operates. Determining the most likely path requires that we consider all transitions into every state and thus, in the absence of pruning, decoding is constrained to proceed in time $O(N^2T)$.

While HMMs operate on a dense frame-based representation, one possible starting point for developing faster alternatives begins with a sparse representation of the speech signal. Consider a framework in which the speech input is reduced to a discrete set of impulses, each corresponding to a phonetic event. The density of this representation is solely a function of the phone production rate, significantly lower than the typical 100-Hz frame rate. Further, it is independent of the phone set or state space dimension N.

The goal is to identify keywords. We desire a system which takes as input discrete phonetic events and efficiently produces a real-valued output where high values correspond to the presence of a keyword. In subsequent sections we will demonstrate that the contribution of each phonetic event to the total detection function in the PPM system can be seen as the addition of a time-reversed score vector. A keyword model can be viewed as a collection of filters $h_p(t)$ for each phone p whose impulse response is the score vector. The score vector for phone p relates to the log likelihood of observing phone p at each position within the word. Phonetic events can be viewed as impulses. Computing a detection function becomes a simple convolution of impulses with filters which can be made to run very fast and no longer depends on frame rate or alphabet size. An example of this operation in the point process keyword search framework is shown in Figure 6.1. In the following

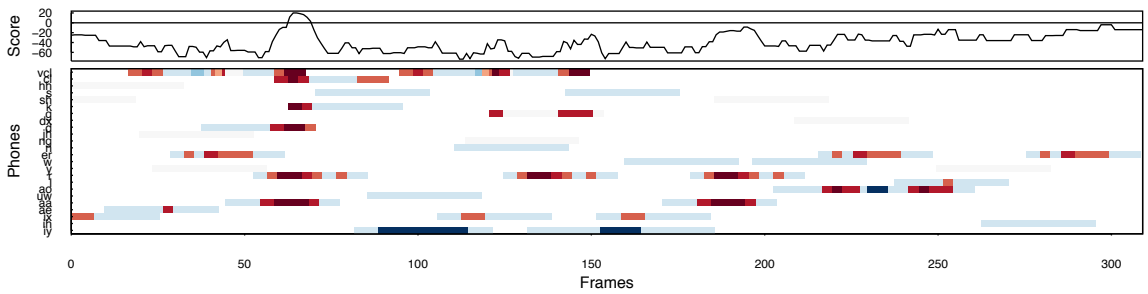

Figure 6.1: A visual representation of the calculation of a keyword detection function (top) by summing over scores from each phonetic event impulse response. Dark red indicates large positive score and blue indicates negative score. The detection function is computed at each frame by summing the scores across the phone set.

sections, we detail how the point process model decoding can be cast in this form.

6.3 Bounding the detection function

In this section we review the point process detection function previously presented in 2.3, and then we present approximations to enable fast searches. In the point process model framework, the input speech signal is represented by an extremely sparse set of phonetic events. As introduced in Chapter 3, we consider events taken as the maxima of a filtered posterior trajectory function which results in a single event per phone occurrence. An illustration of phonetic events for the TIMIT utterance "This was easy for us" is shown in Figure 6.2. Models built upon this representation take advantage of not only the identities of the phones detected but also the sequence and relative timing between the events.

Keyword detections are marked as the maxima of the detection function $d_w(t)$ defined as the ratio of the likelihood of the collection of phonetic events $O(t)$ in the interval $(t, t + \Delta T]$ under the word model (θ_w) relative to its likelihood under a background model

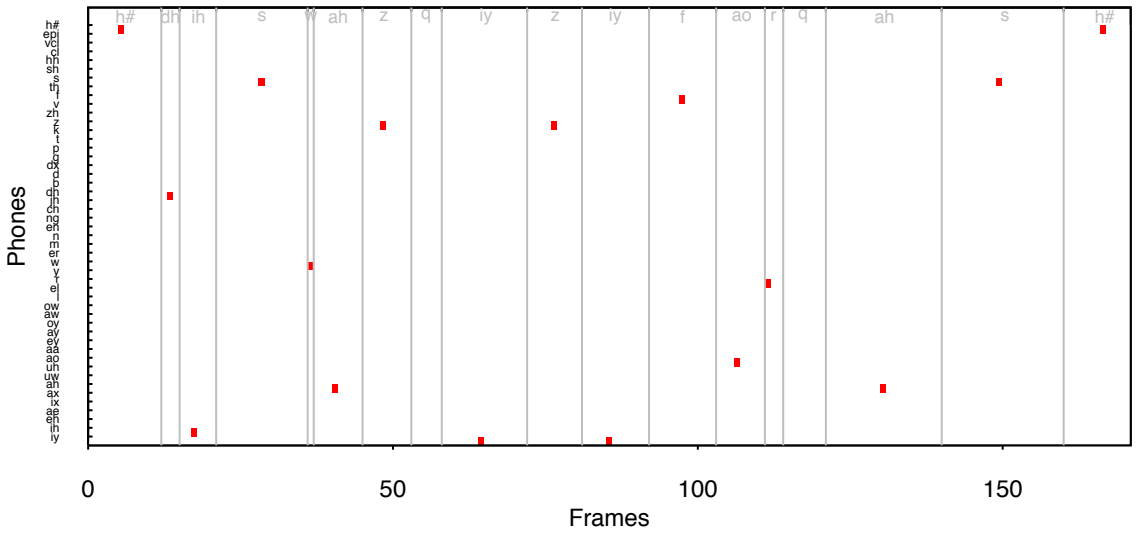

Figure 6.2: Phonetic events for the TIMIT utterance "This was easy for us".

(θ_{bg}). As presented in Section 2.3, the keyword detection function $d_w(t)$ is given as

$$d_w(t) = \log\left[\int_0^\infty \frac{P(O'(t)|T,\theta_w)P(T|\theta_w)}{T^{|O(t)|}P(O(t)|T,\theta_{bg})}dT\right]. \tag{6.1}$$

To simplify the evaluation of Equation (6.1), we will consider the approximate detection function

$$d_w(t) \approx \max_T \log\left[\frac{P(O'(t)|T,\theta_w)P(T|\theta_w)}{T^{|O(t)|}P(O(t)|T,\theta_{bg})}\right] \tag{6.2}$$

and show how terms can be combined. First, note that $|O(t)| = \sum_p n_p$ and $n_p = \sum_d n_{p,d}$. Also, using $\lambda_p = \sum_d \lambda_{p,d}/D$, we may rewrite the argument of the log in Equation (6.2) as

$$P(T|\theta_w)\prod_{p\in\mathcal{P}}\prod_{d=1}^{D}\left(\frac{\lambda_{p,d}}{\lambda_p T}\right)^{n_{p,d}} e^{\lambda_p T/D - \lambda_{p,d}/D}.$$

After taking the log, we find the approximate detection function consists of three terms,

$$\begin{aligned} d_w(t) \approx \max_T \Bigg\{ &\log P(T|\theta_w) + \sum_{p\in\mathcal{P}}\left(\lambda_p T - \frac{1}{D}\sum_{d=1}^{D}\lambda_{p,d}\right) \\ &+ \sum_{p\in\mathcal{P}}\sum_{d=1}^{D} n_{p,d}\phi_{p,d}\Bigg\} \end{aligned} \tag{6.3}$$

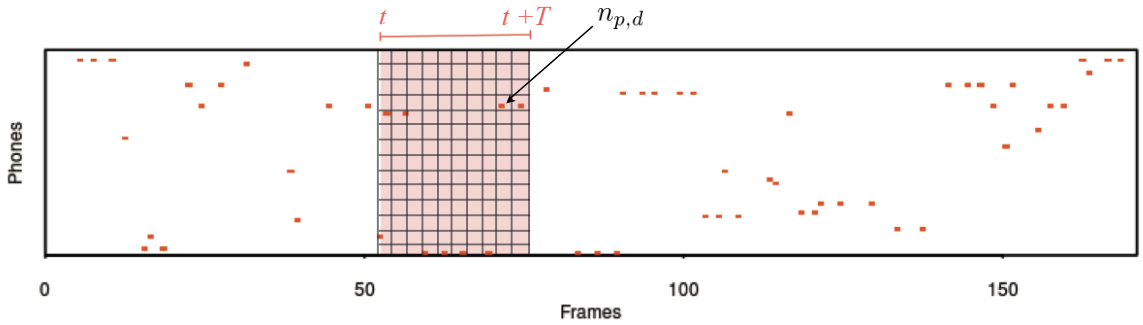

Figure 6.3: Overview of frame-by-frame evaluation of the detection function $d_w(t)$. At each frame t and candidate duration T, the detection function is evaluated by first accumulating counts of phonetic events for each phone p and division d, the quantity $n_{p,d}$. The sum over the product of $n_{p,d}$ and weighting factor $\phi_{p,d}$ is used to compute the detection function at frame t.

where $\phi_{p,d} \triangleq \log(\lambda_{p,d}/\lambda_p T)$. Note that the first two terms depend only on the word duration model $P(T|\theta_w)$ and the Poisson rate parameters $(\lambda_p, \lambda_{p,d})$ but are independent of observed phonetic events. Thus the detection function depends on the total event count $n_{p,d}$ for each phone p and word division d times a weighting factor $\phi_{p,d}$. As illustrated in Figure 6.3, direct evaluation of this function proceeds as follows: (i) for each time t and sample keyword duration T, determine $O(t)$, the set of phonetic events which occur in the interval $(t, t+T]$; (ii) from $O(t)$ accumulate the total count $n_{p,d}$ for each phone and word division; (iii) sum over the product of $n_{p,d}$ and its corresponding weighting factor $\phi_{p,d}$; and, (iv) repeat for each candidate duration T and take the max.

