Parts-based Models and Local Features for Automatic Speech Recognition

by

Kenneth Thomas Schutte

B.S., University of Illinois at Urbana-Champaign (2001)
S.M., Massachusetts Institute of Technology (2003)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2009

Author .... [illegible]
Department of Electrical Engineering and Computer Science
May 22, 2009

Certified by .... [illegible]
James R. Glass
Principal Research Scientist
Thesis Supervisor

Accepted by .... [illegible]
Terry P. Orlando
Chairman, Department Committee on Graduate Theses

Parts-based Models and Local Features for Automatic Speech Recognition

by

Kenneth Thomas Schutte

Submitted to the Department of Electrical Engineering and Computer Science
on May 22, 2009, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

While automatic speech recognition (ASR) systems have steadily improved and are now in widespread use, their accuracy continues to lag behind human performance, particularly in adverse conditions. This thesis revisits the basic acoustic modeling assumptions common to most ASR systems and argues that improvements to the underlying model of speech are required to address these shortcomings.

A number of problems with the standard method of hidden Markov models (HMMs) and features derived from fixed, frame-based spectra (e.g. MFCCs) are discussed. Based on these problems, a set of desirable properties of an improved acoustic model are proposed, and we present a "parts-based" framework as an alternative. The parts-based model (PBM), based on previous work in machine vision, uses graphical models to represent speech with a deformable template of spectro-temporally localized "parts", as opposed to modeling speech as a sequence of fixed spectral profiles. We discuss the proposed model's relationship to HMMs and segment-based recognizers, and describe how they can be viewed as special cases of the PBM.

Two variations of PBMs are described in detail. The first represents each phonetic unit with a set of time-frequency (T-F) "patches" which act as filters over a spectrogram. The model structure encodes the patches' relative T-F positions. The second variation, referred to as a "speech schematic" model, more directly encodes the information in a spectrogram by using simple edge detectors and focusing more on modeling the constraints between parts.

We demonstrate the proposed models on various isolated recognition tasks and show the benefits over baseline systems, particularly in noisy conditions and when only limited training data is available. We discuss efficient implementation of the models and describe how they can be combined to build larger recognition systems. It is argued that the flexible templates used in parts-based modeling may provide a better generative model of speech than typical HMMs.

Thesis Supervisor: James R. Glass
Title: Principal Research Scientist

Acknowledgments

Thanks to my advisor, Jim Glass, for his guidance over the years and for introducing me to the problem of speech recognition. Thanks to my thesis committee, Victor Zue and Tomaso Poggio. I was lucky to have two great internship experiences during my Ph.D years: thanks to Ryan Rifkin at Honda Research Institute and Naoto Iwahashi at ATR. Thanks to all of my fellow SLS members, my friends, and my family. Most importantly, thanks to my mom and my dad for their love and support.

This research was funded in part by Honda and by the T-Party project, a joint research program between MIT and Quanta Computer Inc., Taiwan.

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Introduction

The goal of automatic speech recognition (ASR) is to take a speech recording as input, and produce the text of the words spoken. The need for highly reliable ASR lies at the core of many rapidly growing application areas such as speech interfaces (increasingly on mobile devices) and indexing of audio/video databases for search. While the ASR problem has been studied extensively for over fifty years [72, 23], it is far from solved. There has been much progress and ASR technology is now in widespread use; however, there is still a considerable gap between human and machine performance, particularly in adverse conditions [60, 81].

1.1.1 The need for better models

Despite the significant progress of ASR in the past several decades, surprisingly little has changed in the *model of speech* underlying most current systems. Speech is modeled as a discrete sequence of states, and statistical models are built for these states from short-time spectral measurements. A search procedure finds the best fitting word sequence based on the models' scores at each time step. (A more detailed description of the "standard" approach to ASR is given in Chapter 2.)

For some time, researchers have acknowledged and discussed the problems with this "beads-on-a-string" model of speech [68] and noted problems with the standard acoustic modeling methods [8, 67]. However, most practical improvements in ASR performance have come not from addressing the core modeling assumptions, but various incremental improvements, each providing a moderate decrease in error rates. These include improvements such as adaptation techniques (e.g., maximum likelihood linear regression, MLLR [53]; and vocal tract length normalization, VTLN [52, 95]), discriminative training [65, 71], and improved language modeling [13]. Perhaps most importantly, ASR systems have benefited greatly from general improvements in computer technology. The availability of very large datasets and the ability to utilize them for training models has been very beneficial. Also, with ever increasing computing power, more powerful search techniques can be utilized during recognition.

The view taken in this thesis is that to close the gap between human and machine performance, these incremental improvements are insufficient. Rather, ASR systems will have to improve upon the underlying *model of speech*. As expressed in [8], incremental improvements to standard ASR models is akin to hill-climbing towards a *local maximum* in the landscape of possible methods. Reaching the ultimate goal of ASR – human-level (or beyond) performance in all conditions and on all tasks – will require investigating other regions of this landscape, even if doing so results in back-tracking in progress in the short-term.

1.1.2 The approach of this thesis

To explore alternative acoustic modeling techniques, this thesis attempts to utilize knowledge of acoustic phonetics and human speech perception. We will argue that well known phonetic cues crucial to human speech perception are not modeled effectively in standard ASR systems, and that there is a benefit to model such cues *explicitly*, rather than *implicitly*.

Many of these phonetic cues can be described as patterns in an appropriate time-frequency (T-F) representation. These patterns are often *localized* to a small T-F region, as opposed to spanning the full frequency range. Such cues include phenomena such as formant transitions, bursts in particular frequency bands, and voicing information [87]. Phonetic units (such as phonemes or diphones) often have multiple (redundant) cues, and there exists some typical relationship between them.

The approach advocated in this thesis is to construct models able to capture this kind of information directly. The high-level idea is illustrated in Figure 1-1. Instead of the traditional model of speech as a sequence of spectral profiles (frame-based features and a discrete sequence of states), we seek to model (and detect) phonetic units as a collection of local acoustic-phonetic cues arranged in a flexible configuration. We will argue that more explicit representation of these cues has many benefits, including improved noise robustness and the potential for effective speaker adaptation.

For any alternative acoustic models to compete with the well developed data-driven techniques of current ASR methods, a principled statistical framework for the model is needed. In this thesis, we propose a *parts-based modeling* framework to fill this need. Based on work in machine vision [30] and utilizing the techniques of graphical models, the parts-based models (PBMs) offer a method to implement the approach suggested by the illustration in Figure 1-1. We demonstrate examples on the ISOLET corpus of isolated spoken letters [15], which provides cases of simple, yet easily confusable phonetic distinctions.

1.1.3 Broader implications

As well as offering a framework for novel methods to improve ASR performance (particularly in noise), we argue that the types of models explored in this thesis can help lead to better underlying *generative models of speech* than standard (MFCC/HMM) models. Because they are generative models, HMMs from trained ASR systems can be used to draw samples from the prior distribution which the model defines. This process generally does not result in qualitatively realistic spectrograms. While these ASR systems were not designed for this purpose, it does illustrate the kind of information contained in the model. With PBMs created to directly encode phonetic information in a "schematic" version of a speech spectrogram, samples drawn from its prior do contain the basic phonetic properties of interest.

Furthermore, using models based on phonetic cues can allow ASR systems to concentrate on learning a model of *speech itself*, rather than speech along with the environment in which it occurs (where the environment includes channel, noise, other sound sources, etc.). This intermediate level of phonetic cues can act as a "buffer" between acoustic measurements and abstract phonetic units. Pursuing improvements in these kinds of generative statistical models of speech would not only benefit ASR, but other areas of speech technology as well, including speech synthesis and speech coding.

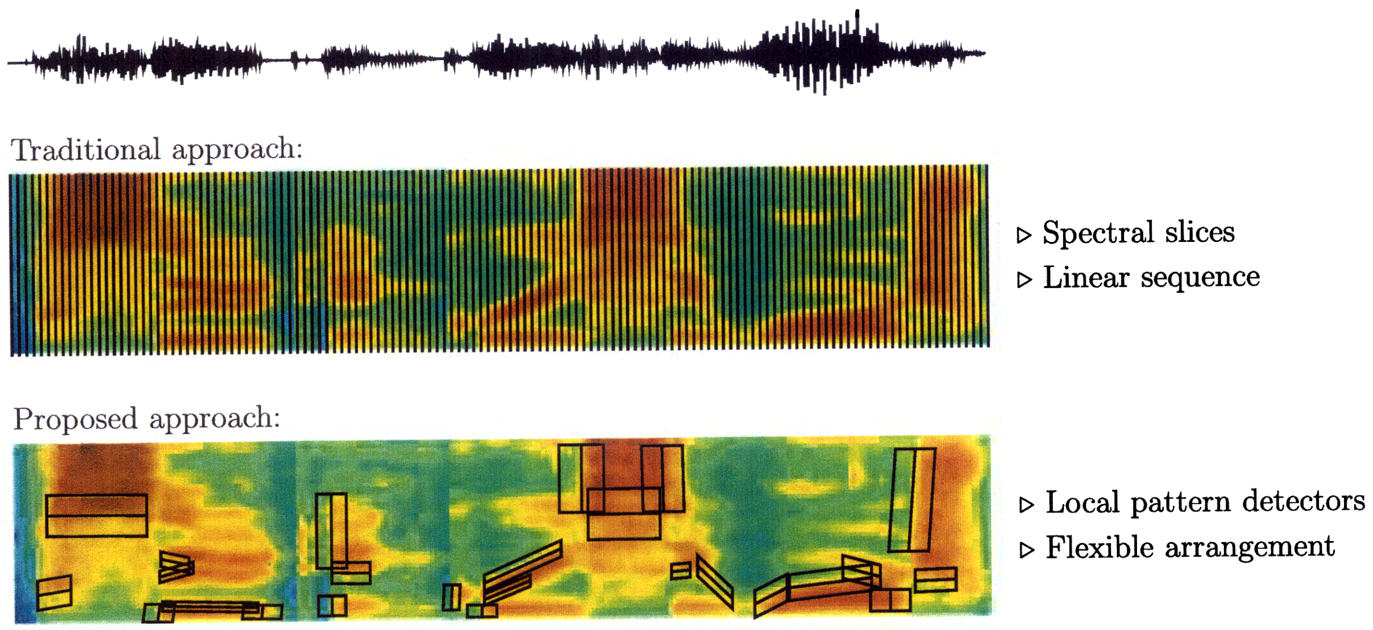


Figure 1-1: High-level illustration of the goals of this thesis. Typical ASR approaches model speech as a linear sequence of states represented by short-time spectral profiles. Our goal is to explore approaches based on collections of local time-frequency pattern detectors.

1.2 Overview

1.2.1 Contributions

In this thesis, we revisit the assumptions underlying many current acoustic models (HMMs and MFCC features) and identify problems which may be limiting the robustness of current ASR technology. We motivate and propose a set of desired features for a new acoustic model: localized time-frequency patterns in a flexible configuration, as opposed to sequences of fixed spectral shapes. We propose and describe a new approach to acoustic modeling using parts-based models (PBMs). We describe examples of applying this model and discuss efficient methods for its implementation. We demonstrate the ability of parts-based models to make robust phonetic distinctions, and describe ways it can be used to build larger ASR systems.

1.2.2 Outline

The remainder of this thesis is organized as follows.

Chapter 2: Standard acoustic modeling for ASR

The standard approach to ASR is described, and we review a set of common techniques for acoustic modeling: feature extraction with Mel-frequency cepstral coefficients (MFCCs), acoustic scoring with Gaussian mixture models (GMMs), and sequence modeling with hidden Markov models (HMMs).

Chapter 3: Local features and the desired acoustic modeling approach

We motivate the desire for local features by looking at problems with standard acoustic modeling, and contrasting current techniques with knowledge of phonetics and known cues used for phonetic discrimination. From these properties, a qualitative description of the desired acoustic model is described: representation of phonetic units with a set of localized features in a deformable pattern. The chapter concludes with a review of previous work having similar goals.

Chapter 4: Parts-based models for speech

We describe the proposed parts-based framework for acoustic modeling. Two examples are shown in detail to highlight the variety of ways to apply the general modeling approach. The relationship to standard HMMs and segment-based models is discussed. We also discuss some implementation details: the choice of signal representation, the use of integral images for fast acoustic scoring, and the max-sum algorithm used for decoding.

Chapter 5: Examples and applications of PBMs

We describe applications of the model to various phonetic discrimination and ASR tasks. It is shown that the model can extract simple phonetic cues such as formant transitions and voice-onset time which are able to robustly make difficult phonetic distinctions with little training data. We describe several other benefits of the parts-based models and discuss two ways which they could be used to build larger recognizers.

Chapter 6: Discussion and future directions

We revisit some previous work to compare to the details of our proposed methods. We discuss a number of directions which future work could address, and end with some higher-level goals and comments.

Appendix A: ISOLET data and baseline recognizer

Appendix A describes the ISOLET corpus, as well as the HMM-based recognizer used as a baseline. The details of the time-frequency representation used throughout the thesis are given.

Appendix B: Graphical models

Appendix B provides a brief overview of the graphical modeling techniques and notation used in this thesis.

Chapter 2

Standard acoustic modeling for ASR

This chapter will describe a "standard" approach for acoustic modeling in automatic speech recognition (ASR) based on hidden Markov models (HMMs). While there have been countless techniques proposed and there is not one standard method, many of the successful approaches can be considered a variation or extension of the basic methods described here. More complete reviews of ASR can be found in the references [72, 41].

2.1 Automatic speech recognition

The ASR problem is typically expressed probabilistically as follows. We are given a sequence of measurements, or observations, $\mathbf{o} = o_1, o_2, \ldots, o_T$, from which we would like to hypothesize a sequence of words, $\mathbf{w} = w_1, w_2, \ldots, w_N$. The desired output, $\mathbf{w}^*$, is the word string with the highest probability, given the observations. This is the *maximum a posteriori* (MAP) decision,

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmax}} P(\mathbf{w}|\mathbf{o}) \tag{2.1}$$

Using Bayes' rule,

$$\begin{aligned} \mathbf{w}^* &= \underset{\mathbf{w}}{\operatorname{argmax}} \frac{p(\mathbf{o}|\mathbf{w})P(\mathbf{w})}{p(\mathbf{o})} & (2.2) \\ &= \underset{\mathbf{w}}{\operatorname{argmax}} p(\mathbf{o}|\mathbf{w})P(\mathbf{w}) & (2.3) \end{aligned}$$

$P(\mathbf{w}) = P(w_1, \ldots, w_N)$ gives the probability of a given word string and is estimated with a *language model*. Language modeling is not discussed in this thesis. We are concerned with the *acoustic modeling* term, $P(\mathbf{o}|\mathbf{w})$, which models how the acoustic measurements, $\mathbf{o}$, relate to the word string, $\mathbf{w}$.

Typical modeling strategies operate on units below the level of words, and there are many methods used to decompose words into sequences of sub-word units. However, for HMM-based recognizers, a word sequence is ultimately reduced to a sequence of *states*. For example, words may be decomposed into triphones (a string of three phones), and the triphones may be decomposed into a sequence of triphone states. The state sequence, $\mathbf{x} = x_1, \ldots, x_T$, represents an additional set of hidden variables which must be marginalized to compute $p(\mathbf{o}|\mathbf{w})$,

$$p(\mathbf{o}|\mathbf{w}) = \sum_{\mathbf{x}} p(\mathbf{o}, \mathbf{x}|\mathbf{w})$$

However, it is common practice to make the assumption,

$$p(\mathbf{o}|\mathbf{w}) \approx \max_{\mathbf{x}} p(\mathbf{o}, \mathbf{x}|\mathbf{w})$$

This is often referred to as the *Viterbi assumption.* With this approximation, the objective becomes finding the optimal state sequence instead of the optimal word sequence. These two objectives are different because a given word sequence will have many possible state sequences. Therefore, the fundamental problem in performing HMM-based recognition is to estimate a state sequence for a given input. The basic steps for doing so are described below.

2.2 Acoustic modeling

The acoustic modeling problem in ASR can roughly be separated into the following three components. The rest of this section will review these three topics by describing a common method associated with each,

1. **Feature extraction** with Mel-frequency cepstral coefficients (MFCC)
2. **Acoustic scoring** with Gaussian mixture models (GMMs)
3. **Sequence modeling** with hidden Markov models (HMMs)

Figure 2-1 shows an overview of how these three steps can be used to hypothesize a string of phonetic units given an input speech signal.

2.2.1 Signal representation: MFCC

The input to an ASR system is a raw audio signal – a stream of waveform samples often at (or resampled to) roughly 16,000 samples per second. The initial stage of *feature extraction* consists of basic signal processing to put the data into a more manageable form for the statistical modeling to follow. A good representation should reduce the dimensionality, preserve relevant information, and put it in a form well suited to the statistical modeling techniques (e.g., decorrelating the features). Mel-frequency cepstral coefficients (MFCC) [24] have become a standard method for feature extraction in ASR.