6.3.1 A simple upper bound

Although computationally simple, the direct implementation of (6.3) requires that for each event we determine the word division d to which it belongs as the window of length T slides right one frame at a time. Our first approach to speeding up the computation of

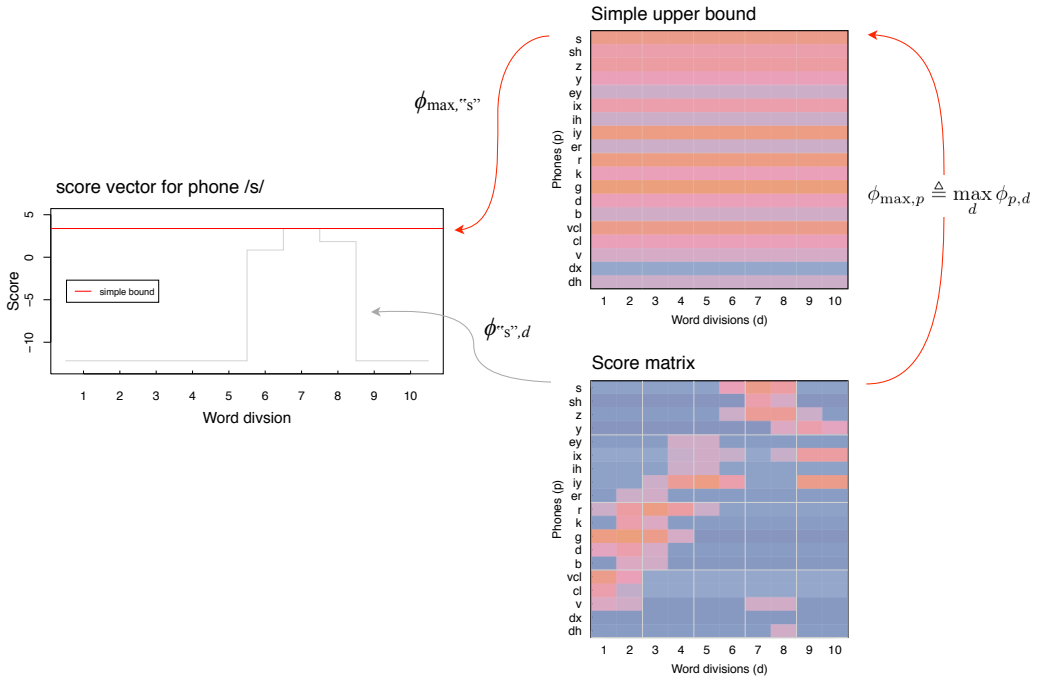

Figure 6.4: Diagram illustrating how simple upper bound is extracted from score matrix for TIMIT keyword "greasy." The bound is maximum score for each phone (row) across all time divisions. A score vector and bound are plotted for phone /s/.

the detection function was to replace full evaluation of (6.3) with a simple upper bound on the score vector. As shown in Figure 6.4, it is easy to see that using $\phi_{\max,p} \triangleq \max_d \phi_{p,d}$, the maximum weighting factor over all divisions d, provides an upper bound on $d_w(t)$ and liberates us from evaluating $n_{p,d}$ (i.e., n_p suffices).

To demonstrate the effect of the simple upper bound, we present the original and bounded detection functions for a TIMIT `sa1` utterance based to two different sources of phone posterior data in Figure 6.5. The upper plot shows oracle phonetic events (i.e., phonetic events derived directly from the true phone labels) and the lower plot shows SMLP-based events are derived from a sparse multilayer perceptron (SMLP) phone recognizer [49] with phonetic matched filtering [67]. Oracle phonetic events represent perfect phonetic

information. Since this simplification reduces the model to a "bag of phones," it is not surprising that the upper bound is loose. We observe that the upperbound is not sufficiently tight except for the case of oracle phonetic event data, and this is corroborated by the poor performance observed in keyword search experiments.

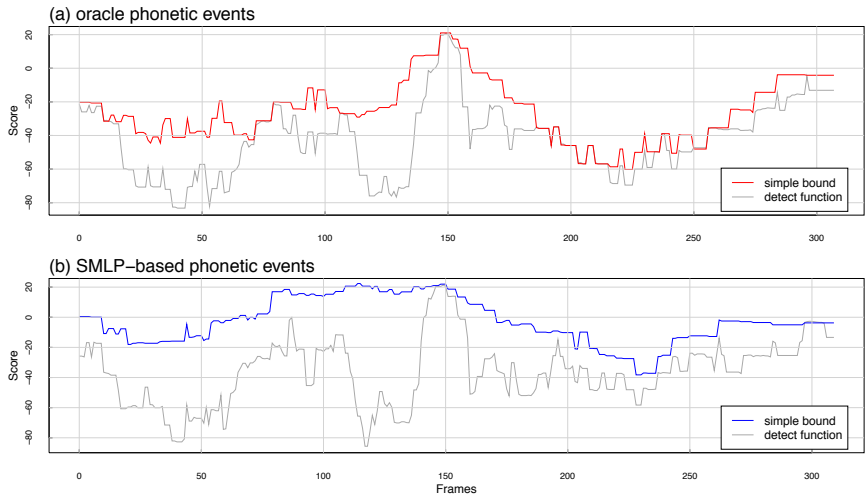

Figure 6.5: Detection function $d_w(t)$ and the simple upperbound for keyword "greasy" using oracle and SMLP-based phonetic events.

Another feature of the bounded detection function is that changes in $d_w(t)$ only occur when an event enters or exits a window of duration T. Instead of storing the value of the detection function at each frame, we may instead retain the much sparser set of *changes* to the detection function. With delta coding we maintain one accumulator array and only perform two additions for each phonetic event. Since we observe 16 phonetic events per second on average, a factor of 6 lower than the frame rate of 100 Hz, recording only score changes entails significantly fewer additions.

6.3.2 Tightening the detection function upper bound

What accounts for the difference in the two bounds in Figure 6.5? The matrix of weights $\phi_{p,d}$ is computed from the homogeneous and inhomogeneous Poisson rate parameters. Intuitively, phonetic events consistent with the keyword (i.e., the correct phone and relative timing) result in positive weights, and those which are inconsistent with the model

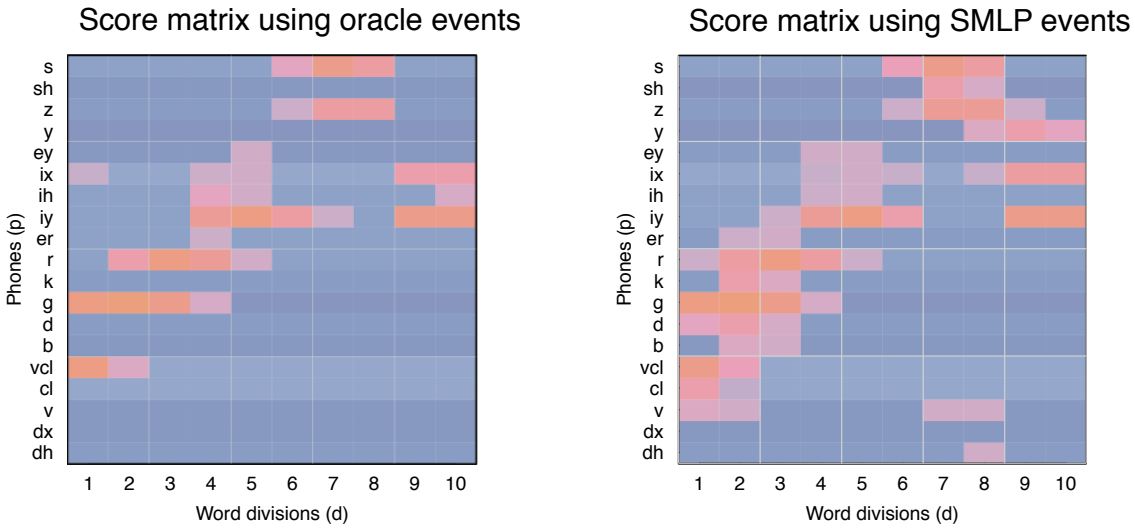

Figure 6.6: Score matrix corresponding to TIMIT keyword "greasy" using oracle (left) and SMLP (right) posterior data. The color red represents large positive score values and blue represents large negative score values. The concentration of positive score values in the oracle matrix due to absence of phone confusions results in more discriminative model particularly when using a simple upperbound.

contains phone detector confusions. This contrast is evident when comparing the difference in score matrices between oracle and non-oracle models as shown in Figure 6.6. In this figure, red denotes scores with large positive values and blue represents large negative values. Comparing two models, the more discriminative one will contain a higher fraction of large negative values in the $\phi_{p,d}$ terms. Large negative $\phi_{p,d}$ terms are produced when the total number of events observed in keyword training examples corresponding to phone p and division d is zero. Phonetic confusions result in non-zero counts, which is exacerbated by simple upper bound which takes only the maximum $\phi_{p,d}$ for all d.

Keyword models derived from oracle data do not exhibit the errors present in real phone detectors which results in relatively few phones with positive scores as illustrated in Figure 6.7. This accounts for the difference in the tightness of the bounds seen in Figure 6.5 and it also offers insight into improving the bound.

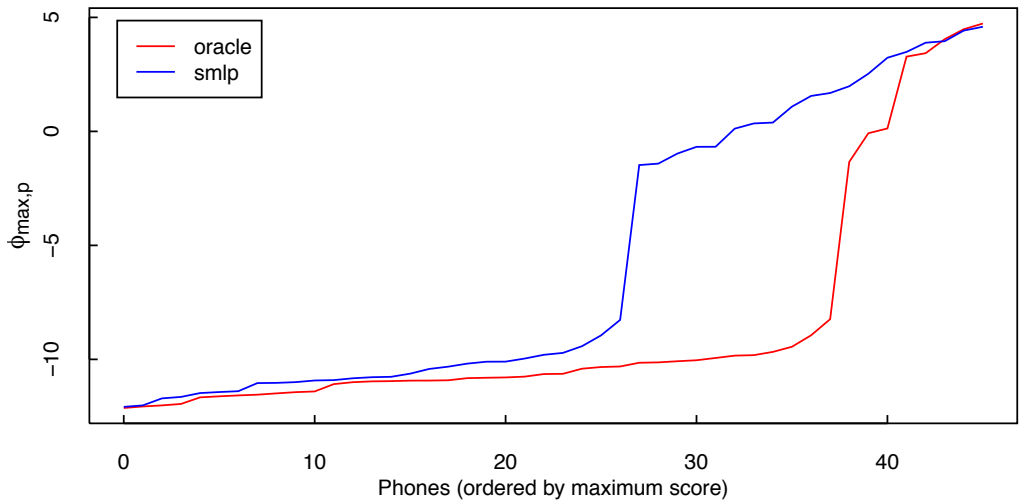

Figure 6.7: Distribution of maximum score $\phi_{\max,p}$ values for oracle and SMLP-based phonetic events for the word "greasy" used in the simple upperbound. The larger fraction of negative scores results in a more discriminative model, and explains the difference in tightness of the bounds in Figure 6.5.

The simple upper bound $\phi_{\max,p}$ is fast because it requires that we encode only two score changes per event. To tighten the bound, we can instead consider multi-segment, piecewise constant upper bounds as illustrated in Figure 6.8. The 3-segment bound may contain up to 4 score changes, but we permit fewer depending on the score vector. Likewise, the D-segment bound may contain as many segments as word subdivisions. By considering multi-segment bounds, we significantly improve the ability to discriminate by phonetic event position, while retaining most of the computational advantage of only evaluating changes in the detection function.

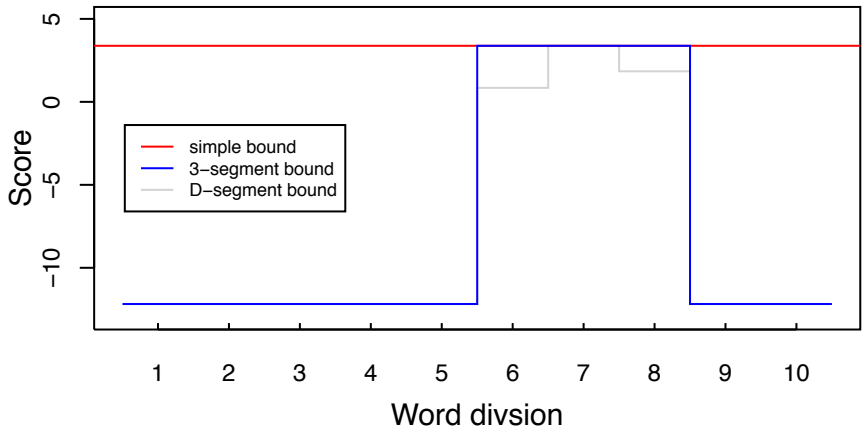

Figure 6.8: Multiple upper bounds for the score vector for phone /s/ in the TIMIT keyword "greasy." The simple upperbound (i.e., 1-segment bound) is just the maximum over all time divisions. The D-segment bound corresponds to partitioning the score vector with as many as D different partitions so as to minimize the difference between the bound and the score vector. The case of $D = 3$ is shown.

6.4 Detection function as a convolution

Another way to envision the computation of the detection function is to recognize it as the summation over all phones of a sequence of phonetic events (i.e., an impulse train for phone p) convolved with its corresponding score vector (i.e., a filter impulse response,

$h_p(t)$). This is depicted more clearly in Figure 6.9. The convolution operation alone does not suggest any computational savings, but because the input is a sparse set of impulses, we are liberated from shifting and multiplying. Direct implement of (6.2) requires computing an event's position within a sliding window $(t, t+T]$ for each frame t. The alternate view just described allows us to *invert* the process; we proceed event-by-event and immediately record incremental contributions to the detection function using the time-reversed score

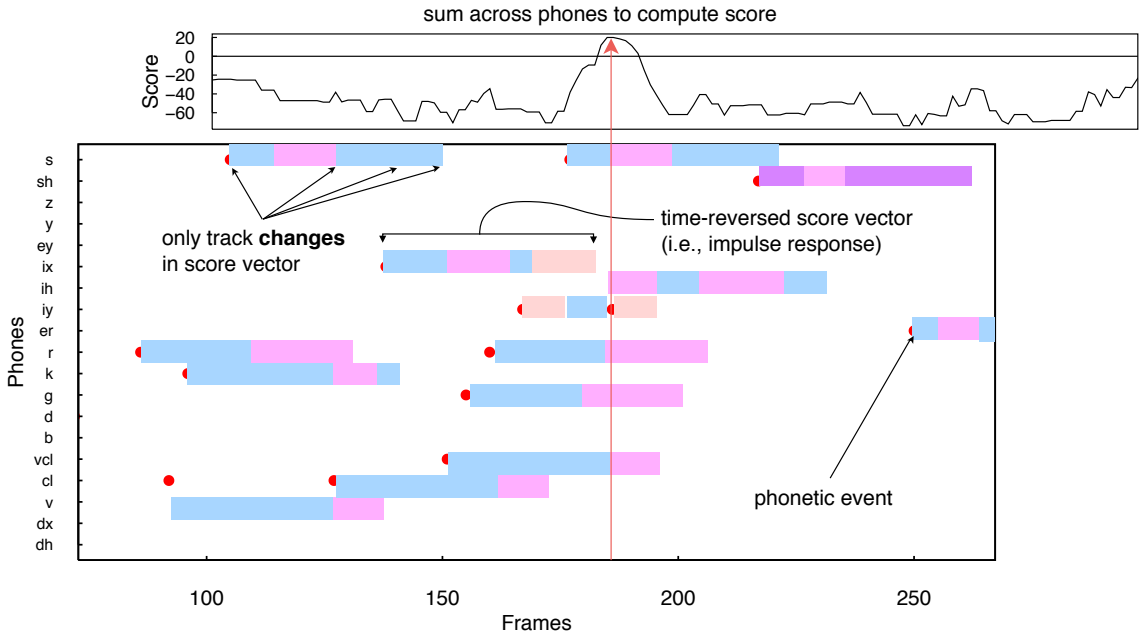

Figure 6.9: Inverting the calculation of the PPM detection function. Each phonetic event (red dot) is shown with its score contribution (i.e., time-reversed score vector). Frame-by-frame calculations can be avoided by only considering score changes. The detection function is computed by summing score contributions across phones.

6.5 Experiments

In this section we report results of keyword search experiments using various detection function bounds. We utilized the Wall Street Journal (WSJ0 and WSJ1) datasets which were partitioned into two folds of 23 hours of speech. The audio data was processed into perceptual linear prediction features and then transformed into a phone posteriorgram representation using a hierarchical MLP with 9 context frames. We then extracted phonetic events from posteriorgram data using phonetic matched filters as described in [67] with a threshold of $\delta = 0.24$.

As detailed in Chapter 4, we assembled a list of 1521 keywords from the WSJ corpus with minimum average duration of 200 ms, a minimum of 4 phones, and which occurred at least 10 times in each data fold. For each keyword and each data fold, we computed keyword model parameters θ_w using all available keyword examples in that fold. The models in these experiments were based on MLE parameter estimates so performance depended on the number of keyword examples. Models from one fold were used to search for keywords in the other fold, and detections were declared at local maxima of $d_w(t)$ above threshold δ_w. Detections within 100 ms of the beginning of the keyword in the transcript were marked as correct. Multiple correct detections of the same keyword were discarded, and all other detections were recorded as false alarms. For the results listed in Table 6.1, we calculated average figure of merit (FOM), the mean detection rate given $1, 2, \ldots, 10$ false alarms per keyword per hour as the threshold δ_w was varied.