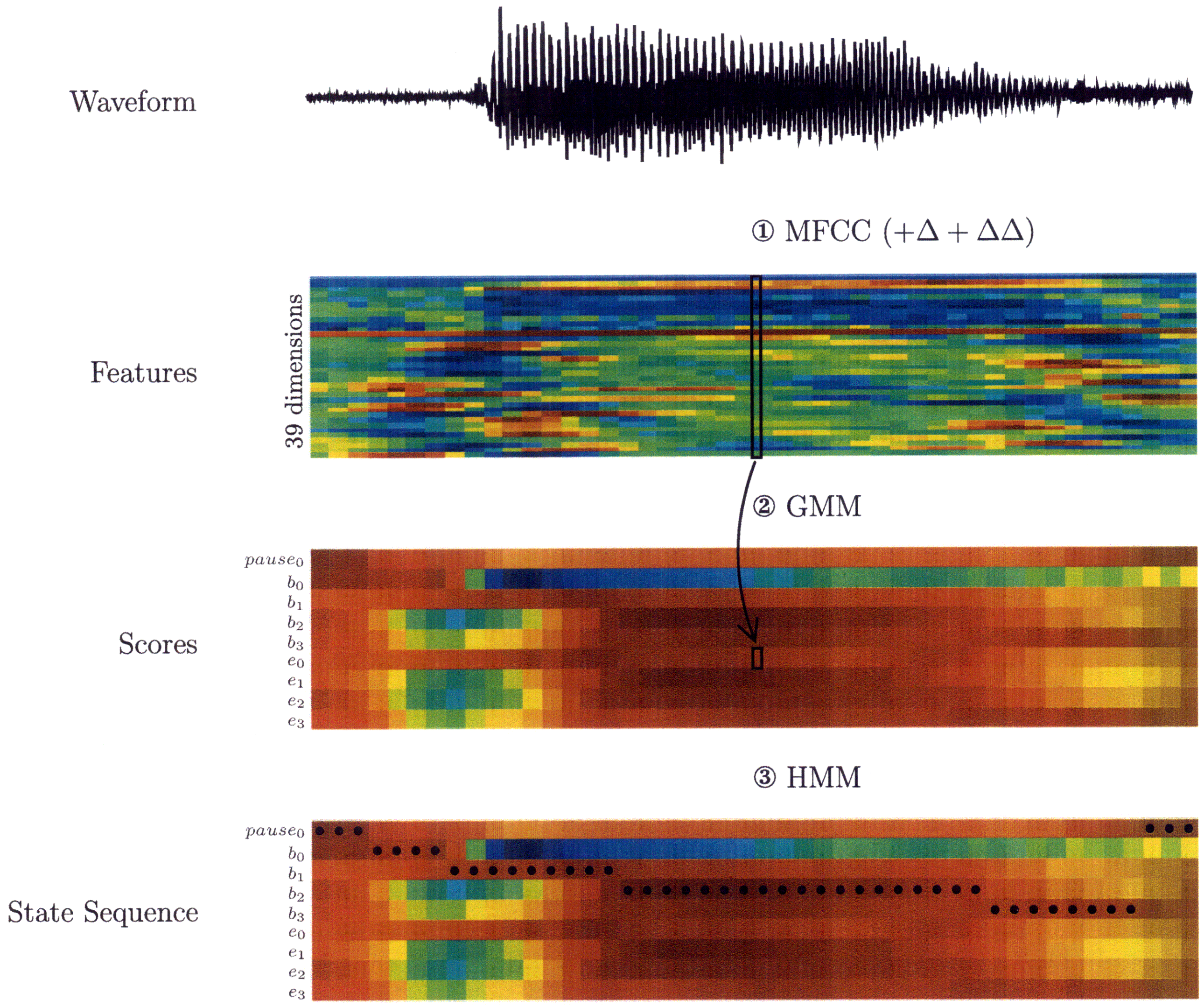


Figure 2-1: Overview of the typical stages involved for a simple ASR task. Shown is a HMM setup for isolated word recognition between two words: b and e, each with four states (w_i denotes i^{th} state of word w). The feature values are shown normalized in each dimension. The image representing the scores shows the log-likelihood for each state at each point in time (red is a high log-likelihood, and blue is low). Each of the three major steps is described in Section 2.2.

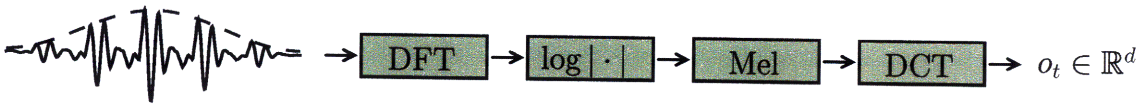


Figure 2-2: The steps involved in computing MFCCs. A windowed speech segment is converted to a low-dimensional (e.g., $d \approx 13$) representation of the short-time spectral shape.

The steps involved in computing MFCCs (shown in Figure 2-2) are,

1. **STFT**. A short-time Fourier transform (STFT) is computed from a (often pre-emphasized) waveform. For one typical choice of parameters, the DFT (discrete Fourier transform) is computed over 25ms hamming windows placed every 10ms. Each window will be referred to as a *frame*. In each frame, only the DFT magnitude is retained, and it is converted to decibels ($10\log_{10}[\cdot]$).

2. **Mel Filterbank**. Over each frame's spectrum, a bank of triangular filters spaced according to the Mel-frequency scale is applied [98]. This reduces the frequency resolution and warps frequencies to be spaced logarithmically (according to the Mel scale). The log-energy under each filter is retained, and the resulting vector (roughly 40 dimensions) is referred to as a set of Mel-frequency spectral coefficients (MFSCs). (The time series of MFSCs will be referred to as a *Mel-spectrogram*.)

3. **DCT**. In each frame, a discrete cosine transform (DCT) of the MFSCs is taken, and only the first N coefficients are kept (typical values of N are in the range 12-14). This helps to decorrelate the features, reduce dimensionality, and effectively keep only the smooth information from the spectral profile.

4. **Delta Features**. The computation of MFCCs is complete after the DCT, but typically the final feature vector also includes some information from a larger time span than single frame. This is often done with *delta features*: time derivatives which measure the change in feature values. One typical way [98] to define delta features for a feature time series, $x[n]$, is,

$$d_N\{x\}[n] = \frac{\sum_{\tau=1}^{N} \tau(x[n+\tau] - x[n-\tau])}{2\sum_{\tau=1}^{N} \tau^2} \tag{2.4}$$

Where N controls the amount of neighboring context. A common feature choice is that of MFCC+Δ+$\Delta\Delta$: MFCCs with first and second derivatives stacked at each frame. For example, 13 dimensional MFCCs with derivatives computed over ± 1 frame ($\tau = 1$) will result in a final 39 dimensional representation of (ignoring constant scaling factors):

$$\begin{aligned}[\mathbf{x},\ \mathbf{d}_1\{\mathbf{x}\},\ \mathbf{d}_1\{\mathbf{d}_1\{\mathbf{x}\}\}] \ = \ & [\mathbf{x}[n], \\ & \mathbf{x}[n+1]-\mathbf{x}[n-1], \\ & 2\mathbf{x}[n+2]+\mathbf{x}[n+1]-\mathbf{x}[n-1]-2\mathbf{x}[n-2]]\end{aligned}$$

A popular alternative to MFCCs are so-called perceptual linear prediction (PLP) coefficients [38]. The PLP technique attempts to model several aspects of human perception to create an auditory spectral representation which is then modeled with an all-pole filter. Both MFCCs and PLPs are based on a short-time Fourier Transform (STFT) and can be considered alternative ways to derive a low-dimensional representation of the short-time spectral shape. Therefore, the problems with MFCCs discussed in Chapter 3 are also relevant to PLP coefficients.

2.2.2 Acoustic scoring: GMM

For every state in the acoustic model, there is a Gaussian mixture model (GMM) which models the likelihood of the extracted features (e.g., the 39-dimensional MFCCs + Δ + $\Delta\Delta$) being generated by a given state. The GMM likelihood for a mixture with K components and observations, o, is given by,

$$p(o) = \sum_{k=1}^{K} w_k p_k(o) \tag{2.5}$$

$$= \sum_{k=1}^{K} w_k \mathcal{N}(o; \mu_k, \Sigma_k) \tag{2.6}$$

where $p_k(o) \sim \mathcal{N}(o; \mu_k, \Sigma_k)$ represents a Gaussian distribution with mean μ_k and covariance matrix Σ_k. The weights, w_k, scale each Gaussian component and sum to one, $\sum w_i = 1$. Often, only diagonal-covariance Gaussians are used, $\Sigma_k = \text{diag}(\sigma_{k,1}^2, \ldots, \sigma_{k,N}^2)$, in which case the log-likelihood for a single Gaussian is given by,

$$\log p_k(o) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{d}\log\sigma_{k,i}^2 - \frac{1}{2}\sum_{i=1}^{d}(o_i - \mu_{k,i})^2/\sigma_{k,i}^2$$

where d is the dimensionality of o. The GMM provides a "score", typically expressed as the log-likelihood, $\log p(o)$, for how well the features o were generated by a particular state. This is illustrated in step (b) of Figure 2-1. Other methods can be used for acoustic scoring, such as multi-layer perceptrons [9].

2.2.3 Sequence modeling: HMM

The GMM outputs a score (log-likelihood) for each state in each frame. Incorporating constraints from frame-to-frame is done with a *hidden Markov model* (HMM). An HMM is a probabilistic model consisting of a sequence of hidden state variables, $\mathbf{x} = x_1, \ldots, x_T$, each with a corresponding observed value, $\mathbf{o} = o_1, \ldots, o_T$. For an N state HMM, each of the state variables takes on one of a discrete set of N possible states, $x_i \in \{s_1, \ldots, s_N\}$.

The *Markov property* asserts that the evolution of the states operates such that the identity of the current state only depends on the identity of the previous state – as opposed to having some dependence on any of its previous history. Once the previous state is known, knowledge of any other state offers no additional information. This is stated mathematically as a conditional independence assumption,

$$P(x_t | x_1 x_2 \ldots x_{t-1}) = P(x_t | x_{t-1})$$

The term *hidden* refers to the fact that the state sequence is not directly observable. Evidence of the state identities comes indirectly through a corresponding sequence of observations, $\mathbf{o} = o_1, \ldots, o_T$.

Joint distribution

The two sets of variables ($\mathbf{x}$ and $\mathbf{o}$) can be viewed as the directed graphical model shown in Figure 2-3 (see Appendix B for background on graphical models). The structure of this graph allows the joint distribution of $\mathbf{x}$ and $\mathbf{o}$, p($\mathbf{x}$, $\mathbf{o}$), to be factored in the following form,

$$p(\mathbf{x}, \mathbf{o}) = P(x_1) p(o_1 | x_1) \prod_{t=2}^{T} p(o_t | x_t) P(x_t | x_{t-1}) \tag{2.7}$$

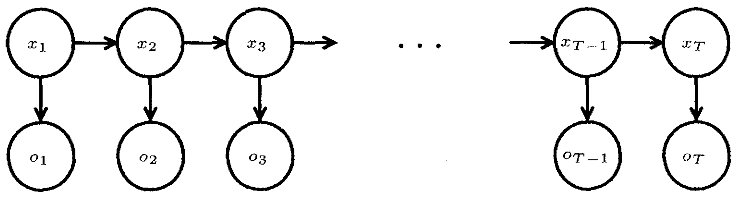


Figure 2-3: An HMM viewed as a directed graphical model with variables representing states, $\mathbf{x} = x_1, \ldots, x_T$, and observations, $\mathbf{o} = o_1, \ldots, o_T$.

The parameters of this distribution define the model and consist of,

- Initial state probabilities.

  $P(x_k)$ is the probability that the first state is state k.

- Transition probabilities.

  $P(x_i|x_j)$ is the probability for transitioning from state x_j to state x_i.

- Emission distribution parameters.

  The *emission densities*, $p(o|x)$, are often parametric distributions. For the common case of GMM emissions, this set of parameters consists of mean vectors and covariance matrices for each mixture component for each state, as well as the mixture weights.

Generative model

An HMM is a *generative* model, defining a probabilistic method for generating data. Data is generated from an HMM by the following procedure,

1. Draw an initial state, x_1, from the distribution, $P(x_k)$.
2. Draw an observation, o_1 from the distribution for state x_1, $p(o|x_1)$.
3. For $t = 2, \ldots, N$,
   (a) Draw state x_t, from the distribution, $P(x_k|x_{t-1})$
   (b) Draw an observation, o_t from the distribution for state x_t, $p(o|x_t)$.

Performing these steps produces values for $\mathbf{x} = x_1, \ldots, x_T$, and $\mathbf{o} = o_1, \ldots, o_T$, i.e. samples from the joint distribution, $p(\mathbf{x}, \mathbf{o})$.

In ASR applications, HMM models are often restricted to be *left-to-right*, meaning that $P(x_i|x_j)$ is non-zero only for $j > i$. A *strictly left-to-right* model has $P(x_i|x_j)$ non-zero only for $j \in \{i, i+1\}$, and initial state distribution $P(x_i) = 0$ for $i \neq 1$. In this case, the hidden state sequence always progresses through the states of the model in order from 1 to N, visiting each state for one or more frames. (Also assuming it is required to end on state N).

2.3 Algorithms

The success of HMM-based ASR systems can be attributed to powerful and efficient data-driven algorithms for training (setting parameters of the model from examples) and testing (decoding a test utterance to hypothesize a word string). Here, we will only briefly mention the details of the common algorithms which are related to methods used later in this thesis. Complete descriptions can be found in [6, 74].

2.3.1 Training

The parameters of the HMMs (means, variances of the Gaussians, GMM weights, and state transition probabilities) are learned in a data-driven manner. The traditional Baum-Welch algorithm [74], is an algorithm for maximum-likelihood training of HMMs parameters (using a variation of the expectation-maximization, or EM, algorithm [25]). Each iteration increases the likelihood of the training data, and thus converges to a local maximum. More recently, significant gains have been made with discriminative training methods [65, 71], which optimizes parameters for their discriminative power rather than to maximize likelihood. The baseline recognizer used in this thesis is a typical application of maximum likelihood training, and is described in Appendix A.

2.3.2 Decoding

The decoding problem is that of hypothesizing the underlying hidden state sequence, given the sequence of observations. This problem can be solved efficiently for HMMs with a dynamic programming algorithm known as the *Viterbi algorithm* [74]. The algorithms used later in this thesis (particular the *max-sum* algorithm for acyclic graphical models discussed in Section 4.7.3) are closely related to the Viterbi algorithm for HMMs. Psuedocode for Viterbi decoding is shown in Algorithm 2.1.

```
Input:
    Observation sequence, $o_1, \dots, o_T$
    Acoustic score: $f_i(o) = \log p(o|x_i)$ for each state, $i = 1, \dots, N$.
    Starting state: $t_i = \log P(x_1 = s_i)$.
    Transitions: $t_{i,j} = \log P(x_t = s_j | x_{t-1} = s_i)$.
Output:
    Optimal state sequence, $\mathbf{x}^* = x_1^*, \dots, x_T^*$.
    Total score, $f(\mathbf{x}^*)$.
begin
    $\delta_1(i) = t_i + f_i(o_1)$

    for $t = 2, \dots, T$ do                                   /* forward pass */
        $\delta_t(i), \psi_t(i) = \text{argmax}_j [\delta_{t-1}(j) + t_{i,j} + f_i(o_t)]$
    $f(\mathbf{x}^*), x_T^* = \text{argmax}_j \, \delta_T(i)$

    for $t = T-1, T-2, \dots, 1$ do                           /* backtrace */
        $x_t^* = \psi_{t+1}(x_{t+1}^*)$
end
```

Algorithm 2.1: Viterbi algorithm for finding optimal state sequence in an HMM. The notation, $a, b = \text{argmax}_x f(x)$, is a combination of the two statements, $b = \text{argmax}_x f(x)$ and $a = f(b)$.

Chapter 3

Local features and the desired acoustic modeling approach

The standard method for acoustic modeling as described in the previous chapter has been the basis for many successful ASR systems. However, there are fundamental problems with these techniques which have been an important factor in limiting the accuracy and robustness of ASR. So, while performance has continually improved, ASR still lags behind human performance, particularly in adverse conditions [60, 81].

This chapter discusses a range of specific problems with the standard MFCC/HMM framework. While many are well known and have been discussed previously, we believe some have not been given appropriate attention, and focusing on these problems motivates the direction of this thesis. In particular, a range of the problems stem from the *non-localized* nature of the typical MFCC features. This chapter motivates the desire for localized features and explores their discriminative ability.

In the course of describing problems with typical features and acoustic models, we are lead to a set of desirable principles to guide new approaches to acoustic modeling. After stating these goals, the chapter concludes with a discussion of previous work with similar motivations.

3.1 A motivating example

A number of the problems we wish to address can be highlighted by looking at a simple phonetic discrimination task and a specific example which causes problems for a typical MFCC/HMM-based recognizer. The ISOLET corpus [15] contains isolated spoken letters of the English alphabet. A detailed description of the corpus and the baseline HMM-based recognizer are given in Appendix A. The baseline recognizer gives error rates of 3.79% (296/7800) on the clean test set and 42.50% (3315/7800) on the noisy test set for full 26-way classification. There is clearly room for improvement in noise, but also in the clean condition. We conducted an informal listening test, in which a human listener made only 13 errors when attempting to classify the 296 utterances which the baseline recognizer misclassified.

Not surprisingly, the HMM errors are not uniformly distributed among letter confusions (see Section A.2.2). The most common confusions are between the letters *B* and *E*. When restricted to binary classification on 600 *B* and *E* utterances, the baseline makes 54 errors (9.0%). Informal testing on these 600 tokens showed two human listeners make 3 and 9 errors respectively (1.0% average). A human spectrogram reader (one person using only the visual inspection of spectrograms) made 20 errors (3.3%). These numbers suggest not only that there a large improvement to make in clean speech, but also there is "visual" and interpretable information in the spectrogram which is not being exploited by the baseline recognizer.