We evaluated four versions of the decoding algorithm: (i) the frame-by-frame, direct implementation of Equation (6.2); (ii) the simple (1-segment) upper bound using

$\phi_{\max,p}$; (iii) a 3-segment upper bound; and, (iv) a D-segment upper bound with a maximum number of segments $D = 10$. All versions were coded in Java, but we also include results for a C++ version of (iv). In computing the real-time speedup (RTS), we included in the processing time the overhead of reading phonetic event data, scoring the detections, and saving the results. These results represent processing on a single-core of a 2.66-GHz Intel E5430 Xeon processor. The relative performance is plotted in Figure 6.10.

In Table 6.1 we observe that compared to the direct implementation of the detection function, computing the simple upper bound is 57 times faster, but results in a 75% relative decrease in average FOM. A simple 3-segment upper bound on $\phi_{p,d}$ reduces speed by only 8% yet recovers almost all of the previous loss in FOM. Finally, a D-segment bound is 13% slower than the simple bound, but offers virtually identical FOM performance as the direct implementation. Finally, we note that with a C++ implementation of the D-segment bound, we can obtain search speeds in excess of 500,000x faster than real-time.

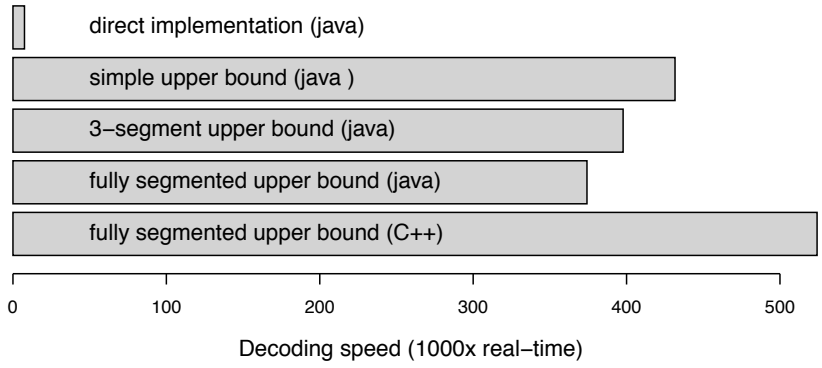

Figure 6.10: Relative search speed performance of various decoding algorithms in terms of real-time factors.

Table 6.1: Comparison of median FOM and search speed on 1521 keyword set using various decoding algorithms.

algorithm	FOM	ΔFOM	Speed (RTS)
direct implementation (java)	70.9	–	7,483
simple upper bound (java)	17.9	-74.7%	431,594
3-segment bound (java)	68.4	-3.5%	397,752
D-segment bound (java)	70.5	-0.5%	374,195
D-segment bound (C++)	70.5	-0.5%	524,189

6.6 Conclusions

In this chapter we have presented a novel framework for keyword search in which speech is represented as a sparse set of phonetic impulses and keyword detection is implemented as convolution with an ensemble of filters. By approximation with an upper bound, we have shown that the point process model keyword detection function can be cast in this framework. Finally, we have demonstrated keyword search experiments which averaged better than 500,000x faster than real time with only negligible loss in FOM. The ability to conduct rapid keyword searches is a key advantage of the point process model approach and makes it possible to consider queries on very large volumes of speech data. This attribute will be highlighted in Chapter 7.

Chapter 7

Spoken Term Detection on Conversational Telephone Speech

Previous chapters have been focused on improved model estimation techniques and efficient search algorithms, but evaluations have been limited to searching relatively easy scripted corpora for simple unigram queries. In this chapter, we introduce techniques for score normalization and the processing of multi-word and out-of-training query terms as required by the 2006 NIST Spoken Term Detection (STD) evaluation, permitting the first comprehensive benchmark of PPM search technology against state-of-the-art word and phonetic-based search systems. We demonstrate the PPM system to be the fastest phonetic system while posting accuracies competitive with the best phonetic alternatives. Moreover, we show that PPM index construction time and size are better than any keyword search system entered in the NIST evaluation.

7.1 PPM for spoken term detection

Previous evaluations of PPM keyword search compared system performance to older HMM-based keyword-filler approaches on relatively simple datasets. To establish a more definitive benchmark against other well documented STD systems, we decided to quantify PPM performance on the 2006 NIST STD evaluation. To complete this evaluation, we addressed several challenges necessary for extending PPM techniques to the task of spoken term detection on conversational telephone speech. Earlier PPM experiments considered the modeling and search for single-word queries and assumed that training examples for all words were available. In contrast, the 2006 NIST STD evaluation plan [73] required the search for "terms" defined as sequences of consecutively spoken words with gaps of up to 0.5 seconds allowable between words. In the following section we consider the modeling of multi-word terms as single units in the PPM framework and briefly address performing multi-word queries by searching for the individual term subcomponents. Beyond the issue of multi-word search terms, the 2006 STD evaluation also necessitated the development of techniques to handle queries which do not appear in training, specifically in the PPM context, the need to estimate word duration absent any word examples. Building from earlier work in Chapter 5, we introduce a Monte-Carlo approach for estimating word duration distributions. Finally, maximization of the actual term-weighted value (ATWV) performance metric used in the NIST evaluation requires accurate assessment of detection confidence level, so we consider the normalization of PPM detection scores.

7.1.1 Whole-word modeling approaches to multi-word terms

We considered four approaches to handling multi-word terms. The first and most basic is a simple concatenation of the phonetic forms of the individual terms. For example, the search term "health insurance" would be constructed from the phonetic (dictionary) form for "health" concatenated with the phonetic form for "insurance." A word model is constructed directly from the phonetic sequence using equidistantly spaced Gaussian distributions with a fixed variance (see dictionary-based models in Section 4.5). We refer to this as the simple dictionary concatenation and it has the advantage of requiring no actual training examples.

In all previous work we have found that long keywords are much easier to identify than short ones, and we expect multi-word terms to be consistent with this finding. However, word model performance is also highly correlated with number of word examples available, and it is likely that we will observe fewer examples of multi-word terms in their entirety. We have previously demonstrated in Chapter 4 that MAP estimation is an effective technique for synthesizing word models from few training examples. Therefore, beginning with a simple dictionary concatenation model prior, we incorporate all the training examples of the term to compute a MAP-estimated whole-word model. We refer to this as a MAP-estimated model using a simple dictionary prior.

Multi-word terms offer another possible approach. It is very likely the case that we have many more examples of the individual words which comprise a multi-word term than we have complete examples of the multi-word term. For instance, for "health insurance" it is probable that there are numerous examples of the individual components "health"

Table 7.1: A comparison of multi-word modeling techniques of 571 multi-word terms on the Switchboard development corpus.

model id	description	ATWV
ppm1	simple dictionary concatenation	0.4002
ppm2	MAP-estimated using simple dictionary (ppm1) prior	0.4925
ppm3	concatenated MAP-estimated unigram prior	0.5179
ppm4	MAP-estimated whole-word using unigram (ppm3) prior	0.5247

and "insurance." This offers the possibility of improving our prior model by starting with individual MAP-estimated models of the words "health" and "insurance," and then concatenating them together to form an improved prior. We refer to this as a concatenated MAP-estimated unigram prior. Finally, the few examples of the complete multi-word term can then be used in a new MAP-estimated model which starts from this improved prior.

To evaluate the relative performance of these approaches, we constructed an STD experiment on 230 hours of the Switchboard dataset and considered detection performance on multi-word terms. Results are listed in Table 7.1. While significant gains are evident between the simple dictionary prior and the MAP-estimated model, the more sophisticated prior and subsequent MAP estimation yielded smaller improvements.

As an alternative to modeling a multi-word term in its entirety, we also considered searching for a term as the ordered union of sub-term detections with loose constraints on timing. Conceivably, this approach has two immediate advantages. First, individual words or sub-term models can be constructed independently which permits flexibility in the creation of detailed models. Second, detections of word sequences with intermediate silences are possible. Unfortunately, this method also raises a number of other issues such as how best to assign scores to multi-word detections. Additionally, conducting independent

searches does incur a search speed performance hit. After some preliminary experiments, we determined that further investigation was not warranted.

7.1.2 Duration modeling of unseen terms

The estimation of word duration is an integral component of PPM search. In its most basic form, searching for a keyword consists of sliding a set of windows over the set of phonetic events and the evaluating the log-likelihood of events under the keyword model. Since the duration of a candidate detection is not known a priori, we consider a set of possible candidate duration windows which are drawn from an estimate of the word's duration distribution. In earlier PPM work with TIMIT in [29] and Chapter 3, every keyword had 462 training examples, sufficiently many to use the empirical distribution. For later experiments on the Wall Street Journal (WSJ) corpus, the number of training examples for each keyword was much lower and use of the empirical distribution was infeasible. In its place, we adopted a parametric description of word duration based on the gamma distribution.