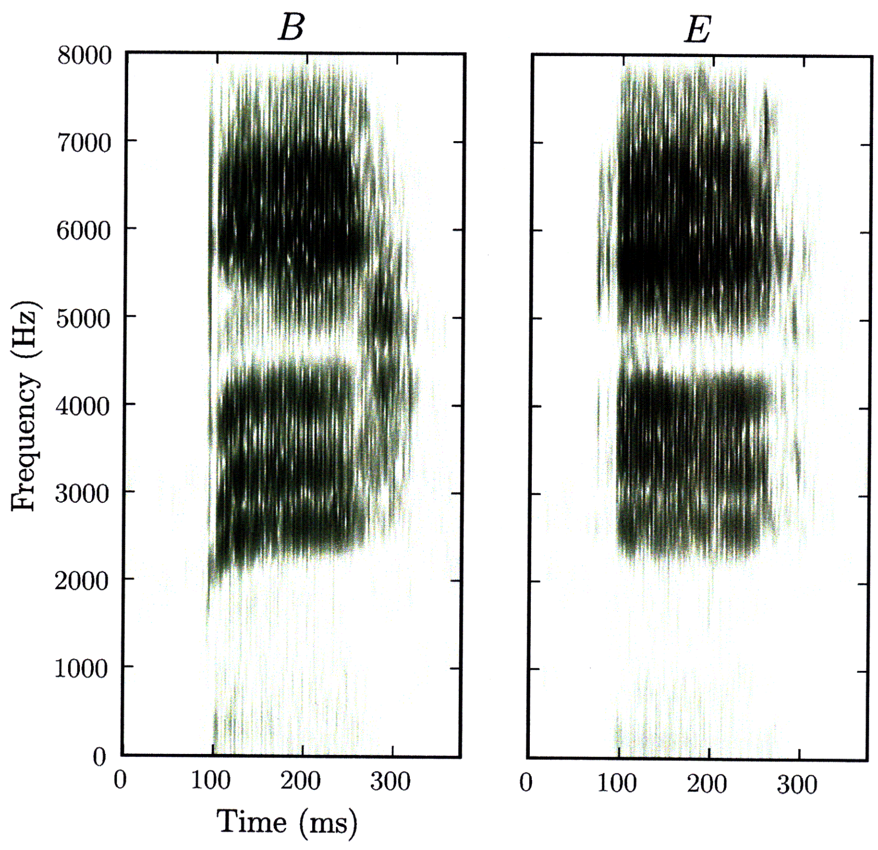


Figure 3-1: On the left: A spectrogram of an example *B* which was misclassified as an *E* by a baseline HMM system. For comparison, shown on the right is a spectrogram of an *E* example from the same speaker.

A spectrogram from one of the *B*-*E* misclassifications is shown in in Figure 3-1. On the left is a spectrogram of the spoken letter *B* (CV syllable /b iy/) which was misclassified by the baseline system as *E* (/iy/). For visual comparison, an example *E* from the same speaker is shown on the right. Seeing this example *B* spectrogram, a phonetician or an experienced spectrogram reader would consider *B* more likely than *E*, primarily due to the slight upward slope in the dark band of energy between 2000 and 2500 Hz. In the *E*, the corresponding edge has no noticeable slope.

These dark bands of energy highlight the *formants*, which are the resonant frequencies of the vocal tract. Formants are denoted F1, F2, F3, etc., and in both examples of Figure 3-1, the formants F2, F3, and F4 are clearly visible between 2000 and 4500 Hz (in these examples, F1 is around 400 Hz and is very weak). Due to basic theories of acoustics, the closing of the lips tends to lower all formant frequencies [87]. The letter *B* is pronounced with the lips beginning in a closed position, and thus formants will tend to rise as the lips open to pronounce the following vowel. Therefore, in this example, the visible upward sloping of the formants at the onset (particularly noticeable in F2) is a strong cue for choosing *B* rather than *E*.[1] Notice that this F2 transition cue is only present in a narrow frequency range, and is not represented by a target spectral shape. Rather, the cue comes from the nature in which the spectral shape is changing. This is a rather subtle distinction, but these kinds of cues must be correctly identified if ASR systems are to reach human-level phonetic discrimination abilities.

3.2 Problems with standard ASR

This example and others like it (seen when viewing spectrograms from ASR errors) help to highlight problems with traditional feature extraction and acoustic modeling and point to ways for possible improvement. In this section, we present a list of (related) problems we hope to address in this thesis.

3.2.1 Non-local features

The typical MFCC-based features are *non-local* in that each feature value is affected by energy at all frequencies (as opposed to a small *local* frequency range). As seen with the *B-E* example of Figure 3-1, many important phonetic phenomena appear as T-F (time-frequency) patterns in only a narrow range of frequencies. This problem of non-local features is central to the ideas of this thesis, and discussed in more detail in Section 3.3.

3.2.2 Fixed spectral templates

In addition to spanning the entire frequency axis, common "spectral-slice" features such as MFCCs rely on the flawed assumption that phonetic units (e.g., phones or phone states) are characterized by a *fixed* spectral template. Note that this issue is distinct from the problem of being non-localized: even if concentrating on detecting patterns within a limited frequency band, their shape and their frequency location will not be fixed.

For a simple example related to speaker variation, Figure 3-2 shows two instances of the letter *C* (i.e., diphone /s iy/); a female on the left and a male on the right. The highlighted region is a point in time which is likely to be associated with the

[1]Another subtle cue is the nature of the narrow onset burst. In the *B*, it is spread across all frequencies. In the *E*, the onset energy is shaped more by the formant profile, with a hole around 4700 Hz. At the onset of *E*, the vocal tract is already in the position for the vowel /iy/ and thus the onset energy is filtered accordingly.

same state in a typical model (the middle of the /iy/). In these two examples the spectral profile at that point in time are considerably different. The male, having a longer vocal tract, has lower formant frequencies (vocal tract resonances) than the female [87].

While such speaker differences are well known and understood, the MFCC/GMM front-end methods ignore them, and often rely on capturing these large differences with different mixture components.[2] Also, adaptation techniques such as MLLR [53] and VTLN [52, 95] are common, however the model itself doesn't capture the commonalities that make it a single phonetic unit in the first place. In GMMs, the model instead represents a collection "prototypes" rather than encoding the essence of the spectral pattern. These problems are only alleviated with large training sets and an increase in the number of Gaussians in the GMMs. The alternative to this would be to find a model based on *flexible* rather than *fixed* pattern detectors. For the /iy/ case shown here, instead of learning the shapes of fixed spectral profiles, we would like to have a model which identifies this phoneme with a large gap between F1 and the cluster of F2 and F3 (and F4). Ideally a statistical model would encode the nature of how these shapes vary between speakers.

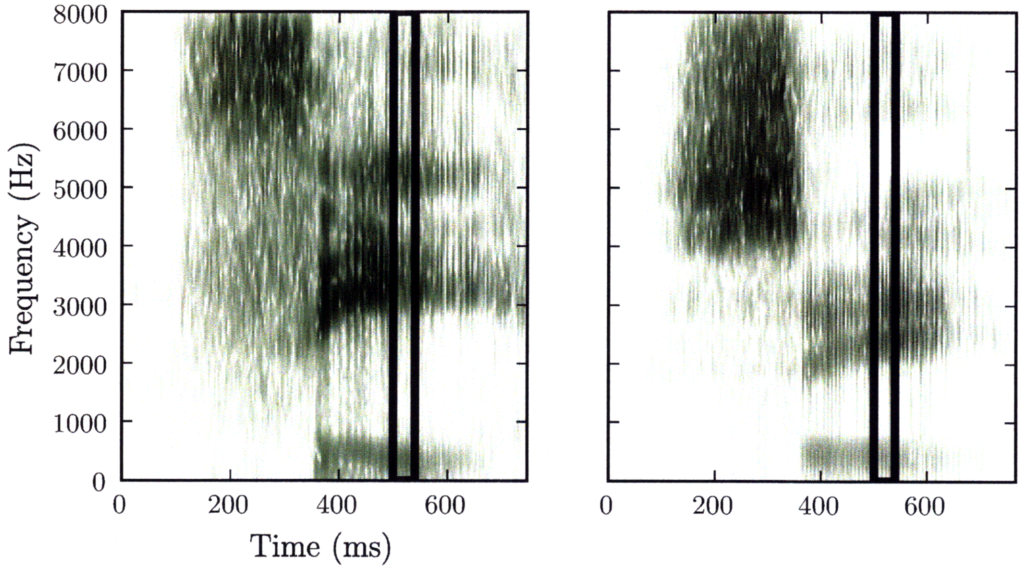


Figure 3-2: Spectrograms of the spoken letter *C* (diphone /s iy/) from two speakers, (a) female, (b) male. The highlighted regions will likely be represented by the same state in our model, but their spectral profiles are quite different (in the sense of mean-squared error).

[2]For example, if we use GMMs with two mixture components, typical training may result in the mixture membership largely based on male in one and female in the other.

3.2.3 Encoding of spectral dynamics

It has been long known that *formant transitions* are a crucial element of speech perception [57]. Formant dynamics do not simply represent a smooth transition between a sequence of spectral targets – rather the *transitions themselves* can be the primary carriers of phonetic identity [58].

Typical acoustic modeling can capture such information, but rather indirectly with delta features and by stepping through discrete HMM states. The underlying assumptions of MFCC/HMM recognizers: the "beads-on-a-string" and "spectral-slice" models of speech [68, 67] are not well suited for capturing dynamics such as formant transitions. The example from Figure 3-1 earlier in this chapter motivates a different approach: creating a model that encodes these spectral dynamics explicitly.

3.2.4 Phonetic interpretability

The information contained in typical acoustic models is hard to interpret phonetically. Ultimately, this may not be important; from an engineering standpoint, performance may be the only objective. However, until ASR systems work perfectly, designers must continually find, analyze, and fix problems with the system.

For example, in the case of the baseline *B*-*E* confusion discussed earlier (Figure 3-1), we would like to know "why" the recognizer got this example wrong. Did it fail to "see" the clearly upward sloping F2 at the start of the vowel? Was there something else about that example which it thought was characteristic of *E*? Because this kind of phonetic information is not encoded explicitly, such introspection is not possible. The only information available is to look at the GMM log-likelihoods for each frame, which is not very informative.

However, one can attempt to understand and visualize the information contained in an HMM acoustic model. For example, we can look at the spectral shape stored in each Gaussian mean vector for each state. Several such examples from ISOLET are shown in Figure 3-3. This figure shows the spectral profile for each Gaussian as represented by 20 MFSCs (inverted from the truncated DCT of the number of cepstral coefficients).[3] While some phonetic features are noticeable, it is qualitatively missing the kind of phonetically interpretable descriptions of these letters, such as the downward slope characterizing the diphthong in the letter *Q*, the rising formants at the onset of *B*, the segmental structure of *H*, etc. Also, note that the more phonetically complex letter *W* would likely need more than four states for a sufficient model.

3.2.5 Quality of generative model

A related problem to phonetic interpretability is that of the generative ability of the HMM model. As described in Section 2.2.3, the HMM is a *generative* model. The model represents an assumption of how a sequence of observations was generated

[3]Note that this visualization does not show all the information in the model; namely, the Gaussian variances, the delta features, the GMM mixture weights, and the state transition probabilities.

probabilistically. Decoding (performing ASR) solves the inverse problem of inferring the underlying states from a sequence of observed variables.

While it is not common to use ASR HMMs to generate speech or spectrograms, we believe this is an instructive way to visualize what information is encoded in the model, and that a good generative model should generate meaningful synthetic outputs. Figure 3-4 shows several examples of the letter *X* generated from its ISOLET model. The state sequence was generated by drawing from the distribution defined by the HMM model (steps listed in Section 2.2.3). In each frame, the state's GMM mixture component is drawn from the GMM weights, and the corresponding Gaussian component mean is visualized (as was done in viewing the model information in Figure 3-3).[4]

One problem evident from these figures is that given the current state, the mixture index is chosen independently of its neighbors. This results in the abrupt discontinuities in spectral shape from frame-to-frame.

3.2.6 Duration modeling

The Markov property of HMMs implicitly places a geometric distribution (the discrete version of an exponential distribution) on the duration of each state [72]. The poor modeling of duration is one of the reasons for the poor generative ability of HMMs. For a strictly left-to-right HMM, there is only a single parameter for transitions from each state. Let p_k be the probability of exiting state k, and $1 - p_k$ be the probability of staying in state k. The probability that the duration of state k, denoted D_k, has a length of d frames is given by,

$$\Pr(D_k = d) = (1 - p_k)^{d-1} p_k \qquad \text{for } d = 1, 2, 3, \ldots \tag{3.1}$$

An example is shown in Figure 3-5. The empirical distributions shown are based on a forced-path alignment, and therefore are similar to what the transition probabilities are trained on. Clearly, the geometric distribution is not appropriate. The underlying problem is that of the *Markov assumption*: how long it has been in a state *does* influence the probability of leaving that state.

[4]As mentioned before, this visualization is not a complete representation of information in the model. In particular to this example, the delta features, if incorporated, could help smooth the differences between frames.

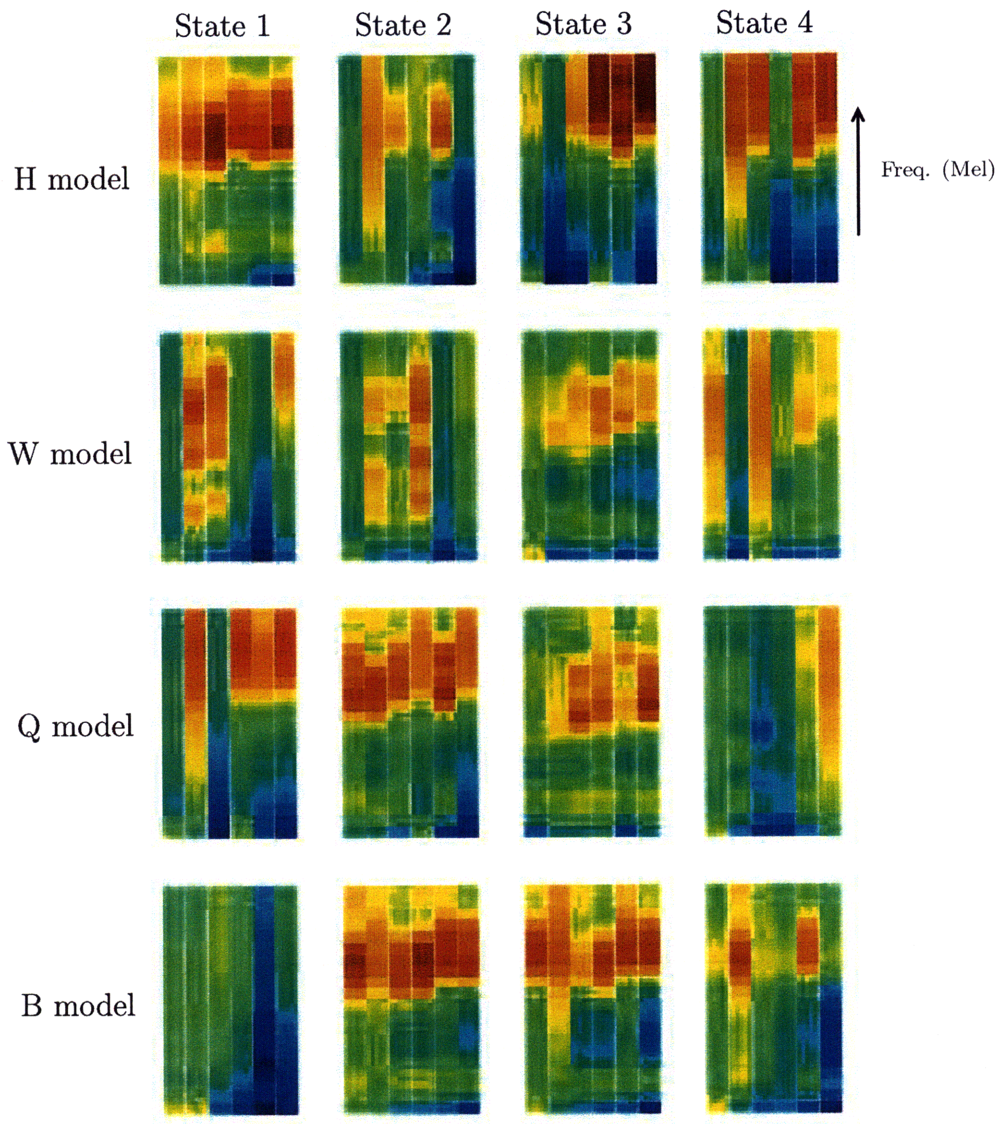


Figure 3-3: Visualization of information in a trained isolated-word HMM recognizer, showing the models for four ISOLET letters: *H*, *W*, *Q*, *B*. Each model has four states, and each state is modeled with a GMM with six Gaussians. For each state in each model, there are six columns showing the spectral profile (Mel-scale) of the Gaussian means.