Handling words for which zero training examples exist requires an alternative approach, and we considered three. Admittedly crude, our first approach was to compute distributions based solely on the number of phones in a word's canonical dictionary form. We simply pooled all word examples of a given phone count and computed MLE estimates for the gamma distribution parameters. For a second and more sophisticated model, we compiled duration models for all the constituent phones. Then, utilizing a technique similar to the Monte Carlo method in Section 5.1.2, we constructed Monte Carlo samples of word duration by sampling from the distributions of the constituent phone duration models and

then estimated MLE gamma parameters from the Monte Carlo word duration examples. Clearly, this model failed to capture any dependencies due to phonetic context on phone duration.

The identical problem was addressed in Section 5.1.3 using a classification and regression tree (CART) approach inspired by text-to-speech synthesis work. In that work, phone duration models were estimated from the pool of examples at each node of the regression tree. Here, we opted for a simpler method to incorporate phonetic context. Our goal was to estimate phone duration models for all phonetic contexts as permitted by the number of examples available. We began by collecting pools of duration examples for each trigram phone context. Of course, many of the $O(40^3)$ possible combinations appear relatively infrequently, so if a context contained fewer than 100 examples, we backed off to the corresponding bigram phone context (and likewise from bigram to unigram). Having established the pools of examples, we then estimated gamma parameters of the duration model of each context. Finally, the estimation of a word duration model proceeded as before with Monte Carlo word duration examples constructed from these context-dependent phone duration models.

To evaluate these three approaches, we considered 230 hours of Switchboard data partitioned into two folds. Assessment was based on computing the likelihood of the word durations observed in one data fold based on training data from the opposite fold using each modeling approach. These results are depicted in Figure 7.1 which shows average word likelihood as a function of word phone count. The context-dependent estimation approach (labeled Monte Carlo (3g) in Figure 7.1) proved to be the best and was adopted

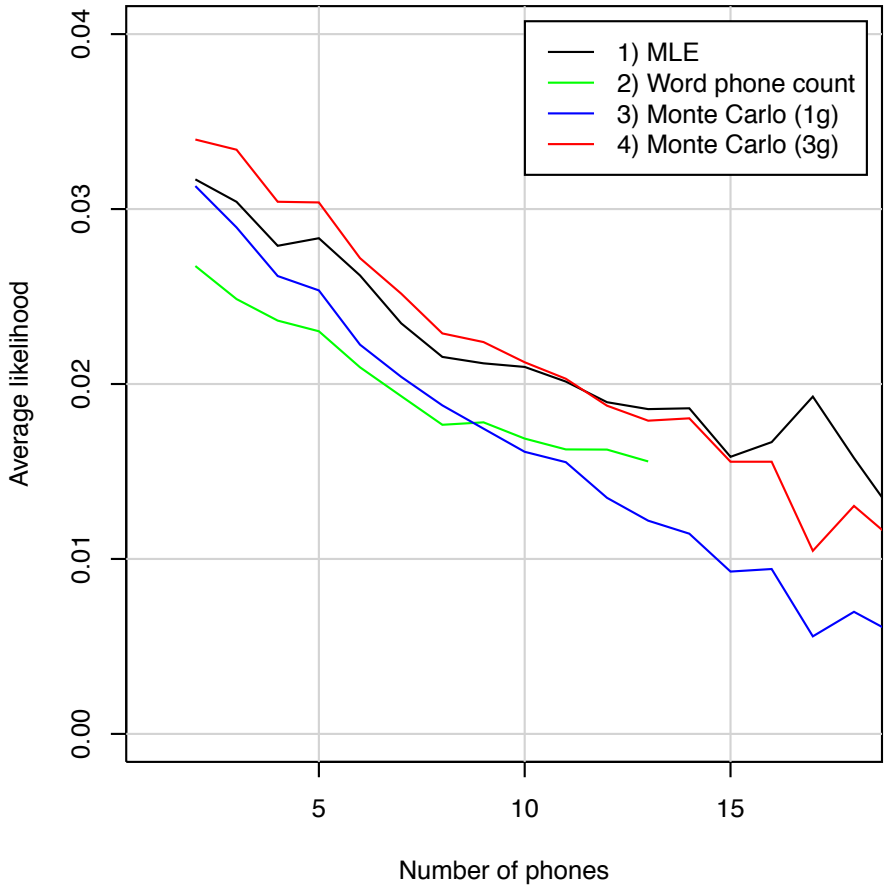

Figure 7.1: Evaluation of word duration modeling approaches. Word duration models were estimated on half the development data and likelihood was computed for the corresponding word examples on the other half of the data. Likelihood was averaged over all words of a given phone count for each of the four modeling approaches.

for all subsequent experiments.

7.1.3 Score normalization

A critical element in properly assessing detections is the conversion from a detection score into the estimated probability of a detection. PPM keyword detections are marked at the local maxima of the detection function (a log-likelihood ratio) as detailed in Section 2.3. A suitable cutoff point for reliable detections varies with the number of

phones in a word. In previous evaluations on TIMIT and WSJ datasets, keyword search performance was reported in terms of average figure of merit and an absolute detection threshold was not required. For the 2006 NIST STD evaluation, the performance metric is ATWV which requires the specification of a uniform decision threshold and a binary decision associated with each putative detection. To map PPM detection scores to a detection probability, we trained a log-linear model using keyword detections from a comparable STD experiment on Switchboard development data. In addition to PPM score, the model also used the logarithm of the keyword duration as an input parameter. These estimates of detection probability also enabled us to calculate *expected counts* of search terms which is necessary for the use of term-specific thresholding in ATWV calculation as described in [74].

7.2 Experiments

Prior to testing on the 2006 STD evaluation data, we conducted extensive developmental work on a 230 hour portion of the Switchboard corpus in order to assess the methods described in the previous section (multi-word modeling, duration modeling of unseen terms and score normalization). We created a Switchboard term list with a composition roughly the same as the 2006 STD evaluation term list in percentages of multi-word terms. For acoustic models, we trained 5-layer deep neural networks to estimate posterior probabilities for 40 phonetic classes, and used them for all subsequent experiments. The Switchboard audio was transformed into 476-dimensional FDLP-M feature vectors [75] and subsequently used to train 5 multilayer perceptrons each of size $476 \times 1500 \times 1500 \times 1500 \times 40$ using 5-fold cross validation training. We then processed the phone posterior data into phonetic events

using phonetic matched filters as described in [67]. Finally, the data was partitioned into two data folds for PPM training and evaluation.

Having completed developmental work on Switchboard, we then performed a series of trials using the 2006 NIST STD evaluation data, and the resulting XML detection list was scored using the original NIST `STDEval` tools. STD results are reported at the bottom of Table 7.2 for ppm4 multi-word models (see description in Table 7.1) along with the results of systems in the original 2006 evaluation. In addition to STD performance, we also provide data on system processing requirements. Additionally, in Table 7.3 we provide system hardware descriptions and processor benchmark data.

7.2.1 Reference systems

To provide context for the PPM system performance, we have included the results from notable LVCSR and phonetic STD systems in the 2006 evaluation (available at [76]). Overall, BBN fielded the top performing entry in the category of English conversational telephone speech (CTS) achieving an ATWV of 0.8335 [74]. The structure of BBN's system consisted of a large-vocabulary, HMM-based speech recognition system to process audio into deep word lattices upon which word posterior probabilities were estimated and a word index was generated. Multi-word term detections were determined by locating sequences of constituent words in the index that satisfied ordering and timing constraints. A key advantage of the BBN system over similar LVCSR entries came from the determination of an optimal detection threshold for each term using the expected term counts from word posterior probability estimates. Another notable entrant was the LVCSR system from IBM which achieved an ATWV of 0.7392 [70]. Both of these entries benefited tremendously from

the presence of a large language model, which provided better estimates of word posterior probabilities (especially for short words) compared with systems that relied on phonetic likelihoods alone.

In contrast to the LVCSR systems, we also present two phonetic-based systems from Brno University of Technology (BUT) and Queensland University of Technology (QUT). The top performing phonetic system fielded by BUT achieved an ATWV of 0.2977. In this system the acoustic models, trained on 277 hours of primarily Switchboard data [77], were the same used in BUT's LVCSR-based primary system except that the decoding produced phoneme lattices using a phoneme bigram language model. Locating candidate detections was performed by converting the search term into a phonetic sequence using grapheme-to-phoneme tool and then obtaining candidate sequences of overlapping phoneme trigrams from an inverted index of the phone lattice. Next, candidate sequence scores were derived from the ratio of the likelihood of the term's phone sequence to the likelihood of the best path in the phone lattice [78].

The QUT system was also based on phonetic lattice search and it yielded an ATWV of 0.0873. As described in [79], tied-state triphone HMM acoustic models were constructed using PLP acoustic features with a bigram phone language model to generate phonetic lattices. Next, a hierarchical index of the phone sequences and broad phone class (vowels, nasals, etc.) sequences was constructed. Query terms were converted into phonetic sequences, and then a technique termed Dynamic Match Lattice Spotting (DMLS) [80] returned putative detections of the sequences in the lattice using minimum edit distance to allow for phonetic substitutions.

In terms of performance, the PPM approach to STD falls in between that of BUT and QUT's phonetic-based entries. The QUT system accomplishes relatively fast lattice-based search, however, we observed that the inherently sparse representation of the PPM system permits it to search 8 times faster than DMLS with more than twice the accuracy (note: this value has been normalized based on relative processor speed benchmarks in Table 7.3). On the other hand, the BUT approach trades speed for accuracy and achieves the best ATWV for phonetic-base systems. Nonetheless, our PPM results are 75% of BUT's accuracy while operating 400-times faster (also normalized) with a significantly smaller footprint.

system	BBN	IBM	BUT	QUT	PPM t02 (GPU)	PPM a07 (non-GPU)
type	LVCSR	LVCSR	phonetic	phonetic	ppm	ppm
Indexing time (HP/HS)	16.109	7.563	86.823	18.088	**0.058**	0.610
Search speed (sec.P/HS)	**0.0014**	0.0041	13.5489	0.3300	0.0107	0.0303
Index Memory Usage (MB)	2,829.39	1,653.43	2,180.91	1,274.66	—	—
Search Memory (MB)	130.34	269.13	2.42	468.64	—	—
Index processing (HP)	48.20	22.63	259.81	54.13	**0.17**	1.83
Index size (MB)	1.17	0.98	1,528.23	1,670.52	**0.47**	0.47
ATWV	**0.8335**	0.7392	0.2977	0.0873	0.2180	0.2180

Table 7.2: A comparison of NIST STD 2006 evaluation system processing resources and detection accuracy for English conversational telephone speech. For all systems, the total hours of speech (HS) is 2.99 hours.

system	BBN	IBM	BUT	QUT	PPM t02 (GPU)	PPM a07 (non-GPU)
CPU	4 CPU Intel Xenon 3.40 GHz	4 CPU Intel Xenon 3.06 GHz	various	1 CPU Intel Pentium 4 3.00 GHz	12 CPU Intel i7-3930K 3.20 GHz	24 CPU Intel Xeon E7450 2.40 GHz
L2 Cache (KB)	1024	512	various	512	12288	12288
OS	Linux 2.4.21	Linux 2.4.30	Linux 2.6.16	Linux 2.4.28	Linux 3.2.0	Linux 3.2.0
C compiler (gcc)	v3.2.3	v3.4.4	v3.4.6	v3.3.2	v4.7.2	v4.7.2
libc	ld-2.3.2.so	ld-2.3.2.so	libc-2.3.4.so	ld-2.3.2.so	libc-2.13.so	libc-2.13.so
nbench performance						
memory index	14.497	9.946	13.967	13.459	**43.85**	27.498
integer index	9.585	9.658	11.369	9.176	**40.317**	18.921
floating-point index	18.371	16.411	19.622	18.787	**59.443**	33.399

Table 7.3: NIST 2006 STD evaluation system hardware descriptions and processor benchmarks.

7.2.2 System description and processing resources

In addition to detection results, the 2006 STD evaluation also required participants to report resource and processing utilization for both indexing and search. In general, processing time is roughly 10 times slower than real time for producing LVCSR word lattices. Phonetic lattices contain significantly more connections and require even more processing time. In the PPM system, what we call an "index" is just the collection of phonetic events. In addition to being very compact, its creation is a relatively straightforward process of feature extraction, MLP forward-pass, and matched filtering of the resulting phone posteriorgrams. The extraction of phonetic events from audio can be accomplished at roughly 17 times faster real time. We should note that in the phonetic event production pipeline, only the MLP software currently takes advantage of the GPU; feature extraction and filtering code is not currently GPU aware. Table 7.2 shows both GPU and non-GPU performance.

For search, both LVCSR systems achieve very fast search times thanks to the inverted word index. Searching a phonetic lattice is a more complex endeavor [78, 80]. The BUT triphone lattice is three orders of magnitude larger than its corresponding LVCSR word lattices and search is three orders of magnitude slower. The DMLS approach in the QUT phonetic system is somewhat faster. The PPM search, while fairly fast, is still basically a linear search. However, phonetic events represent an extremely sparse representation of speech, and search speed benefits because of the tiny index size. The quoted index size of 492KB for 3 hours of speech represents an uncompressed index (compression such as `gzip` provides a further 20% reduction in this case).

The extremely compact size of the PPM index is a significant advantage of our

approach. It permits our system to consider extremely large volumes of audio data without being overwhelmed by either processing time or storage considerations. Additionally, the small memory footprint required by phonetic events will permit our approach to be ported to multiprocessor devices (GPU) enabling extremely fast parallel search.

In evaluating the relative system performance, it is necessary to consider the computation speed of the systems at the time of the original evaluation. To offer some perspective on the relative speed, we present system descriptions and benchmarks in Table 7.3. Overall the `t02` GPU machine is roughly 3-4 times faster than 2006-era machines and `a07` is approximately twice as fast.

7.3 Conclusions

In this chapter we have addressed many of technical challenges required to enable the PPM system to accomplish spoken term detection. Furthermore, this study provides the first side-by-side comparison of a PPM system for spoken term detection in the context of other well documented systems on a standard evaluation dataset. Unquestionably, LVCSR-based systems will outperform systems that do not currently benefit from a language model. Yet, we clearly observe that PPM keyword search achieves performance results competitive with other state-of-the-art phonetic-based systems. More significantly, PPM keyword search accomplishes this while requiring a fraction of the computational and storage resources.

Chapter 8

Conclusions

The body of work contained in this dissertation records the many significant improvements to various components of the point process model for keyword search which enabled its evolution from proof-of-concept experiments on TIMIT into a fast, lightweight spoken term detection system for conversational speech that is competitive with other well-documented phonetic STD approaches. Underlying the modeling decisions which give rise the point process framework is the notion that speech is the product of the physical movement of articulators, and thus robustly coded by temporal relations between distinct acoustic events. While the use of phonetic events is a departure from the original conception of acoustic landmarks and distinctive binary features of [18], it significantly facilitates system development by enabling compatibility with a wealth of existing phonetic recognition systems and labeled training resources while preserving the essential whole-word temporal structure of distinct events in time. Studies of human physiology have a well-developed understanding of the spectral-resolving ability of lower levels of the auditory system, and this

knowledge has long been reflected in speech feature design. However, the use of short term analysis windows that is common to all HMM-based recognizers has no parallel in human speech perception [81]. On the other hand, strong evidence exists for the preeminence of temporal cues in human recognition from the robustness of human recognition to corruption of spectral information [33] to the inability of children with basic temporal processing deficits to develop language skills normally [38]. The HMM framework is mathematically tractable because of the assumption of conditional independence, but this condition renders HMMs ill-suited to model long-term correlations between variables.

The first aspect of the point process model considered Chapter 3 was the derivation of phonetic events. As an alternative to drawing events from the local maxima of unfiltered phone posterior trajectories, we instead considered trajectories smoothed using phone-specific matched filters. Filtering had the effect of integrating posterior estimates over a long windows. A secondary benefit was the resulting reduction in the number of events towards a minimal representation consisting of one event per phone, an attribute which greatly simplified parametric modeling approaches developed in Chapter 4. A necessary component of phonetic event selection is the determination of an appropriate event threshold which we addressed using a mutual information based event selection metric. Experiments detailed in Chapter 3 demonstrated the use of events derived from filtered posteriorgrams reduced the number of events by 40% and simultaneously improved average keyword search performance by 23% [67].

As documented in [29], a basic deficiency limiting the utility of the point process modeling of keywords was the need for numerous keyword training examples. The

parametric approaches developed in Chapter 4 addressed this limitation. An examination of phonetic event distributions in length-normalized word examples suggested that they could be properly modeled using Gaussian distributions. Further, this finding suggested that variation in parameter estimates arising from insufficient examples could be mitigated through Bayesian estimation techniques and a natural prior estimate could be derived from the word's phonetic form. An equally important finding was the necessity of introducing phonetic variation into the models which was also incorporated in the Bayesian approach. Besides TIMIT experiments, we further evaluated these techniques on the significantly larger Wall Street Journal corpus and demonstrated a 97% relative improvement in keyword search performance when limited keyword examples were available [68]. The modeling techniques developed in Chapter 4 were fundamental to all subsequent work.

Substantial improvements in keyword search performance were realized in Chapter 4 through a Bayesian approach to model parameter estimation using the simplest prior model of phonetic timing. In Chapter 5 we examined several improved methods of estimating prior models using techniques inspired by text-to-speech synthesis. Applying a Monte Carlo approach, we estimated the means and variances of phonetic timing distributions by sampling examples of words synthesized from individual phone duration distributions. To capture contextual dependencies between phones, we adapted a CART model to learn context dependent distributions. Ultimately, these more complex approach to estimating phone-timing distributions yielded a modest 4.2% relative improvement in average FOM compared to using a simple dictionary models [82].

Distinct from frame-by-frame, dense representations and Viterbi decoding, the

PPM keyword system is inherently lightweight due to its sparse representation of the speech signal and had the potential of achieving extremely fast search speeds. In Chapter 6, we reformulated the process of keyword detection to capitalize on this sparse phonetic event representation. Factoring the keyword detection function revealed that it could be simplified into a sum over the product of phonetic event counts and a score matrix. Further, it facilitated the determination of an upperbound on the keyword detection function. We next demonstrated how the evaluation of the detection function could be inverted; instead of a frame-by-frame sliding evaluation we proceed event-by-event and only accumulate changes in the score. These refinements resulted in a factor of 50 times improvement in decoding speed [83].