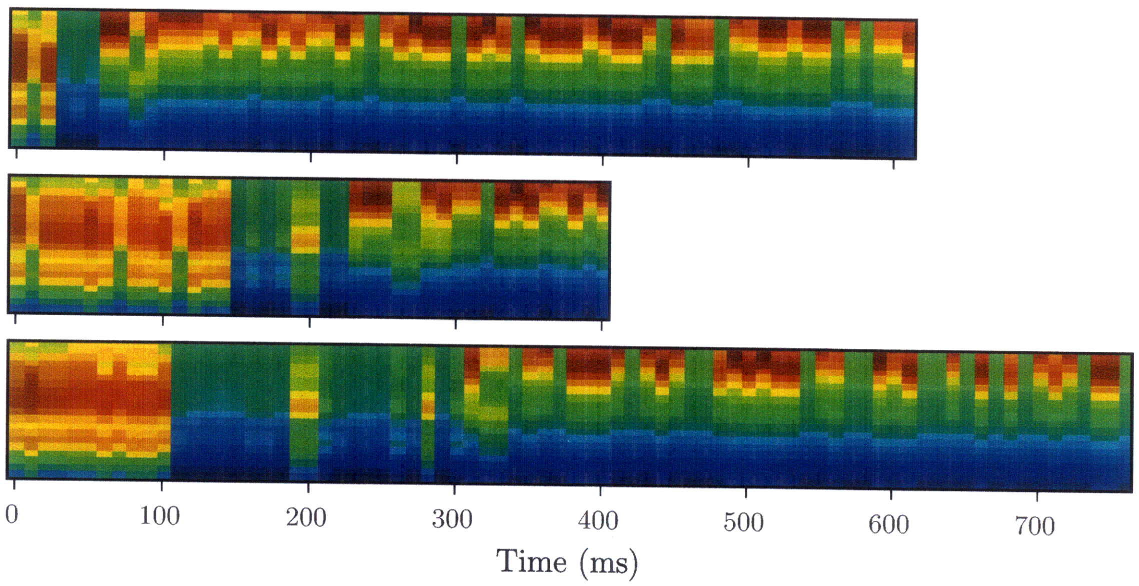


Figure 3-4: Visualization of the generative ability of models in an MFCC/GMM/HMM isolated word recognizer. Each of the three rows shows an example of sampling from the distribution of the model for the ISOLET letter *X*. Shown is a reconstruction of the Mel-spectrogram.

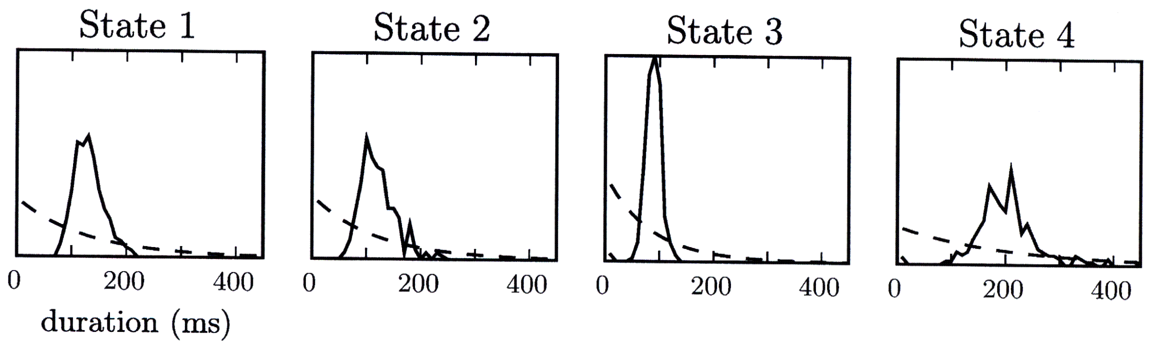


Figure 3-5: Comparison between the duration model imposed by an HMM (dashed) and the empirical distribution of the duration (solid). The four states of the model for ISOLET letter *X* are shown. Other letters have qualitatively similar results: a peaked empirical distribution (roughly shaped like a Gaussian or Gamma distribution), while being modeled by an exponential distribution.

3.3 Local features

MFCCs provide a low-dimensional, compact representation of a signal's short-time, smoothed spectral shape (Section 2.2.1). MFCCs have been empirically successful and have been found to work well with the successive stages in the ASR system (e.g. as input to diagonal-covariance GMMs). However, MFCC features are not localized in frequency. Each feature coefficient includes information from all frequencies due to the DCT bases spreading across the entire frequency axis. There are several closely related reasons why this may be considered problematic:

- **Noise Robustness**

  If there is a noise source corrupting one frequency band, all MFCCs are affected. Recent work on missing feature methods [18, 3] has shown great improvements in noise robust ASR by ignoring (or rather marginalizing out) T-F regions which can be identified as corrupted. These methods will not work on non-localized features.

- **Speech Perception**

  There are also many reasons to believe that human speech perception relies heavily on processing information in limited frequency bands [1]. The "glimpsing" model of speech perception suggests that humans can robustly decode noise-corrupted speech by taking advantage of local T-F regions having high SNR, and is supported by empirical evidence and computational models [17].

- **Auditory Neuroscience**

  Recent work in auditory neuroscience also provides evidence. Tonotopic maps appear at all known levels of auditory processing [97], meaning that most neurons have a strongest response at a particular frequency. (an "MFCC neuron" would not have a best frequency, but rather have peaks spread across all frequencies, corresponding to the cosine peaks of the DCT). Also, recent research seeking to characterize the behavior of individual neurons in the mammalian auditory cortex has resulted in models in which cortical neurons act as localized spectro-temporal pattern detectors [94, 14]. (represented by their so-called *spectro-temporal receptive field*, or STRF).

Because of these factors, we seek to explore features occupying a limited region of T-F instead of features spanning the entire frequency range. Such features will be extracted from a rectangular region over a T-F image, and thus will often be referred to as "patches".

3.3.1 MFCCs as "patches"

To explore patch-based features as alternatives to MFCCs, it is instructive to view MFCCs themselves in terms of patches over a spectrogram. Typical MFCC feature extraction performs two operations after computing the MFSCs (the Mel-spectrogram):

a DCT and delta-feature extraction (see Section 2.2.1). Because these are both linear operations, they can be combined and viewed as one linear operation applied to the Mel-spectrogram. Figure 3-6 shows a visualization of some of the resulting bases. Each can be thought of as a T-F "patch" which produces features when convolved over the Mel-spectrogram. Since the patch spans the entire frequency range, they slide over the spectrogram only horizontally and thus each produces one feature value per frame. Therefore, the baseline feature set can be thought of as being generated from 39 patches which look like those in Figure 3-6. With the knowledge of the types of phonetic information needed to be detected, it is unlikely these patches will ultimately be the optimal set.

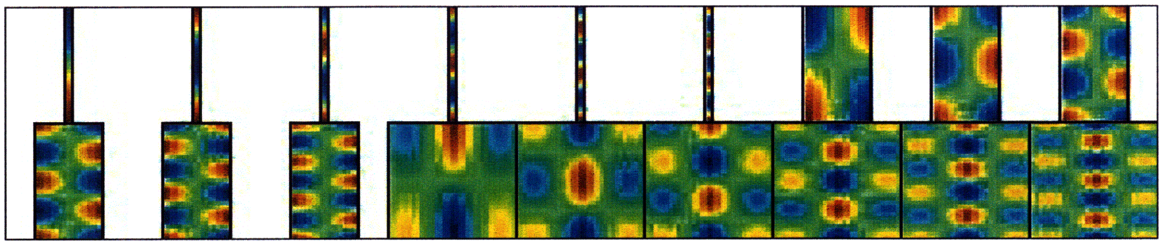


Figure 3-6: MFCC+Δ+$\Delta\Delta$ features when viewed as patches over a Mel-spectrogram. Here, an example 18 out of a typical 39 are shown arranged in two rows. Those shown correspond to the first 5 DCT coefficients for each of the MFCC, Δ, and $\Delta\Delta$ features. The horizontal (time) dimension reflects the delta computations, while the vertical (frequency) dimension represents the cosine transform.

3.3.2 Discriminative ability

Frequency-localized features may be desirable for noise robustness and phonetic interpretability, but will only be useful if the local information has the power for phonetic discrimination. This question was explored with some simple masking and classification experiments on sets of difficult letter pairs from the ISOLET database.

For a given letter pair, the following steps were performed to test localized classification:

1. All tokens are aligned to a common example using dynamic time warping (DTW) on spectrograms (the wideband spectrograms described in Section A.3). This warping results in a fixed-length feature vector for each utterance. (The feature vector consists of all T-F points in the aligned spectrogram).

2. The "most important" T-F regions are found by taking the points with the maximal difference in average spectra between the two classes. A set of masks can be produced by varying the threshold of the percentage of points to discard.

3. Classification is performed by simply collecting the set of points within the mask as input to a binary classifier.[5]

[5]Classification is done with a linear regularized least-squares (RLS) classifier. Other generic

Resulting masks and classification error rates are shown for the *B* vs. *E* case in Figure 3-7. Several other difficult letter pairs at various threshold levels are shown in Figure 3-8. From these results, several observations can be made,

- Classification based on a limited region of T-F is not only sufficient for good performance in most cases, but beneficial. Ignoring information irrelevant to discrimination helps the classifier concentrate on the important data (even in this case of uncorrupted, clean data). While the absolute performance will differ by the classification scheme used, success with any classifier gives evidence that the relevant information is contained in the local T-F regions used.

- For many of the examples, the resulting binary mask highlights regions which are interpretable as detecting phonetic features useful for classifying the given letter pair. For example, consider the 5% mask for *B*-*E* discrimination in Figure 3-7. It favors T-F regions around (a) low-frequency pre-voicing, (b) the burst onset, and (c) the F2 transition near the onset.

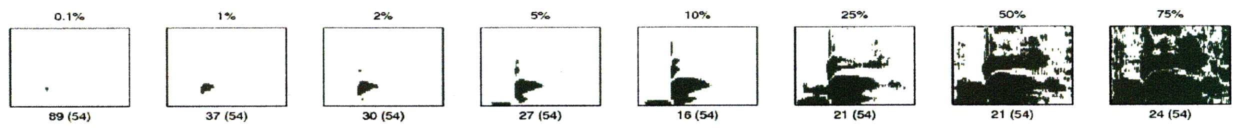


Figure 3-7: Masks for localized *B*-*E* discrimination, determined by keeping the top N% differing T-F points. Below each image is the number of classification errors (out of 600) and the number, 54, of errors made with the baseline HMM recognizer.

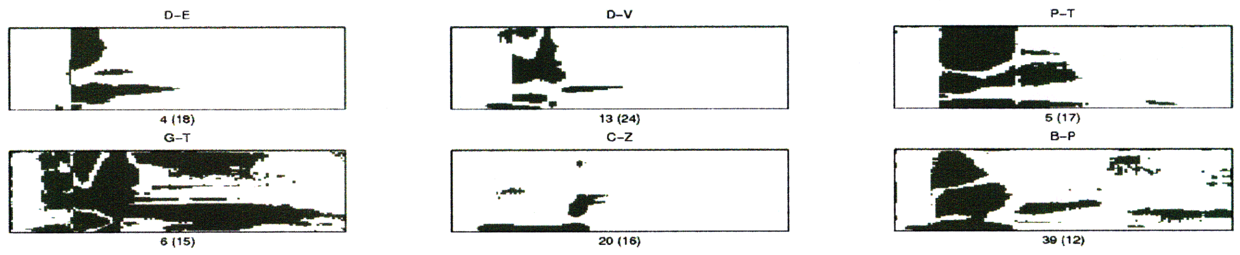


Figure 3-8: Examples of other difficult letter pairs at their optimal level of masking. Below each image is the number of classification errors (out of 600) with the number of errors made with the baseline recognizer in parentheses.

3.3.3 Phonetically inspired patches

With the viewpoint of feature extraction methods as determining a set of suitable patches, there is clearly a wide range of possibilities. While many previous meth-

classifiers would also be appropriate (such as support vector machines [22]), but we use RLS classifiers several times in this thesis due to their simplicity and because they allow efficient automatic determination of the regularization parameter through leave-one-out errors [77].

ods are based on data-driven or signal processing techniques, one of our primary motivations is from knowledge of acoustic phonetics.

Consider the patch shown in Figure 3-9. Here, a spectrogram is filtered by a patch designed to detect a very specific phonetic event – a formant transition with a bandwidth of 300 Hz and a slope of 15 Hz/ms. Notice that the response is very strong at the occurrence of its target feature, as expected. It is likely that very specific features (extracted with very specific patches) will be more robust in detecting their target than using a collection of generic features.

It is also worth noting the biological case for using specific, local patches as features. As mentioned in Section 3.3 when motivating local features, recent work in auditory neuroscience has found neurons in the auditory cortex which act as localized time-frequency pattern detectors [26, 14]. Figure 3-10 shows the *spectro-temporal receptive field* (STRF) for both a measured neuron and a proposed model.

Contrast the patches inspired by phonetics (Figure 3-9) and biology (Figure 3-10) with the MFCC patches of Figure 3-6. The phonetically motivated patches are more biologically plausible and may offer benefits over MFCCs for both noise robustness and phonetic interpretability.

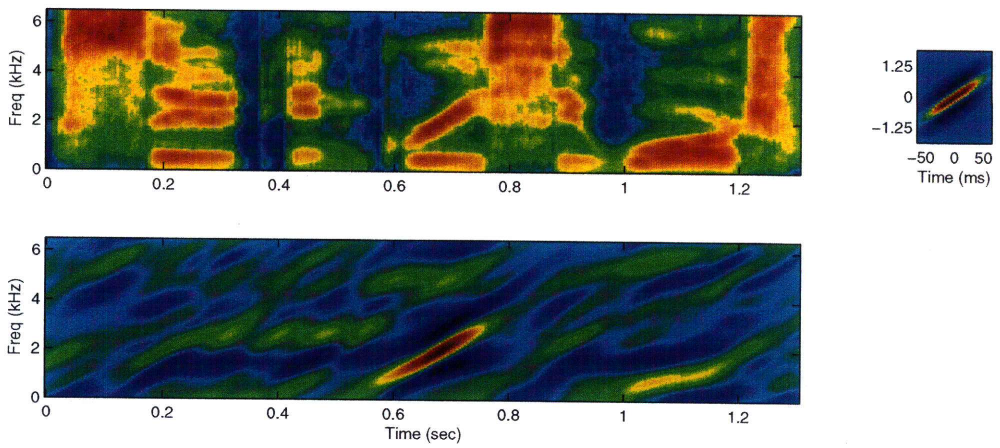


Figure 3-9: An example spectrogram and its response to a small time-frequency filter specifically designed to detect formant transitions at slope 15Hz/ms. Notice the strong response starting at t=0.6 seconds. (This happens to be the start of the /r iy/ in the word "greasy").

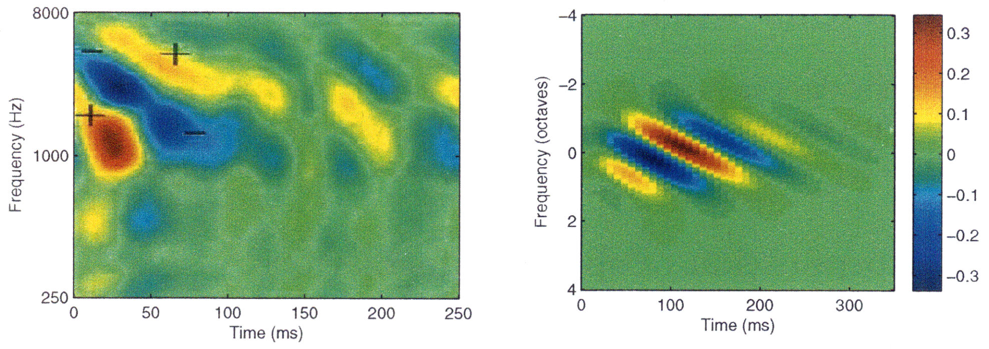


Figure 3-10: (a) A measured spectro-temporal receptive field (STRF) from a neuron in the auditory cortex of a ferret (reproduced from [46]; credited to David J. Klein [26]). (b) An example response from the cortical model proposed by Chi, Ru, and Shamma [14].

3.4 The desired approach to ASR

This chapter described a list of (related) problems with standard acoustic modeling methods for ASR. From this list, a set of properties emerge which we would like alternative acoustic modeling strategies to have,

- **Localized T-F pattern detectors**

  Instead of features derived from "spectral-slice" measurements, the acoustic model should be based on localized pattern detectors concentrating on specific spectro-temporal patterns.

- **Explicit modeling of phonetic cues**

  The localized detectors should be designed for specific, phonetically-interpretable cues. Heterogeneous measurements can be tailored to specific cues, and interpretability will lead to increased ability to tune system performance.

Therefore, this chapter has motivated the overall goal of this thesis (recall the visualization from the introduction, Figure 1-1 on page 17):

Model each phonetic unit as a deformable template of spectro-temporally localized cues rather than a sequence of fixed spectral shapes.

The success of HMM-based ASR has come from having principled statistical methods to learn from large training sets, and having efficient methods for decoding. For any new alternative approaches to compete, similar techniques are needed. In the following chapters, we outline a statistical model able to implement this desired approach.

3.5 Previous work

We will conclude this chapter with a review of some previous work in acoustic modeling having similar goals to those stated above. Relevant work has come from a variety of topics, each of which will be reviewed briefly in a subsection below.

3.5.1 Incorporation of speech information

As discussed in Section 3.1, one motivation for this thesis is the observation that human spectrogram readers can identify phonetic information which is not being fully utilized by standard speech recognizers. Zue and colleagues have previously advocated the use of speech knowledge in ASR [100], and were also motivated by the experience of human spectrogram reading [99, 49]. The methods to be proposed in the next chapter are quite similar to the the early work by Leung and Zue [54], which attempted to use the visual information in spectrograms for phonetic discrimination. Others have also attempted to use machine vision-related techniques on speech spectrograms [2, 76].

Work by Cole et al. on the FEATURE system [16] is very closely related to our desired approach. Their discussion of "templates vs. features" is similar to the comparison of "spectral slices vs. local pattern detectors" discussed earlier. The motivations and goals of this thesis are similar to those of [16], but we hope to explore more complete statistical methods able to implement these ideas.

More recent research aiming to incorporate speech knowledge in ASR includes a large amount of work to utilize articulatory information into recognition systems. This includes building front-end classifiers for articulatory classes such as manner and place [45, 83] as well as utilizing such information in alternative architectures to HMMs, particularly dynamic Bayesian networks (DBNs) [32, 61, 5]. In [62], multiple ways to combine front-end articulatory information into ASR systems are discussed. A recent review of work using speech production knowledge in ASR can be found in [44]. Note that much the work on articulatory features still uses conventional features, such as MFCCs or PLPs.