In Chapter 7 we drew upon all of the advances previously introduced in order to benchmark the performance of the point process model relative to other phonetic keyword search systems on the NIST 2006 STD evaluation. In addition to addressing the modeling of multi-word terms, we also introduced improvements to word duration modeling and detection score normalization necessary for ATWV calculation. Notably, these experiments marked the first trial of a PPM system on conversational telephone speech data. The NIST 2006 STD evaluation results showed that PPM keyword search performs on par with other state-of-the-art phonetic-based systems, furthermore it accomplishes this significantly faster and while requiring a fraction of the computational and storage resources.

8.1 Future directions

In Chapter 6 we demonstrated the potential of PPM methods to achieve extremely fast search speeds, and these techniques were employed in the STD evaluation reported in Chapter 7. The end-to-end processing time numbers listed in Table 7.2 were a concatenation of the times for several sequential operations from feature extraction to MLP forward pass to PPM keyword search. While only the MLP computations currently benefit from the use of GPUs, we believe the point process keyword search could also benefit tremendously. In current implementations, keyword search is accomplished sequentially one word at a time but every search operates on the same set of phonetic events and only differs in the word model being evaluated. The process is a natural candidate for parallelization since the algorithm is simple, requiring mainly addition operations, and does not have a large memory footprint. The ideal implementation for PPM search would be to simultaneously evaluate all keywords in a single pass through the data.

The improvements to point process modeling, decoding and other enhancements have enabled the PPM system's viability relative to other phonetic systems while requiring significantly less processing overhead. Yet, common to all phonetic-based systems, a persistent gap in performance still exists between phonetic and LVCSR approaches. Without question, the source of this discrepancy is the tremendous power of language modeling in the estimation of the likelihoods of alternative decodings. Methods of incorporating language modeling into the estimation of PPM word detection scores is an obvious area for future investigations. This effort would naturally benefit from the simultaneous search for all keywords suggested in the previous paragraph. Additionally, the decoding algorithm

would likely benefit from changing the denominator term of the likelihood ratio. Instead of considering the likelihood of a background model, we could instead consider likelihood relative to all other words.

Bibliography

[1] F. Jelinek, L. Bahl, and R. Mercer, "Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech," *Information Theory, IEEE Transactions on*, vol. 21, no. 3, pp. 250 – 256, May 1975.

[2] J. Baker, "The DRAGON System–An Overview," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 23, no. 1, pp. 24–29, Feb 1975.

[3] F. Jelinek, "Fast Sequential Decoding Algorithm Using a Stack," *IBM Journal of Research and Development*, vol. 13, no. 6, pp. 675–685, 1969.

[4] J. Proakis and M. Salehi, *Digital Communications*, 5th ed. McGraw-Hill, New York, 2008.

[5] M. R. Schroeder, "Recognition of complex acoustic signals," *Life Sciences Research Report*, vol. 5, no. 324, p. 130, 1977.

[6] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, no. 4, p. 1738, 1990.

[7] D. Jurafsky, J. H. Martin, A. Kehler, K. Vander Linden, and N. Ward, *Speech and*

Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. MIT Press, 2000, vol. 2.

[8] R. P. Lippmann, "Speech recognition by machines and humans," *Speech Communication*, vol. 22, no. 1, pp. 1–15, 1997.

[9] F. Seide, G. Li, and D. Yu, "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks." in *Proceedings of INTERSPEECH*. ISCA, 2011, pp. 437–440.

[10] O. Scharenborg, "Reaching over the gap: A review of efforts to link human and automatic speech recognition research," *Speech Communication*, vol. 49, no. 5, pp. 336–347, 2007.

[11] C. Chelba, "Speech and natural language: Where are we now and where are we headed?" Mobile Voice Conference, 2013.

[12] F. Wessel and H. Ney, "Unsupervised training of acoustic models for large vocabulary continuous speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 1, pp. 23–31, Jan 2005.

[13] L. Lamel, J.-L. Gauvain, and G. Adda, "Unsupervised acoustic model training," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 1, 2002, pp. I–877–I–880.

[14] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling

in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[15] R. K. Moore, "A comparison of the data requirements of automatic speech recognition systems and human listeners." in *Proceedings of EUROSPEECH*, 2003, pp. 2582–2584.

[16] F. Jelinek, *Statistical methods for speech recognition*. MIT Press, 1997.

[17] R. Jakobson, G. Fant, and M. Halle, *Preliminaries to speech analysis. The distinctive features and their correlates*. MIT Press, Cambridge MA, 1951.

[18] K. N. Stevens, "Toward a model for lexical access based on acoustic landmarks and distinctive features," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1872–1891, 2002.

[19] C. Y. Espy-Wilson, "A feature-based semivowel recognition system," *The Journal of the Acoustical Society of America*, vol. 96, no. 1, pp. 65–72, 1994.

[20] A. Juneja and C. Espy-Wilson, "Segmentation of continuous speech using acoustic-phonetic parameters and statistical learning," in *Neural Information Processing, 2002. ICONIP'02. Proceedings of the 9th International Conference on*, vol. 2. IEEE, 2002, pp. 726–730.

[21] ——, "Speech segmentation using probabilistic phonetic feature hierarchy and support vector machines," in *Neural Networks, 2003. Proceedings of the International Joint Conference on*, vol. 1. IEEE, 2003, pp. 675–679.

[22] ——, "A probabilistic framework for landmark detection based on phonetic features

for automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 123, p. 1154, 2008.

[23] V. Zue, J. Glass, M. Phillips, and S. Seneff, "The MIT SUMMIT speech recognition system: A progress report," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1989, pp. 179–189.

[24] J. R. Glass, "A probabilistic framework for segment-based speech recognition," *Computer Speech & Language*, vol. 17, no. 2, pp. 137–152, 2003.

[25] M. Hasegawa-Johnson, J. Baker, S. Borys, K. Chen, E. Coogan, S. Greenberg, A. Juneja, K. Kirchhoff, K. Livescu, S. Mohan *et al.*, "Landmark-based speech recognition: Report of the 2004 Johns Hopkins Summer Workshop," Johns Hopkins University, Tech. Rep., 2005.

[26] M. Lehtonen, P. Fousek, and H. Hermansky, "Hierarchical approach for spotting keywords," in *2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, Edinburgh, United Kingdom, July 2005.

[27] H. Hermansky, P. Fousek, and M. Lehtonen, "The role of speech in multimodal human-computer interaction," in *Text, Speech and Dialogue*, ser. Lecture Notes in Computer Science, V. Matoušek, P. Mautner, and T. Pavelka, Eds. Springer Berlin Heidelberg, 2005, vol. 3658, pp. 2–8.

[28] A. Jansen and P. Niyogi, "Point process models for event-based speech recognition," *Speech communication*, vol. 51, no. 12, pp. 1155–1168, 2009.

[29] ——, "Point process models for spotting keywords in continuous speech," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 8, pp. 1457 – 1470, Nov 2009.

[30] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, 1980.

[31] D. Robinson and R. Dadson, "A redetermination of the equal-loudness relations for pure tones," *British Journal of Applied Physics*, vol. 7, pp. 161–181, 1956.

[32] S. S. Stevens, "On the psychophysical law." *Psychological review*, vol. 64, no. 3, p. 153, 1957.

[33] R. V. Shannon, F.-G. Zeng, V. Kamath, J. Wygonski, and M. Ekelid, "Speech recognition with primarily temporal cues," *Science*, vol. 270, no. 5234, pp. 303–304, 1995.

[34] J. C. R. Licklider and I. Pollack, "Effects of differentiation, integration, and infinite peak clipping upon the intelligibility of speech," *The Journal of the Acoustical Society of America*, vol. 20, no. 1, pp. 42–51, 1948.

[35] R. P. Lippmann, "Accurate consonant perception without mid-frequency speech energy," *Speech and Audio Processing, IEEE Transactions on*, vol. 4, no. 1, p. 66, 1996.

[36] T. Houtgast and H. J. Steeneken, "A review of the MTF concept in room acoustics and its use for estimating speech intelligibility in auditoria," *The Journal of the Acoustical Society of America*, vol. 77, no. 3, pp. 1069–1077, 1985.

[37] H. Hermansky, "Should recognizers have ears?" *Speech Communication*, vol. 25, no. 1–3, pp. 3–27, 1998.

[38] P. Tallal, S. Miller, and R. H. Fitch, "Neurobiological Basis of Speech: A Case for the Preeminence of Temporal Processing," *Annals of the New York Academy of Sciences*, vol. 682, no. 1, pp. 27–47, 1993.

[39] U. Goswami, J. Thomson, U. Richardson, R. Stainthorp, D. Hughes, S. Rosen, and S. K. Scott, "Amplitude envelope onsets and developmental dyslexia: A new hypothesis," *Proceedings of the National Academy of Sciences*, vol. 99, no. 16, pp. 10 911–10 916, 2002.