3.5.2 Local features

Many researchers have investigated the use of local features, often with the goal of improving noise robustness. Some work has involved doing standard feature extraction and recognition in subbands [7, 64], while others have proposed alternative feature extraction techniques using only limited bands of the spectrum. 2-D Gabor features (sinusoids tapered by a decaying exponential) have been extracted from spectrograms and used as features in HMM recognizers [47, 46]. Similar to Gabor analysis, 2-D localized cepstra (local 2-D DCT) were used for phonetic classification in [10]. TRaP features ("temporal patterns") [39, 67] have shown benefits by utilizing information in a single frequency channel over relatively long time scales.

Another system based on features derived from local T-F patches is the recent work by Ezzat and Poggio [28]. They present a model influenced by biologically-inspired hierarchical models of vision [85], which use local patches (*receptive fields*), and max-pooling layers to incorporate some amount of translational invariance in the location of the patches. In [28], a word-spotting system is described using models consisting of a set of patches randomly extracted from the Mel-spectra of training instances. During testing, a fixed amount of T-F flexibility is given to the patches to match on a potential target. This work has some similarities to the ideas and methods proposed in this thesis. A more detailed comparison is given in Section 6.1 after the details of our proposed methods are described.

3.5.3 Missing feature methods

Related to the work on local features is the work on *missing feature methods*, which has shown great promise for noise robust ASR in recent years [18, 75]. These methods allow a conventional recognizer to "ignore" corrupted regions of time-frequency by treating the corresponding observations as missing data, and marginalizing them out in the likelihood calculations. This requires one to determine which regions of a T-F image have been corrupted by noise. Thus, much attention has been given to the estimation of the so-called *spectral mask* indicating which T-F regions belong to the target sound source [92, 86].

This line of work has lead to a more complete system by Barker, Cooke, and Ellis referred to as the *speech fragment decoder* [3]. This system uses bottom-up grouping cues (inspired by auditory scene analysis [11]) to essentially cluster the T-F plane into regions such that the energy in each region is likely to have originated from a single sound source. During decoding, in addition to the typical estimation of a state sequence, the search also simultaneously selects a subset of these T-F regions as belonging to the target speech source. Regions not chosen as part of the speech input are marginalized using standard missing feature methods. Missing feature methods are also discussed in Section 6.1 after presenting the details of our proposed model.

3.5.4 Alternatives to HMMs

Earlier in this chapter, we described several fundamental problems with hidden Markov models, such as the invalid conditional independence assumptions and the poor modeling of duration. There has been a wide variety of work on alternatives to HMMs such as segment-based models [35, 69], and related methods meant to deal specifically with the problem of duration modeling [89]. As mentioned previously, many articulatory-based methods utilize dynamic Bayesian networks (DBNs), which generalize the structure of the HMM graphical model [5, 101]. In additional to DBNs, other architectures have been proposed which are variations on graphical models such as the factorial HMM [34], Boltzmann chains [79], and conditional random fields (CRFs) [36]. The models proposed in this thesis are also based on alternative graphical modeling architectures. These are described in the following chapter.

Chapter 4

Parts-based models for speech

Based on the goals presented in the previous chapter, this chapter describes the proposed "parts-based" approach to acoustic modeling. After discussing the high-level concept, the formal definition is given and explained by showing two complete examples. We then discuss the relationships to typical HMM-based models. The final section of the chapter describes issues related to implementing the model.

4.1 Parts-based models

The proposed model of speech is based on previous work on parts-based models for machine vision. A prototypical vision application of parts-based models is that of face detection. Instead of trying to build a single pattern detector which locates a face as a whole, faces are modeled as a collection of individual *parts.* Local detectors can independently look for the parts (eyes, nose, mouth, etc.), while the overall system attempts to find the set of parts in roughly the correct configuration (e.g., a nose should be in-between and below two eyes, a mouth should be below a nose, etc.).

Such an approach was proposed as early as 1973 by Fischler and Elschlager (Figure 4-1a), where it was framed as an energy minimization problem [31]. More recently, Felzenzwalb and Huttenlocher [30] used a similar strategy, but reformulated the problem probabilistically using graphical models and explored additional applications such as pose detection with a collection of jointed body parts (Figure 4-1b,c). Many other researchers have also recently used the parts-based approach to problems in vision [88, 66, 4]. The methods explored in this thesis follow closely the work of [30].

The high-level idea for creating a parts-based model (PBM) of speech is illustrated in Figure 4-2. The "parts" will be localized detectors, each trying to detect a specific phonetic cue. Figure 4-2a highlights several cues for the letter B (the diphone /b iy/), some of which were discussed in the previous chapter (recall the B example from Figure 3-1, page 30). Figure 4-2b shows how these cues might be encoded with a collection of local patch-based part detectors. The patches detect individual cues, while the spring connections between patches indicate that their relative placement has some flexibility. Therefore, it represents a *deformable template.*

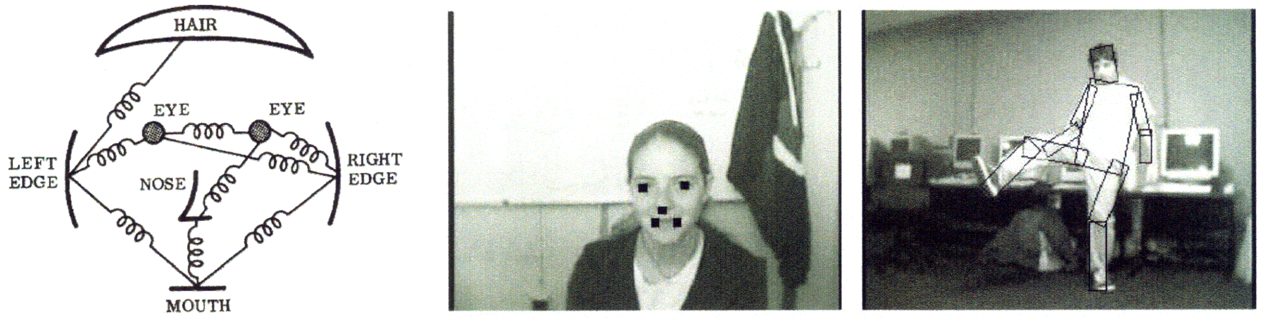


Figure 4-1: The use of parts-based models in vision. (a) Fischler and Elschlager [31] (b) and (c) Felzenzwalb and Huttenlocher [30]. Figures from the original publications.

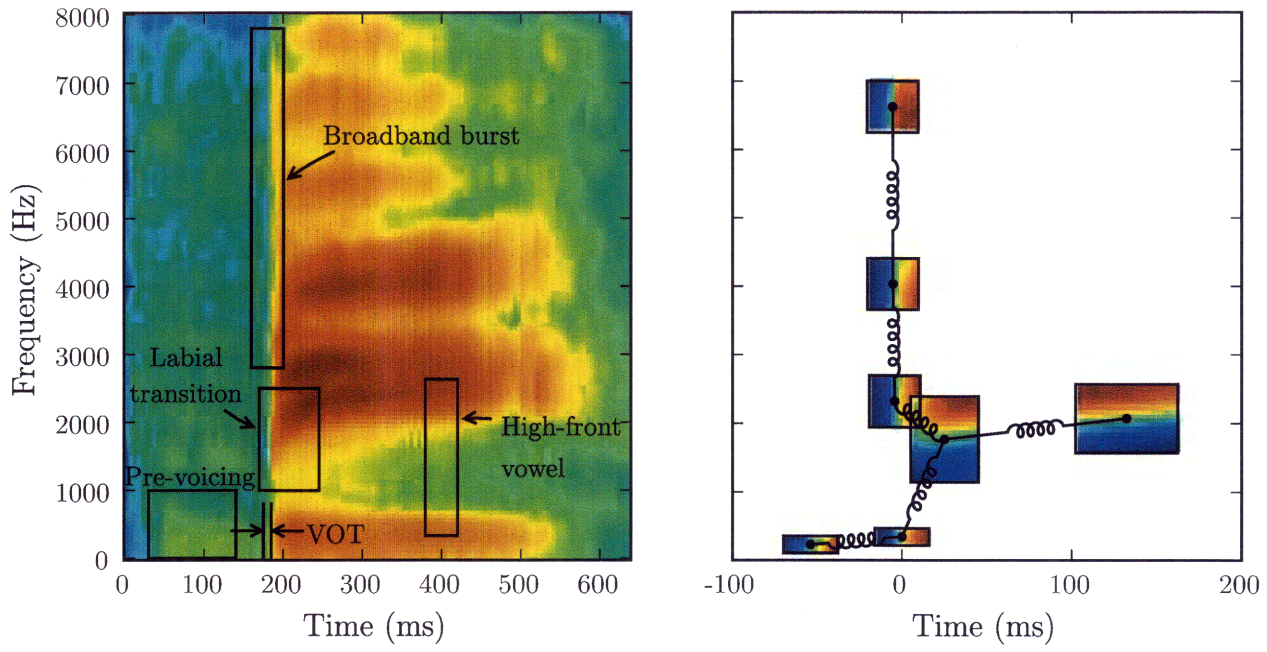


Figure 4-2: High-level description of parts-based modeling for speech. (a) Labeled phonetic cues for the diphone (letter) *B*. (b) The visualization of a parts-based model to capture these cues with localized patch detectors. The generation of this model is described in Section 4.4.1.

4.2 Formal description

Each phonetic unit (e.g. a phone, triphone, syllable, word, etc.) has a model, $\mathcal{M}$, consisting of a set of N localized spectro-temporal cues arranged in a flexible configuration. Part i has one or more "deformable" parameters represented by the variable x_i. The joint configuration of $\mathcal{M}$ is the collection of deformable parameters for all parts, $\mathbf{x} = \{x_1, \ldots, x_N\}$. The model also has an associated graph[1], $G = (V, E)$, with N vertices representing the parameters, x_i, and edges connecting some subset of the pairs of vertices, (x_i, x_j). We will only consider tree-structured graphs – connected graphs without cycles. An objective function, $f(\mathbf{x}, \mathbf{o})$, provides a score of how well a particular configuration, $\mathbf{x}$, matches a particular set of observations, $\mathbf{o}$. In decoding, we are given a set of observations, $\mathbf{o}$, and would like to infer the joint configuration.[2] Therefore, the goal of decoding is to determine the best setting of the flexible parameters,

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmax}} f(\mathbf{x}, \mathbf{o}) \tag{4.1}$$

With $\mathbf{x}^*$ computed, the value $f(\mathbf{x}^*, \mathbf{o})$ provides a score of how well the observed data, $\mathbf{o}$, fits the model, $\mathcal{M}$.

Assuming the dependencies between parts allow them to be represented in a pairwise Markov random field (MRF) (which is the case for trees. See Appendix B), then the objective function can be written as,

$$f(\mathbf{x}, \mathbf{o}) = \sum_{i \in V} f_i^A(x_i, \mathbf{o}) + \sum_{i,j \in E} f_{i,j}^E(x_i, x_j) \tag{4.2}$$

where f_i^A are functions associated with each vertex, and $f_{i,j}^E$ are functions associated with each edge. In the most general setting, we will assume another set of terms associated with each node, f_i^N,

$$f(\mathbf{x}, \mathbf{o}) = \sum_{i \in V} f_i^A(x_i, \mathbf{o}) + \sum_{i,j \in E} f_{i,j}^E(x_i, x_j) + \sum_{i \in V} f_i^N(x_i) \tag{4.3}$$

[1]The model will be expressed within a *graphical modeling* framework. Appendix B provides a brief review of graphical models and presents the notation used in this chapter and throughout this thesis.

[2]While reading the more abstract definition of the model, it may be beneficial to keep in mind a simple special case: the *configuration*, x_i, is the time-frequency location of part i. Thus, the *joint configuration*, $\mathbf{x}$, refers to where (in the T-F plane) to place all the parts.

Maximizing this objective, $f(\mathbf{x}, \mathbf{o})$, by finding the optimal setting of $\mathbf{x}$ is an attempt to satisfy several potentially competing objectives:

- **Acoustic score**: $f_i^A(x_i, \mathbf{o})$

  The parameters of part i, x_i, should be chosen such that this represents a good match to the observed data, $\mathbf{o}$. This can be viewed as a "part detector": a high value of $f_i^A(x_i, \mathbf{o})$ represents the "detection" of part i, having configuration x_i.

- **Edge prior**: $f_{i,j}^E(x_i, x_j)$

  This term encodes the relationships between parts. The configuration of part i can influence the configuration of part j, and this is represented by edge (i, j) in the graph. Because $f_{i,j}^E$ is not a function of $\mathbf{o}$ (it does not depend on the current input observations), it is referred to as a *prior*. A high value of $f_{i,j}^E(x_i, x_j)$ means that the configurations x_i and x_j are highly compatible.

- **Node prior**: $f_i^N(x_i)$

  This term can be used to place a prior on individual parts. The setting of parameters of part i may be biased independently of the configuration of any other parts, and this bias can be encoded in f_i^N. Note that an equivalent model can be specified in which f_i^N is absorbed into the acoustic score by redefining f_i^A.

4.3 Probabilistic interpretation

This model can be treated non-probabilistically (viewed as a *non-probabilistic factor graph* [48, 51]), however there is a natural probabilistic interpretation. $f(\mathbf{x}, \mathbf{o})$ is (if properly normalized) the joint log-likelihood, and decoding solves the *maximum a posteriori* (MAP) estimation problem,

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmax}} \, p(\mathbf{x}|\mathbf{o}) \tag{4.4}$$

where the full form of the objective function (Equation 4.3) is obtained by applying Bayes' rule, making several assumptions on how terms can be factored, and working in the log domain,

$$\begin{aligned} p(\mathbf{x}|\mathbf{o}) &\propto p(\mathbf{x})p(\mathbf{o}|\mathbf{x}) && (4.5)\\ &\propto p(\mathbf{x})\prod_{i=1}^{N} p(\mathbf{o}|x_i) && (4.6)\\ &\propto p(x_r)\prod_{\substack{k=1\\k\neq r}}^{N} p(x_k|x_{\pi(k)})\prod_{i=1}^{N} p(\mathbf{o}|x_i) && (4.7)\\ \log p(\mathbf{x}|\mathbf{o}) &= \log p(x_r) + \sum_{\substack{k=1\\k\neq r}}^{N} \log p(x_k|x_{\pi(k)}) + \sum_{i=1}^{N} \log p(\mathbf{o}|x_i) && (4.8) \end{aligned}$$

where r is the index of the root node, and $\pi(k)$ is the index of the (single) parent of node k. The factoring of $p(\mathbf{x})$ is based on the structure of the directed graph (see Appendix B), and factoring $p(\mathbf{o}|\mathbf{x})$ relates to the assumption that the part detectors will work independently.

4.4 Examples

The preceding description is best understood by looking at some concrete examples. This section will give two related examples to demonstrate the flexibility of the proposed model.

4.4.1 Collection of patches

One way to apply this framework is to treat the spectrogram[3] as an image and follow an approach similar to what has been done in vision (e.g. the face detector implementation in [29]): each part is represented by a fixed patch located at a particular point in the time-frequency (T-F) plane, and the prior encodes the preferred relative T-F location of the parts (recall Figure 4-2b). Motivated by the discussion of the letters B and E in Section 3.1, this section will describe an implementation of approach to model these two letters.

Overview

The variables (the deformable parameters) for each part are the T-F coordinates, $x_i = \{t_i, f_i\}$. The decoding process determines the T-F locations of all parts such that the objective is maximized, i.e. it determines the MAP values $x_i^* = \{t_i^*, f_i^*\}$, $i = 1, \ldots, N$. The number of parts is fixed, and in this example of modeling the

[3]The term *spectrogram* here is used in the general sense of any time-frequency two-dimensional representation. We explored several options, but all examples in this thesis use "conventional" wide-band spectrograms (the log magnitude of a short-time Fourier transform). This choice is discussed in the section on implementation (page 64).

letter B, it was chosen to be seven. The decision on the number and choice of the parts was made subjectively to cover the major known phonetic cues for B (shown in Figure 4-2a): (part 1) F1 onset, (part 2) pre-voicing, (part 3) F2 transition following the burst, (part 4) F2 after "flattening out" in the /iy/, (parts 5,6,7) the broadband burst onset in three different frequency bands. This example will use the directed graph formulation, and all terms in the objective will be proper distributions. (See Appendix B).

Acoustic score

The individual, localized cue detectors will provide a score for an input $\mathbf{o}$, $p(\mathbf{o}|x_i) = p(\mathbf{o}|t_i, f_i)$, which should indicate how well the local time-frequency region centered around point (t_i, f_i) matches the i^{th} cue. These distributions will be modeled with diagonal Gaussians,

$$p(\mathbf{o}|x_i) = \mathcal{N}(\nu_i, \Gamma_i), \quad \Gamma_i = \text{diag}(\gamma_1^2, \ldots, \gamma_{d_i}^2) \tag{4.9}$$

The dimensionality, d_i, is the size of the T-F patch used to detect cue i, and is fixed a priori.