[40] S. E. Levinson, "Continuously variable duration hidden Markov models for automatic speech recognition," *Computer Speech & Language*, vol. 1, no. 1, pp. 29–45, 1986.

[41] D. Burshtein, "Robust parametric modeling of durations in hidden Markov models," *Speech and Audio Processing, IEEE Transactions on*, vol. 4, no. 3, pp. 240–242, 1996.

[42] M. Ostendorf, V. Digalakis, and O. Kimball, "From HMM's to segment models: a unified view of stochastic modeling for speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 4, no. 5, pp. 360–378, Sep 1996.

[43] J. D. Ferguson, "Variable duration models for speech," in *Proc. Symposium on the application of hidden Markov models to text and speech*, 1980, pp. 143–179.

[44] M. Russell and A. Cook, "Experimental evaluation of duration modelling techniques for automatic speech recognition," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'87.*, vol. 12. IEEE, 1987, pp. 2376–2379.

[45] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden Markov modeling for speaker-independent word spotting," in *Acoustics, Speech and Signal Processing (ICASSP), 1989 IEEE International Conference on*, 1989, pp. 627–630.

[46] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate Bayesian a posteriori probabilities," *Neural Computation*, vol. 3, no. 4, pp. 461–483, 1991.

[47] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing: Algorithms, Architectures and Applications*. Springer-Verlag, 1990, pp. 227–236.

[48] N. Morgan, H. Hermansky, H. Bourlard, P. Kohn, and C. Wooters, "Continuous Speech Recognition Using PLP Analysis with Multilayer Perceptrons," in *Proceedings of Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, May 1991, pp. 49–21.

[49] G. Sivaram and H. Hermansky, "Multilayer perceptron with sparse hidden outputs for phoneme recognition," in *Proceedings of Acoustics, Speech, and Signal Processing (ICASSP), 2011 International Conference on*, May 2011, pp. 5336–5339.

[50] J. Pinto, S. Garimella, M. Magimai-Doss, H. Hermansky, and H. Bourlard, "Analysis of MLP-based hierarchical phoneme posterior probability estimator," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 2, pp. 225–241, 2011.

[51] M. Lehtonen, P. Fousek, and H. Hermansky, "Hierarchical approach for spotting keywords," IDIAP Research Report, IDIAP-RR 05-41, Tech. Rep., 2005.

[52] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue,

"TIMIT Acoustic-Phonetic Continuous Speech Corpus," CDROM, Philadelphia, PA, 1993.

[53] K. Lee and H. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1989.

[54] C.-H. Lee, B.-H. Juang, F. Soong, and L. Rabiner, "Word recognition using whole word and subword models," in *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, 1989, pp. 683–686.

[55] M. Ostendorf, V. Digalakis, and O. Kimball, "From HMM's to segment models: a unified view of stochastic modeling for speech recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 4, no. 5, pp. 369–378, 1996.

[56] A. Cutler, D. Dahan, and W. Van Donselaar, "Prosody in the comprehension of spoken language: A literature review," *Language and Speech*, vol. 40, no. 2, pp. 141–201, 1997.

[57] J. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 2, pp. 291–298, 1994.

[58] M. H. DeGroot, *Optimal statistical decisions*. John Wiley & Sons, Inc., 1970.

[59] K. P. Murphy, "Conjugate Bayesian analysis of the Gaussian distribution," University of British Columbia, Technical report, 2007.

[60] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR

corpus," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.

[61] G. Zweig, P. Nguyen, D. Van Compernolle, K. Demuynck, L. Atlas, P. Clark, G. Sell, M. Wang, F. Sha, H. Hermansky, D. Karakos, A. Jansen, S. Thomas, S. G.S.V.S., S. Bowmand, and J. Kao, "Speech recognition with segmental conditional random fields: A summary of the JHU CLSP 2010 summer workshop," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, 2011, pp. 5044–5047.

[62] D. H. Klatt, "Linguistic uses of segmental duration in English: Acoustic and perceptual evidence," *The Journal of the Acoustical Society of America*, vol. 59, no. 5, pp. 1208–1221, 1976.

[63] J. P. Van Santen, "Assignment of segmental duration in text-to-speech synthesis," *Computer Speech & Language*, vol. 8, no. 2, pp. 95–128, 1994.

[64] M. D. Riley, "Tree-based modelling for speech synthesis," in *The ESCA Workshop on Speech Synthesis*, 1990, pp. 229–232.

[65] Carnegie-Mellon University, "Carnegie-Mellon Pronouncing Dictionary," July 2013. [Online]. Available: http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[66] J. Pinto, B. Yegnanarayana, H. Hermansky, and M. Magimai-Doss, "Exploiting contextual information for improved phoneme recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2008 IEEE International Conference on*. IEEE, 2008, pp. 4449–4452.

[67] K. Kintzley, A. Jansen, and H. Hermansky, "Event selection from phone posteriorgrams using matched filters," in *Proceedings of INTERSPEECH*, 2011, pp. 1905–1908.

[68] ——, "MAP estimation of whole-word acoustic models with dictionary priors," in *Proceedings of INTERSPEECH*, 2012.

[69] D. Can and M. Saraclar, "Lattice indexing for spoken term detection," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 8, pp. 2338–2347, 2011.

[70] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2007, pp. 615–622.

[71] K. Thambiratnam and S. Sridharan, "Rapid yet accurate speech indexing using dynamic match lattice spotting," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 1, pp. 346–357, 2007.

[72] J. G. Wilpon, L. R. Rabiner, C.-H. Lee, and E. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 38, no. 11, pp. 1870–1878, 1990.

[73] National Institute of Standards and Technology, "The Spoken Term Detection (STD) 2006 Evaluation Plan," Sept 2006. [Online]. Available: http://www.itl.nist.gov/iad/mig//tests/std/2006/docs/std06-evalplan-v10.pdf

[74] D. R. Miller, M. Kleber, C.-L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection." in *Proceedings of INTERSPEECH*, 2007, pp. 314–317.

[75] S. Ganapathy, S. Thomas, and H. Hermansky, "Comparison of modulation features for phoneme recognition," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 2010, pp. 5038–5041.

[76] National Institute of Standards and Technology, "Results of the Spoken Term Detection (STD) 2006 Evaluation," Dec 2006. [Online]. Available: http://www.itl.nist.gov/iad/mig/tests/std/2006/pubdata/std06_results_20061207.tgz

[77] T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiat, M. Lincoln, J. Vepa, and V. Wan, "The AMI meeting transcription system: Progress and performance," in *Machine Learning for Multimodal Interaction*, ser. Lecture Notes in Computer Science, S. Renals, S. Bengio, and J. Fiscus, Eds. Springer Berlin Heidelberg, 2006, vol. 4299, pp. 419–431.

[78] L. Burget, H. Cernocky, M. Fapso, M. Karafiat, P. Matejka, P. Schwarz, and I. Szoke, "Indexing and search methods for spoken documents," in *Text, Speech and Dialogue*, ser. Lecture Notes in Computer Science, P. Sojka, I. Kopecek, and K. Pala, Eds. Springer Berlin Heidelberg, 2006, vol. 4188, pp. 351–358.

[79] R. G. Wallace, R. J. Vogt, and S. Sridharan, "A phonetic search approach to the 2006 NIST Spoken Term Detection Evaluation," in *Proceedings of INTERSPEECH*. ISCA, 2007, pp. 2385–2388.

[80] K. Thambiratnam and S. Sridharan, "Dynamic match phone-lattice searches for very fast and accurate unrestricted vocabulary keyword spotting," in *Acoustics, Speech, and*

Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on, vol. 1, 2005, pp. 465–468.

[81] H. Bourlard, H. Hermansky, and N. Morgan, "Towards increasing speech recognition error rates," *Speech Communication*, vol. 18, no. 3, pp. 205–231, 1996.

[82] K. Kintzley, A. Jansen, and H. Hermansky, "Text-to-Speech Inspired Duration Modeling for Improved Whole-Word Acoustic Models," in *Proceedings of INTERSPEECH*. ISCA, 2013.

[83] K. Kintzley, A. Jansen, K. Church, and H. Hermansky, "Inverting the point process model for fast phonetic keyword search," in *Proceedings of INTERSPEECH*. ISCA, 2012.

Vita

Keith Kintzley received his Bachelor of Science degree in Electrical Engineering from the United States Naval Academy, Annapolis, Maryland in 1993, and Master of Science degree in Electrical Engineering from Texas A&M University, College Station, Texas in 1995.

In 2006, he returned to the Electrical & Computer Engineering Department at the U.S. Naval Academy as a military instructor. In 2008 he was selected for the Navy's Permanent Military Professor Program. In and he began his doctoral work in Center for Language and Speech Processing (CLSP) at the Johns Hopkins University in 2009. He completed his PhD in Electrical Engineering in 2014 and is presently assigned as a Permanent Military Professor at the United States Naval Academy in Annapolis, Maryland.

His research interests include speech recognition, acoustic modeling, spoken term detection and machine learning.