Prior

The prior, $p(\mathbf{x})$, models the arrangement of patches and factors according to structure of the tree,

$$p(\mathbf{x}) = p(x_1, \ldots, x_N) = p(x_r) \prod_{\substack{k=1 \\ k \neq r}}^{N} p(x_k|x_{\pi(k)}) \tag{4.10}$$

where r is the index of the root node, and $\pi(k)$ is the index of the (single) parent of node k. Each conditional distribution in Equation 4.10 will be modeled with a full-covariance Gaussian in two dimensions, encoding the relative position of cues i and j in the time-frequency (T-F) plane,

$$p(x_i|x_j) \propto \mathcal{N}(\Delta x_{ij}; \mu_{ij}, \Sigma_{ij}), \qquad \text{where } j = \pi(i) \tag{4.11}$$

$$\Delta x_{ij} = \{t_i - t_j, \quad f_i - f_j\} \tag{4.12}$$

For the visual models used in [30], the root node is given a uniform prior to provide spatial translational invariance. For speech, however, it is likely that we *do* want the location of a cue to be biased to a particular absolute frequency. Therefore, we use a Gaussian prior on the frequency location of the root node $(x_r = (t_r, f_r))$,

$$p(x_r) = p(f_r) \propto \mathcal{N}(\mu_r, \sigma_r^2) \tag{4.13}$$

Notice that $p(x_r)$ is independent of time, making the overall model time-shift invariant – a desirable property for speech.

Training

In this example, all the training data has the locations of each of the N cues marked as follows. For 16 randomly chosen examples for each class, the T-F position of each of the N parts was labeled by hand. A simple graphical interface was used which enabled a human labeler to simply click on N points (the patch centers) in a spectrogram. Therefore, manual labeling of 16 examples only takes 2–3 minutes. For the rest of the training set, the cue locations were labeled by dynamic alignment in time *and frequency* to one of the original 16 (the one with the best match). With all points labeled and because we are using simple Gaussian distributions, most of the parameters are simply given their maximum-likelihood settings by using the sample mean and sample variance.

Local cue detectors: ν_i, Γ_i The individual cue detectors model the energy in T-F regions surrounding the location of each cue. A patch of the spectrogram is extracted around all training points, from which diagonal Gaussian distributions are trained. Since we are given the locations of the training points, we can trivially do maximum-likelihood training.

Deformable model: μ_{ij}, Σ_{ij} With the location of the cues labeled in the training data, we can simply use maximum-likelihood training to estimate full-covariance Gaussian parameters, μ_{ij}, Σ_{ij}, to model pairwise distances between each (i, j) pair of cues. Because we are restricted to trees, the factored form of $p(\mathbf{x})$ will require using only $N-1$ of these $N(N-1)/2$ distributions. However, computing all pairs on the training set can be used to learn the tree structure, as described below.

Tree structure: r, π As mentioned above, efficient decoding requires that the factorization of $p(\mathbf{x})$ takes the form of a tree (see also Appendix B). One possibility is to arbitrarily choose a simple factorization, such as choosing a root and having all other nodes as its children. However it is also possible to learn the "optimal" structure of the tree from the training data [30].

Given M independent training utterances with configurations $p(\mathbf{x}^m)$, and using Equation 4.10, the total log-likelihood on the training data is,

$$\log \prod_{m=1}^{M} p(\mathbf{x}^m) = \sum_{m=1}^{M} \log p(x_r^m) + \sum_{\substack{k=1 \\ k \neq r}}^{N} \sum_{m=1}^{M} \log p(x_k^m | x_{\pi(k)}^m) \tag{4.14}$$

The two terms on the right correspond to choosing the root node, r, and choosing the structure of the graph, π. We can choose these independently,

$$r^* = \operatorname*{argmax}_r \sum_{m=1}^{M} \log p(x_r^m) \tag{4.15}$$

$$\pi^* = \operatorname*{argmax}_\pi \sum_{\substack{k=1 \\ k \neq r}}^{N} \sum_{m=1}^{M} \log p(x_k^m | x_{\pi(k)}^m) \tag{4.16}$$

Consider a fully-connected, undirected graph, G, with N nodes, and edges (i, j) having a weight set to the total training data log-likelihood of the corresponding two points, $\sum_{m=1}^{M} \log p(x_i^m | x_j^m)$. The solution to Equation 4.16 corresponds to finding the *minimum spanning tree* (MST) of G, which can be solved efficiently (e.g. with Prim's algorithm [21]). The root node is simply chosen as the point which maximizes Equation 4.15.

Resulting models

This model was trained on 200 B and E ISOLET utterances, and a visualization of the resulting models is shown in Figure 4-3. Notice that these models directly capture some of the relevant phonetic cues discussed in Chapter 3. In the B model, the transition out of a labial stop is suggested by the horizontal edge in filter 3 having a slightly positive slope, and occurring at a lower frequency than point 4. In contrast, the E model has parts 2 and 3 reliably occurring at the same frequency, and the filter of part 2 forms a right angle at the onset of F2. Parts 5, 6, and 7 in B capture a sharp common onset time at a range of frequencies, encoded as the very small variance in the time direction of these points. Also, edge $1 \rightarrow 5$ in B directly encodes the voice-onset time (VOT), which in this model is very short. This measurement could aid in, for example, the discrimination of B from P [87]. Classification results and further discussion of this model are presented in the next chapter.

4.4.2 A "speech schematic" model

The previous example created a model consisting of fixed patches with flexible T-F locations. This has some of the desired properties discussed in Chapter 3, but there are several characteristics which motivate an alternative way to construct a parts-based model. In viewing the models of several letters trained as described above, most part detectors end up being essentially *edge detectors*. Simple edges (in either time or frequency) cover most of the phonetic cues of interest: horizontal edges for formants, vertical edges for onsets and offsets, etc. If the parts are assumed a priori to be simple edges, this greatly reduces the number of parameters in the model and has computational benefits (discussed later in the chapter).

Further motivation for another PBM variation comes from the classic speech perception work on the "pattern playback machine" by Liberman, Delattre, Cooper, et

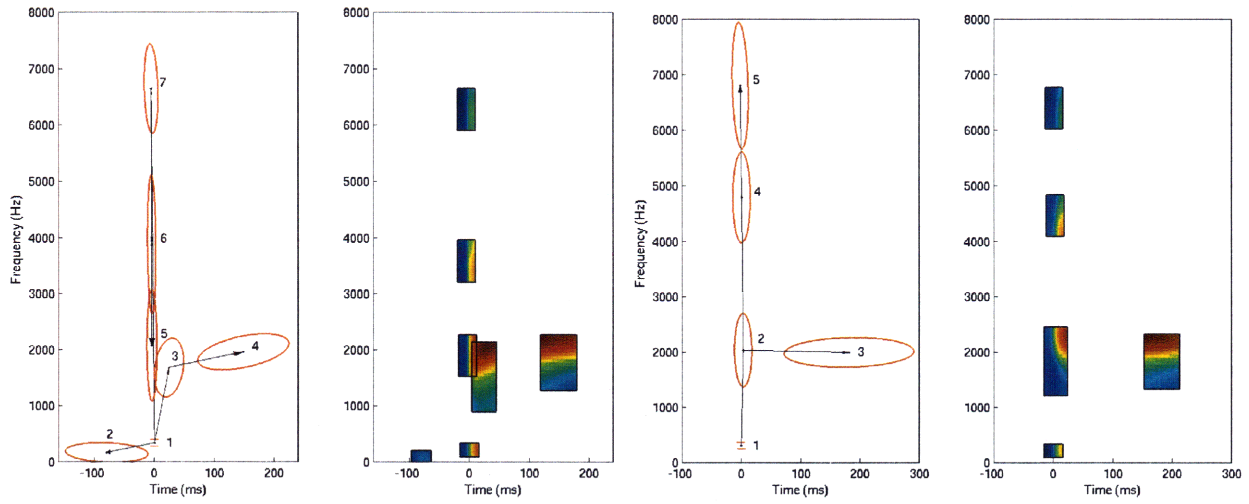


Figure 4-3: Visualizations of trained PBMs for B (left) and E (right). On the left for each is a representation of the prior, $p(\mathbf{x})$. Arrows indicate the parent/child relationship in the graph, and each ellipse (showing an iso-contour line of a Gaussian) indicates the distribution of the location of a part, given its parent's location. The root node has a distribution only in the frequency dimension (two horizontal lines indicate $\mu_r \pm \sigma_r$). The figure on the right for each letter shows the mean vectors for the cue detectors, ν_k in $p(o|x_k)$, arranged in the configuration which maximizes $p(\mathbf{x})$.

al. [56, 20]. To systematically study human perception of speech, they drew simplified versions of spectrograms and created a machine able to synthesize audio having a spectral envelope matching the drawing. An example of this is shown in Figure 4-4 (for a historical review of the work of Liberman and colleagues at Haskins Laboratories see [55]). Note that this simplified version of speech captures the essential T-F patterns of interest. Therefore, the second PBM variation we will present consists of a collection of simple edge detectors meant to directly model the shape of these major features in the spectrogram. This will be referred to as a "speech schematic" model – a model that mimics how a simplified picture of a spectrogram would be drawn to depict a given phonetic unit.

An example for the letter B is shown in Figure 4-5. Notice how it represents the basic structure of the strong edges in a typical B spectrogram (see Figure 4-2a on page 46 for an example). This model was created by hand with five parts meant to capture the basic structure of the T-F representation of a B utterance. This model has the following properties,

- The parts are simple edge detectors, with scores given by a dot-product with ± 1 patches (in the figure, darker regions being $+1$ and lighter being -1). For each part, each setting of the parameters defines a binary patch, $h(t, f)$, which when applied to the input observed spectrogram, $\mathbf{o} = o(t, f)$, gives an acoustic score of,

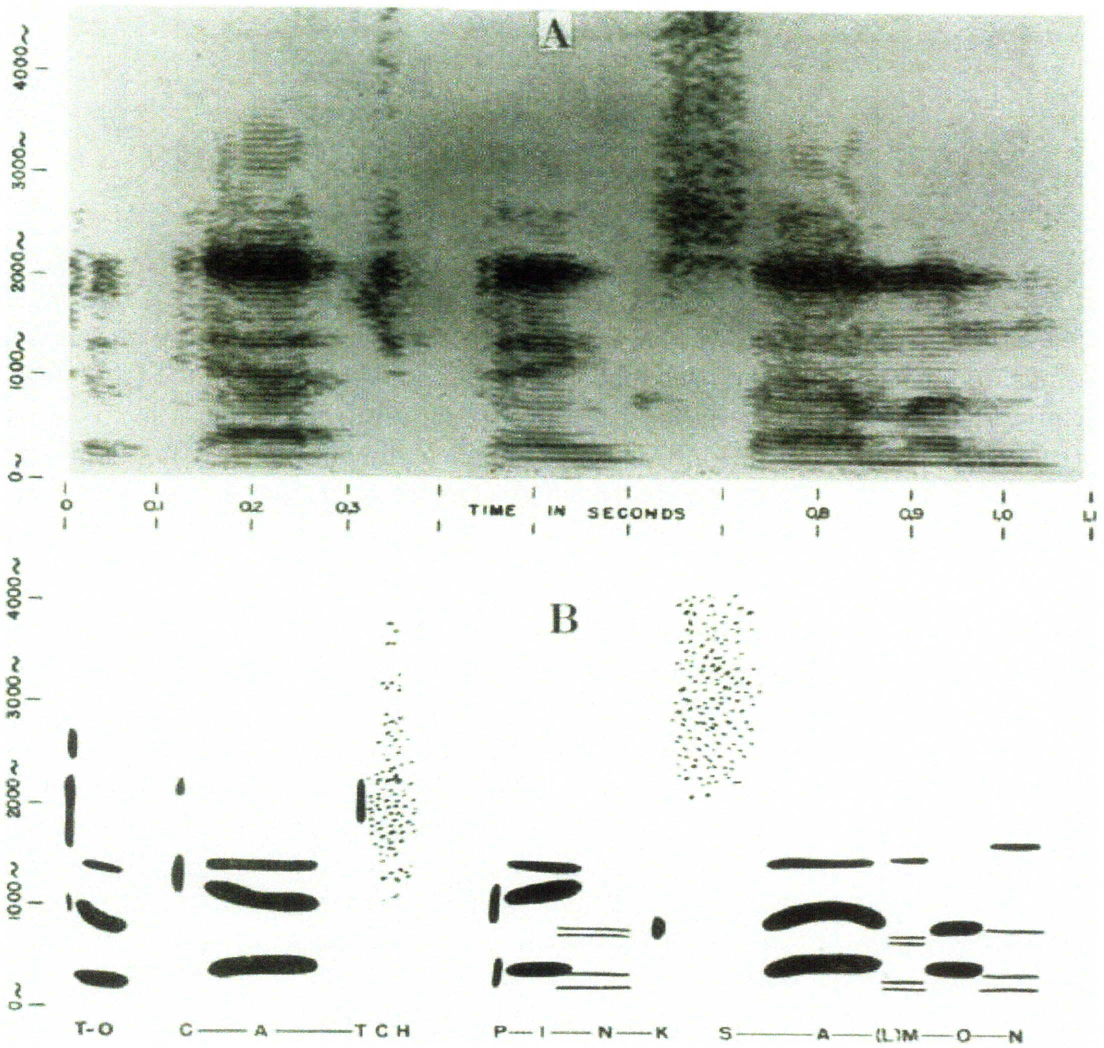


FIG. 1. SHOWING SPECTROGRAM (A) AND SIMPLIFIED VERSION (B) OF A SPOKEN PHRASE

Figure 4-4: Figure from Liberman et al. [56] demonstrating the conversion (in this case, by hand) of a spectrogram to a simplified drawing, or a "speech schematic".

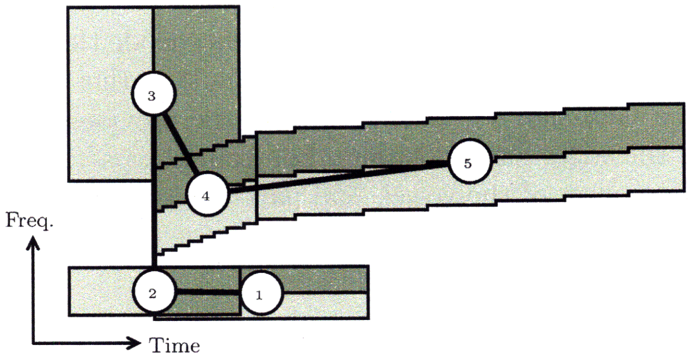


Figure 4-5: A speech schematic model of B constructed of edge detectors. The nodes and thick black line represent the underlying graph. The corresponding edge detector for each node is shown in a common configuration.

$$f_i^A(x_i, \mathbf{o}) = \sum_{h(t,f)=+1} o(t,f) - \sum_{h(t,f)=-1} o(t,f) \tag{4.17}$$

Unlike the Gaussian log-likelihood scores used in the previous section, these acoustic scores are not based directly on probability distributions, but rather the log of some node potential function (an unnormalized distribution).

- While both this model and the previous model are based on localized patches, in this model the patches are *not necessarily fixed.* Parts 4 and 5 in the figure are sloped edge detectors, and the slope can be made a deformable parameter. Therefore, the acoustic score for these parts is not found by sliding a single fixed T-F filter over the spectrogram. Different settings of the parameters have different corresponding patches. As will be discussed, we believe this to be an important property.

- Like the previous example, edges between parts encode the relationship between the parts' parameters. However, in this example, rather than just providing a weight (e.g., a Gaussian log-likelihood as in the previous example), the model makes use of the edges for *constraints.* For example, part 4 tries to measure the F2 slope at the onset of the vowel, and part 3 tries to detect the overall onset. The edge connecting parts 3 and 4 can encode the constraint that part 4 begins exactly at the point of onset that part 3 detects.

Examples using this model will be given in Chapter 5.

4.5 Classification

To perform classification with PBMs, we can create a set of models – one for each class, $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_C$. In the probabilistic setting, we would like to choose the model with the highest MAP probability,

$$\mathcal{M}^* = \underset{\mathcal{M}}{\operatorname{argmax}} P(\mathcal{M}|\mathbf{o})$$

To do so, we make the following approximation,

$$P(\mathcal{M}|\mathbf{o}) \approx \max_{\mathbf{x}} P(\mathbf{x}|\mathbf{o})$$

In other words, the score for a model is assumed to be the score for the best scoring setting of that model's parameters – as opposed to considering all possible settings of the parameters (requiring a summation over $\mathbf{x}$). This is equivalent to the *Viterbi approximation* used in HMM-based ASR (Section 2.2). Another method of classification that will be used in Chapter 5 is to simply decode with each model, and use the MAP setting of the parameters, $\mathbf{x}^*$, as features for another simple classifier. For example, it will be shown that taking the slope values of parts 4 and 5 in the B model can provide strong features for discrimination.

As an example, classification can be performed with the B and E models presented in Section 4.4.1 (and shown in Figure 4-3 on page 53) by simply choosing the model with the higher posterior probability. The resulting binary classification errors are shown in Table 4.1. In both clean and noise conditions, the PBM outperforms the baseline HMM by over 30% relative error.[4]

	HMM	PBM	ΔError
Clean	9.00%	6.17%	-31.44%
Noise	25.50%	22.50%	-33.33%

Table 4.1: Classification error rates (%) of B vs. E classification on 600 tokens. HMM is the baseline recognizer (see Appendix A). PBM refers to patch-based models shown in Figure 4-3.

4.6 Comparison to HMMs and segment-based models

It is useful to discuss some of the similarities and differences between the proposed parts-based model (PBM) other ASR techniques: HMM-based and segment-based models. HMMs were discussed in Chapter 2. Segment-based models [35] seek to model speech as a sequence of variable-length segments, as opposed to a sequence of

[4]The noise condition has a variety of additive noise types at a range of signal-to-noise ratios. See Appendix A for more information on the dataset.

fixed-length frames as in HMMs. The overall structure of these models is compared in Figure 4-6.

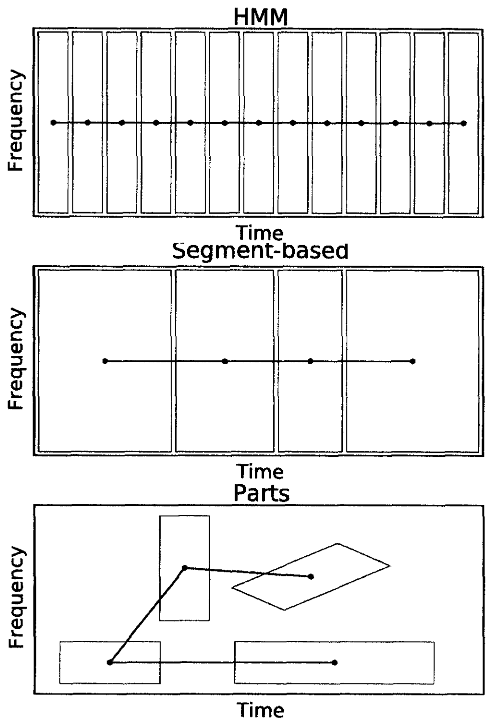


Figure 4-6: Comparison between models. (a) HMM/MFCC: sequence of fixed-duration observations. (b) Segment-based: sequence of variable-length segments, (c) PBM: deformable template of localized events.

4.6.1 Decoding comparison

The similarities and differences to standard HMM-based ASR are most clearly seen by looking at the decoding process. Figure 4-7 compares the two models. Both can be viewed as inference in a graphical model. In the HMM, there is an unknown variable for each frame in the input utterance. Viterbi decoding assigns a *state label* to each unknown. In the parts-based model, the unknowns are the "flexible parameters" for each part. Given an input utterance, the decoder determines the optimal setting of these parameters such that the objective function is maximized. The example shown in Figure 4-7 consists of parts with time-frequency coordinates as their flexible parameters. Both HMMs and PBMs are based on the same basic dynamic programming algorithm to determine the optimal setting of unknowns. The primary differences being, (1) in the PBM, the Markov chain is generalized to a tree, and (2) the PBM has a more general relationship between observations and unknowns. Rather than each hidden variable being associated with one observation variable, all observed data (i.e., the spectrogram) is available to all hidden variables.

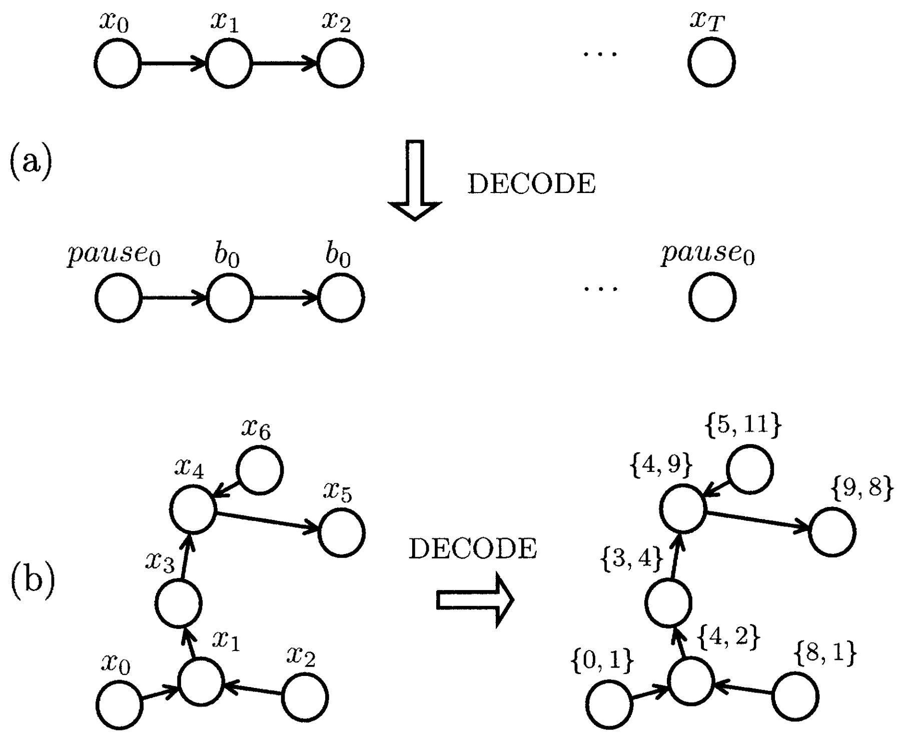


Figure 4-7: Examples comparing the HMM and PBM decoding process and their unknowns. (a) An HMM decoder assigns state labels to each unknown state variable. (b) In the PBM decoder, the unknowns are the values of the "flexible" parameters of the deformable model, here, the T-F coordinates, $x_i = \{t_i, f_i\}$. The structure of the model has been generalized from a chain to a tree.

4.6.2 Information in edges

One key benefit to the proposed PBM techniques is the amount of information encoded in the edges of the corresponding graphical model. The edges of the graph represent opportunities for nodes to "pass information" to their neighbors which enforce prior constraints or apply weights based on prior distributions. In an HMM, the edges represent transitions between states, as encoded in the state transition probabilities. The allowable state transitions enforce the possible ordering of states (e.g. left-to-right topology, or skip states, etc.). The values of the transition probabilities implicitly create prior duration models on the states (which are not accurate as discussed in Section 3.2.6). The proposed PBM generalizes this substantially. More "information in the edges" can give the model a much stronger *prior* model of speech. Figure 4-8 shows a few potential examples of how edge scores can encode prior information in the model.

4.6.3 HMMs as a special case

The parts-based model as represented by Equation 4.3 is very general. HMMs can be viewed as a special case in two different ways: the parts can either represent *frames* or *states*. Both will give identical scores to the standard HMM. We briefly describe each and summarize the details in Table 4.2.

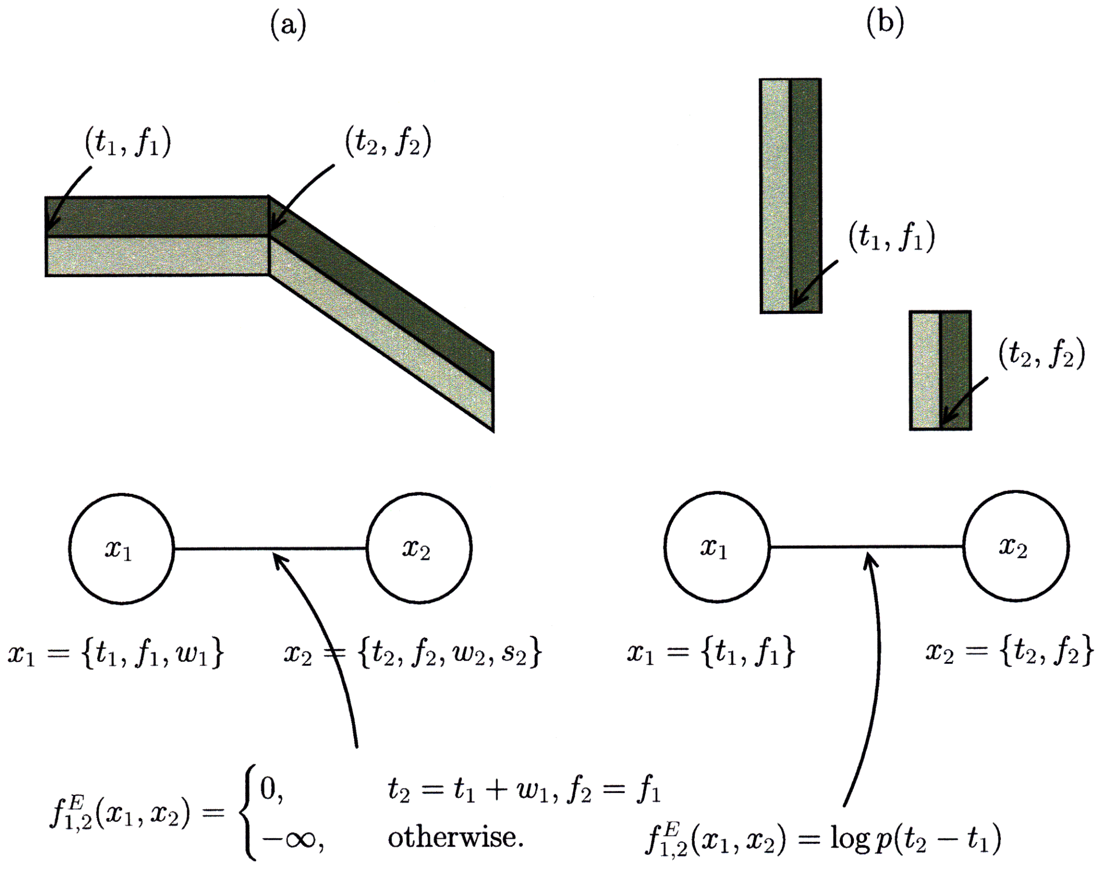
Parts as frames

An exact equivalence to HMMs can be made by simply constructing a PBM with one part (i.e. one node in the graph) for each *frame* in the input utterance to be decoded. In this case, the flexible parameters, x_i, will take on values of the state identity: $x_i \in \{s_1, \ldots, s_N\}$, for an N-state HMM. This is seen directly equating the graphical models (a) and (b) from Figure 4-7. Note that in this case, there is technically a different model for each input utterance. With one part for each frame, the number of parts changes according to the length of the utterance.

This equivalence can also be seen by comparing the decoding objective functions. Recall the total likelihood function for HMMs (Equation 2.7 from Section 2.2.3),

$$p(\mathbf{x}, \mathbf{o}) = P(x_1)p(o_1|x_1)\prod_{t=2}^{T} p(o_t|x_t)P(x_t|x_{t-1}) \tag{4.18}$$

By taking the logarithm and rearranging, we can put in a form similar to the PBM objective function (Equation 4.3),



(a)
(b)
(t_1, f_1)
(t_2, f_2)
(t_1, f_1)
(t_2, f_2)
x_1
x_2
x_1
x_2
$x_1 = \{t_1, f_1, w_1\}$
$x_2 = \{t_2, f_2, w_2, s_2\}$
$x_1 = \{t_1, f_1\}$
$x_2 = \{t_2, f_2\}$
$f^E_{1,2}(x_1, x_2) = \begin{cases} 0, & t_2 = t_1 + w_1, f_2 = f_1 \\ -\infty, & \text{otherwise.} \end{cases}$
$f^E_{1,2}(x_1, x_2) = \log p(t_2 - t_1)$

Figure 4-8: Two examples showing the variety of information that can be encoded in "edge priors". (a) *Continuity*: the edge score $f^E_{1,2}(x_1, x_2)$ imposes a hard constraint that the edge in part 1 aligns with that in part 2 (i.e. the constraint makes T-F points coincide, $\{t_1 + w_1, f_1\} = \{t_2, f_2\}$). (b) *Duration*: A distribution, $p(\cdot)$, on the "distance" between two parts can encode duration.

$$\begin{aligned} f(\mathbf{x}, \mathbf{o}) &= \log p(\mathbf{x}, \mathbf{o}) && (4.19) \\ &= \underbrace{\sum_{t=1}^{T} \log p(o_t | x_t)}_{\substack{\text{acoustic score,} \\ f^A_i(x_i, \mathbf{o})}} + \underbrace{\log P(x_1)}_{\substack{\text{node prior,} \\ f^N_i(x_i)}} + \underbrace{\sum_{t=2}^{T} \log P(x_t | x_{t-1})}_{\substack{\text{edge prior,} \\ f^E_{i,j}(x_i, x_j)}} && (4.20) \end{aligned}$$

Parts as states

Instead of one part per frame, one can construct a PBM with one part per HMM *state*. This is equivalent to viewing it as segment-based model, such as in Figure 4-6b. In this case, the flexible parameters denote the start frame and the end frame of each state (or any other equivalent parameterization, such as start frame and duration). Therefore, decoding determines the segmentation of an utterance into states.

Definitions:

N	Number of states, $s_1, \ldots, s_N$.
T	Number of frames, i.e. number of observations $\mathbf{o} = o_1, \ldots, o_T$.
i	Part index.
S_k, E_k	State k begins at frame S_k and ends on frame E_k.
$p(o_t \mid s_k)$	GMM likelihood for frame t, state k.
p_k	HMM transition probability for leaving state k ($k \to k+1$).
$1 - p_k$	HMM transition probability for staying in state k ($k \to k$).
π_k	Probability HMM begins in state k.

	Parts as Frames	**Parts as States**
	Frame-based	Segment-based
Number of parts	T	N
Flexible parameters	$x_i = s_i \quad (x_t = s_t)$	$x_i = \{S_i, E_i\}$
Acoustic Score, $f_i^A(x_i, \mathbf{o})$	$\log p(o_i \mid x_i)$	$\sum_{t=S_i}^{E_i} \log p(o_t \mid s_i)$
Edge Prior, $f_{i,j}^E(x_i, x_j)$	$\begin{cases} \log(1-p_i), & x_{i+1} = x_i \\ \log p_i, & x_{i+1} = x_i + 1 \\ -\infty, & \text{otherwise.} \end{cases}$	$\begin{cases} 0, & S_{i+1} = E_i + 1 \\ -\infty, & \text{otherwise.} \end{cases}$
Node Prior, $f_i^N(x_i)$	$\log \pi_i$	$(E_i - S_i)\log(1-p_i) + \log p_i$

Table 4.2: Two ways to view a strictly left-to-right N-state HMM as a special case of a parts-based model. Both are linear chains of variables (length T and N respectively). The last three rows define the objective function such that the MAP decision will be the same for all cases. Note that scores of $-\infty$ represent hard constraints.

4.7 Implementation

The final section of this chapter deals with some practical issues on the implementation of the proposed model. Subsections will discuss the choice of data representation, efficient methods for acoustic scoring, and a description of the decoding algorithm.

4.7.1 Spectral representation

In the parts-based framework, the information from a given input is contained in a set of observed variables, $\mathbf{o}$. In general, the observations can be any information extracted from the signal. However, for all the examples used in this thesis, the observations consist of a single time-frequency representation (spectrogram) of the input utterance (some alternatives are discussed in Section 6.2.3 as possible future work). As shown in the examples earlier in this chapter, the acoustic scores, f_i^A, represent pattern detectors operating on this time-frequency image.

There are many possibilities when choosing the spectral representation, including conventional spectrograms [73], Mel-spectrograms [24], and various auditory models [42, 12]. We have chosen to use a conventional DFT-based wideband spectrogram (the details are described in Appendix A). This representation largely captures the features of interest, particularly emphasizing the formant locations and transitions. Commonly used auditory models often have the low harmonics resolved, giving rise to strong edges that which are not directly related to the phonetic identity. Also, subjectively, many strong formant transitions appear as roughly straight lines on a linear frequency axis, but are curved on a log frequency axis. Keeping a T-F representation with linear frequency allows the use of edge detectors with simple linear slopes.

Also note that some desirable properties of other T-F representations can be "simulated" with the DFT-based representation by an appropriate redefinition of the patches. For example, a Mel-spectrogram is computed from a conventional spectrogram with a simple linear transformation. The effect of the triangular Mel filters can be absorbed into the definition of the patches. The fact that the critical bands of human hearing increase with frequency can be (partially) accounted for by having wider (in frequency) patches at higher frequencies (although the bandwidth of filters in the underlying filterbank of course stay constant). If constrained to having patches of constant bandwidth (vertical height), then the conventional spectrogram may not be appropriate. Future work may explore alternative spectral representations, but for these reasons, we believe the conventional wideband spectrogram is sufficient to explore parts-based modeling techniques.

4.7.2 The integral image for fast acoustic scoring

From the chosen two-dimensional spectral representation, the acoustic feature detectors typically consist of local "patches" applied to this image. Decoding with a set of models may require a large number of patches, which are typically evaluated many times in the course of decoding (perhaps evaluated at all T-F points). This can lead to prohibitive computational costs. Also, these costs rise (linearly) with the resolution of the underlying T-F representation. For a patch of fixed duration, increasing the time resolution (frame-rate) by a factor of k increases the size of the patch by a factor of k. Likewise for the frequency resolution.

In this section, we describe methods of using *integral images* for a class of patch-based acoustic scoring techniques. This substantially reduces the amount of computation needed, and makes the costs independent of the T-F resolution (i.e. constant with respect to patch size). We present simple extensions of previous methods used in machine vision to other features of interest for speech applications, and discuss some implementation details.

Definition

For an N-by-M matrix with elements $x[i, j]$, its *integral image* is an N-by-M matrix with elements $I_x[i, j]$,

$$I_x[i,j] = \sum_{k=1}^{i} \sum_{l=1}^{j} x[k,l] \tag{4.21}$$

This is simply the running cumulative sum of the elements of the matrix (Figure 4-9a).[5] With the integral image pre-computed and stored, many useful features can be quickly computed because of the redundancy of summing neighboring regions of the input. For example, a simple edge detector can take the form of a dot-product with a simple patch of values ± 1 such as shown in Figure 4-9b (recall the types of patches used for model described in Section 4.4.2). Direct computation requires looping over all points within the patch region, and this would be repeated at any all T-F locations that the patch is evaluated. Using an integral image, each score can be computed with only 6 additions (and two multiplications by 2) from pre-computed values of the integral image, as indicated by the points in Figure 4-9c. The weighted areas cancel each other out to leave the desired ± 1 weights over the correct regions.

Extensions

The work of [91] used integral images to efficiently compute dot-product features consisting of horizontal and vertical edges. Here, we describe generalizing the concept to other features relevant to the models of this thesis.

Sloped edge detectors As discussed previously (Section 3.2.3), formant transitions are a very important phonetic cue, and we would like to detect them with diagonal (sloped) edge detectors. Integral images were originally proposed for patches having only horizontal and vertical edges [91]. Others have since extended their use to 45 degree diagonal edges [59]. In order to reliably detect more subtle formant

[5] All images here are shown with the origin in the lower-left corner, to match the typical display of spectrograms.

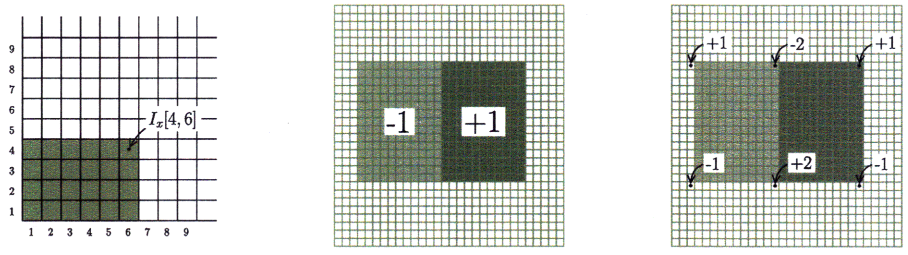


Figure 4-9: The use of integral images. (a) Definition: $I_x[a,b]$ is the sum of all points in $x[i,j]$ s.t. $i \leq a, j \leq b$. (b) A vertical edge onset detector based on subtracting the sum of one region from the sum of another. (c) Achieving the computation of (b) using six pre-computed values from the integral image.

slopes than only 0°, ± 90°, or ± 45°, we present a simple extension of the integral image concept to arbitrary slopes.

Conceptually, this is straight forward: a separate integral image can be computed for each desired slope. The region of summation for other classes of slopes is shown in Figure 4-10. With these integral images computed, computing dot-products with piecewise constant regions is analogous to the original case. An example for a two-region edge detector is shown in Figure 4-11. We will only consider sloped patches taking the shape of a parallelogram with vertical edges on the left and right. Other variations can be efficiently computed, but this case is sufficient for the types of features we wish to detect.

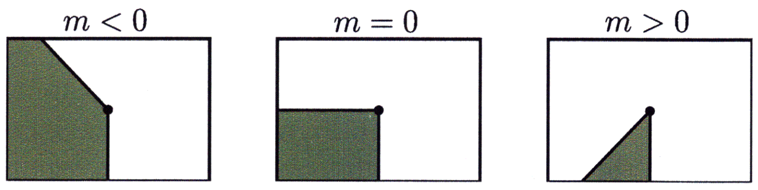


Figure 4-10: Summation ranges for sloped integral images, for slope (a) negative, (b) zero, (c) positive. (The origin is in the lower-left).

With arbitrarily sloped integral images, there is a complication that must be handled appropriately. The root of the problem is the fact that discretized slopes are not "shift-invariant". For integer $i = 0, 1, 2, \ldots$ and floating point slope m, let $y_i = \text{round}(mi)$. Unlike the simple cases of $m \in \{0, \pm 1\}$, $y_{i+k} - y_i$ is, in general, not a constant with respect to i for all fixed k. Because integral image computations are based on perfect cancellation of terms, even off-by-one indexing problems can lead to large errors.

A solution to this problem is to pre-compute and store the line $\ell_j = \text{round}(mj)$, $j = 1, \ldots, M$ for each slope required (M is the maximum spectrogram width required). Then, all calculations using m use must this line to measure distances with respect to the origin, rather than with respect to the point being evaluated. This illustrated in Figure 4-12. In this example, the goal is to compute a sum over the region shaded gray (a sloped patch with $m = 0.7$), specified by having height H and width W and an upper-right corner of (i, j). With the right integral image, this sum can be computed with just three additions/subtractions from four points in the integral image, $I_A - I_B - I_C + I_D$.

Given the point A in Figure 4-12, the step that takes careful consideration is choosing the vertical coordinate of point C. Naively, one would use $j_C = i - \text{round}(mW)$, which works for the simple cases of $m = 0$ and $m = 1$. However, due to this shift-invariance problem of arbitrary m, the cancellation in the original integral image is not certain. With line ℓ_j pre-computed (for $j = 1, \ldots, M$), the correct point can be chosen if the vertical displacement is taken with respect to the origin via ℓ, $j_C = i - (\ell_j - \ell_{j-W})$. Using this method guarantees correct computation, and keeps the calculation very fast. It only requires one extra array lookup for each score, and storing a length M array for each slope.

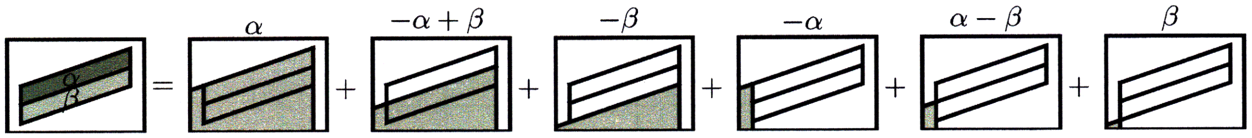


Figure 4-11: Decomposition of a sloped edge detector into six values of a sloped integral image. The patch on the left has two regions of constant value, α and β respectively. On the right, the sum over each shaded region is contained in one value of the integral image. The corresponding weight for each term is shown above.

Gaussian log-likelihoods Acoustic scoring based on diagonal-covariance Gaussian log-likelihoods can also be done efficiently with integral images if the mean and variance are both piecewise constant over a small number of regions. For a multivariate Gaussian distribution, $\mathcal{N}(\mu, \Sigma)$, with dimensionality d, the log-likelihood function, $f(x) = \log p(x)$, is given by,

$$f(x) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) \tag{4.22}$$

For the case of diagonal-covariance matrices, $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_d^2)$. Define $\lambda_i = \sigma_i^{-2}$. $|\Sigma| = \Pi\lambda_i$, and Equation 4.22 can be written as,

$$f(x) = -\frac{d}{2}\log(2\pi) + \sum_{i=1}^{d}\log\lambda_i - \frac{1}{2}\sum_{i=1}^{d}\lambda_i(x_i - \mu_i)^2 \tag{4.23}$$

The final term can be re-arranged by expanding the squared term,

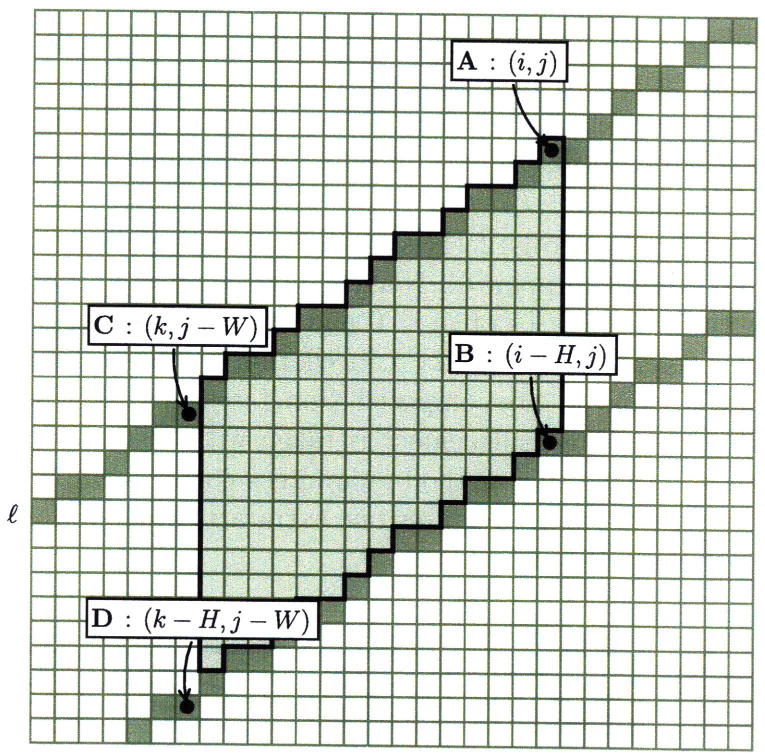


Figure 4-12: Illustration of the computation of a summation over the shaded region using an arbitrarily sloped integral image. Given point A, point C must be determined using the pre-computed discretized slope, line ℓ_j.

$$\sum_{i=1}^{d} \lambda_i (x_i - \mu_i)^2 = \sum_{i=1}^{d} \lambda_i x_i^2 - 2 \sum_{i=1}^{d} \lambda_i x_i \mu_i + \sum_{i=1}^{d} \lambda_i \mu_i^2 \tag{4.24}$$

Simplifying and re-arranging terms we have,

$$f(x) = \sum_{i=1}^{d} \lambda_i \mu_i x_i - \frac{1}{2} \sum_{i=1}^{d} \lambda_i x_i^2 + b \tag{4.25}$$

where b is a constant with respect to the input, x, and thus can be pre-computed,

$$b = -\frac{d}{2} \log(2\pi) + \frac{1}{2} \sum_{i=1}^{d} \log \lambda_i - \frac{1}{2} \sum_{i=1}^{d} \lambda_i \mu_i \tag{4.26}$$

For constant mean and variance, $\lambda_i = \lambda$ and $\mu_i = \mu$ (or a small number of constant regions), Equation 4.25 can be evaluated quickly if we have the integral image for x

and x^2,[6]

$$f(x) = \lambda\mu \sum_{i=1}^{d} x_i - \frac{1}{2}\lambda_i \sum_{i=1}^{d} x_i^2 + b \tag{4.27}$$

Note that we can also use arbitrary functions of $\sum x_i$ and $\sum x_i^2$. Using quadratic functions offers a wider range of features, for example, having a part depend on overall level. Note that the simple ± 1 symmetric edge detectors are level-independent: scaling the input by a constant does not change the value. We can use quadratic features (and do so efficiently) to produce scores which have some desired absolute output value.

Additional notes

Here, we give several other comments on the time and space requirements for implementing integral images for acoustic scoring. If computing sloped integral images, the storage requirements for K slopes and N-by-M matrices is $O(KNM)$, which can become prohibitively large if there are many different slopes (large K) and if the utterances are long (large M). Note, however, that the integral images can be computed in a running fashion. If the longest duration patch is m, then only that amount of history is needed, and the storage requirements can be reduced to $O(KNm)$. With useful patches most likely between tens and hundreds of milliseconds, $m \ll M$ and these savings can be significant.

The computation time required for computing the integral images will generally be small relative to the decoding process, but if not done properly, it can become prohibitive for large K. It is straightforward to compute an integral image with one pass over the data [91]. Note that computing K sloped integral images can also be done efficiently. With slopes sorted largest to smallest, the summation regions at a given point are nested such that their computation can be shared.

Finally, some computation can be saved by handling the boundary conditions appropriately. The definition given was that the integral image for an N-by-M matrix was also N-by-M. In practice, however, it is useful to construct an $(N+1)$-by-$(M+1)$ integral image, where the first row and first column are all zeros. In this case, computing sums over *any* region of the original image can be done efficiently without the need to check for boundary conditions.

4.7.3 Decoding: the max-sum algorithm

The MAP estimation problem (Equation 4.1) for tree-structured graphical models can be solved efficiently with the *max-sum algorithm*.[7] This is a generalization of

[6]Note: one could also utilize the integral image of $(x_i - \mu_i)^2$. However, here, the computational advantages will disappear if different features/models have different μ.

[7]This algorithm and related algorithms have a number of similar, confusable names. If dealing with probabilities instead of log-probabilities, the sums become products, and it is often called the

the well-known Viterbi decoding algorithm used for HMMs [74], where the Markov chain of hidden variables is allowed to have the more general structure of a tree. This section will give a brief description of the algorithm. The description here will use terminology of graphical models provided in Appendix B. Additional descriptions of this and related algorithms can be found in the references [6, 43, 48].

The psuedocode is shown in Algorithm 4.1. Starting with an undirected tree, $G = (V, E)$, choosing a single root node defines a directed tree. From this directed tree, a *topological sort* is performed, resulting in a linear ordering of the nodes such that each node occurs before any node connected to it with an outbound edge [21]. The convention used here is that all edges point towards the root, rather than away from the root. The sorted list of nodes begins at the leaves (of which there can be one or more), and ends at the root (of which there is exactly one). In general, there are multiple ways to sort the nodes, but they all will give the same result. However, different orderings can affect the amount of computation required.

The running time is $O(L^2N)$, where N is the number of nodes and L is the number of possible values for each node. One must loop through each edge (the number of edges in a tree is $N-1$), and for each edge consider the cross-product of the state spaces of the two nodes on that edge (L^2). Note that there are a total of L^N possible settings of the parameters, and yet the optimal parameter setting can be found in a time linear with N. This is possible due to the "optimal substructure" in the problem which is exploited by dynamic programming.

Note the similarities between the max-sum algorithm and the Viterbi decoding algorithm for HMMs (shown on page 28). For an HMM, it is natural to choose the final frame as the root node, in which case the only possible topological sort is the temporal ordering of the nodes, $t = 1, \ldots, T$. At this point, it can be seen to be a simple special case of Algorithm 4.1. The only difference is that in the case of trees, a node can receive "messages" from multiple nodes rather than a single node.

In practice, constraints between the parts often allow the number L^2 to be much smaller, and for some cases the runtime is closer to $O(LN)$ than $O(L^2N)$. At an edge, for a given value of one node, the other node may have many fewer potential settings than the full L possibilities. This occurs if the edge prior encodes *constraints*. This is important because L can be quite large. For example, it could represent all points in the T-F representation of an utterance, or all T-F points as well as other flexible parameters such as width, height, or slope. Note that the same situation can occur for some HMMs, for which the Viterbi algorithm also takes time $O(L^2N)$, where L is the number of states and N is the number of frames. For a strictly left-to-right HMM topology with no skip states, instead of considering L transitions for each state at each frame, one only needs to consider exactly two. The state is either kept the same or incremented. In this case, the running time becomes $O(LN)$.

max-product algorithm. If the inference problem is one of marginalization – estimating distributions instead of point estimates – one can use the very similar *sum-product* algorithm. The terms *belief propagation* and *message passing* are also used for this and closely related algorithms [6].

```
Input:
  Global observations, $\mathbf{o}$.
  Tree-structured graph, $G = (V, E)$. $|V| = N$. $|E| = N - 1$.
  Scoring functions, $f_i^A, f_{i,j}^E, f_i^N$ (as in Equation 4.3).
Output:
  Optimal parameter settings, $\mathbf{x}^* = x_1^*, \dots, x_N^*$.
  Total score, $f(\mathbf{x}^*)$.
begin
  Choose a root node, $x_N$.
  Topologically sort from leaves to root, $x_1, \dots, x_N$.

  for $t = 1, \dots, N$ do                                  /* forward pass */
    $\delta_t(i), \psi_t(i) = \text{argmax}_j \left[f_t^A(x_j, \mathbf{o}) + f_{t,i}^E(x_j, x_i) + f_t^N(x_j) + \sum_{k \in c_t} \delta_k(j)\right]$

  $f(\mathbf{x}^*), x_N^* = \text{argmax}_j \left[f_N^A(x_j, \mathbf{o}) + f_N^N(x_j) + \sum_{k \in c_t} \delta_k(j)\right]$

  for $i = N - 1, N - 2, \dots, 1$ do                          /* backtrace */
    $x_t^* = \psi_{t+1}(x_{t+1}^*)$
end
```

Algorithm 4.1: The max-sum algorithm for MAP decoding of tree-structured graphical models. c_t refers to the set of children of node t. $\delta_t(i)$ represents the information passed from node t to its parent. The notation, $a, b = \text{argmax}_x f(x)$, is a combination of the two statements, $b = \text{argmax}_x f(x)$ and $a = f(b)$.

Chapter 5

Examples and applications of PBMs

5.1 Explicit encoding of phonetic information

Chapter 3 presented examples of phonetic cues which are not well modeled with typical HMM/MFCC-based systems. Here, we demonstrate that the proposed parts-based models (PBMs) can explicitly model some of these phenomena, and extracting very simple phonetic features can perform accurate classification – particularly in situations conventional models break down: noisy environments and small training sets.

5.1.1 Formant transitions

Formant transitions are crucial for speech perception, and one of the primary motivations for investigating alternatives to conventional acoustic modeling (see Section 3.2.3). We would like to explore the possibility that *explicit* encoding of formant transitions within a parts-based framework may have advantages over the *implicit* representation in typical features and classifiers. We have previously discussed the formant transition cues for labial stops (e.g., *B*) and how they provide a strong cue for discrimination from easily confused alternatives (such as *E*, See Figure 3-1 on page 30). Here, we examine the performance of *B*-*E* classification based on this formant transition information alone.

Experimentation was done with the simple parts-based model shown in Figure 5-1a, which is used for both *B* and *E*. Parts 3 and 4 represent detectors for the lower edge of the second formant (F2), and these parts are given flexible slopes. Labial stops typically exhibit the rising of formant frequencies coming out of the stop, which for *B* utterances will be most prevalent in F2. To perform classification, the deformable model is fit to an input utterance, and the slope parameters are extracted. These two features (slopes for parts 3 and 4) will be inputs to a simple linear classifier (a regularized least-squares classifier [77]). Several example spectrograms fit with the model are shown in Figure 5-1b. Classification results are summarized in Figure 5-2, showing the error rate as a function of the number of training examples.

Notice that using the F2 slope information alone (just two parameters) is sufficient in most cases to match or exceed classification performance by a full HMM recognizer. Only in the case of clean speech with a large training set does the HMM have a slightly lower error rate. For small training sets, the PBM-based slope measurement performs much better than the baseline. This is not surprising, considering its simplicity: a linear classifier in two dimensions. Only a few examples are needed to effectively set classification thresholds. Compare this to the HMM, which needs over 100 examples to approach its optimal performance. Note that each HMM model has nearly 2000 parameters. Furthermore, in noise, the PBM-based classifier is significantly better in all cases. The slope parameters provide an intermediate representation above the level of acoustics. In noise, the slopes may be more difficult to detect, but when they are correctly identified, their values are independent of any noise corruption.

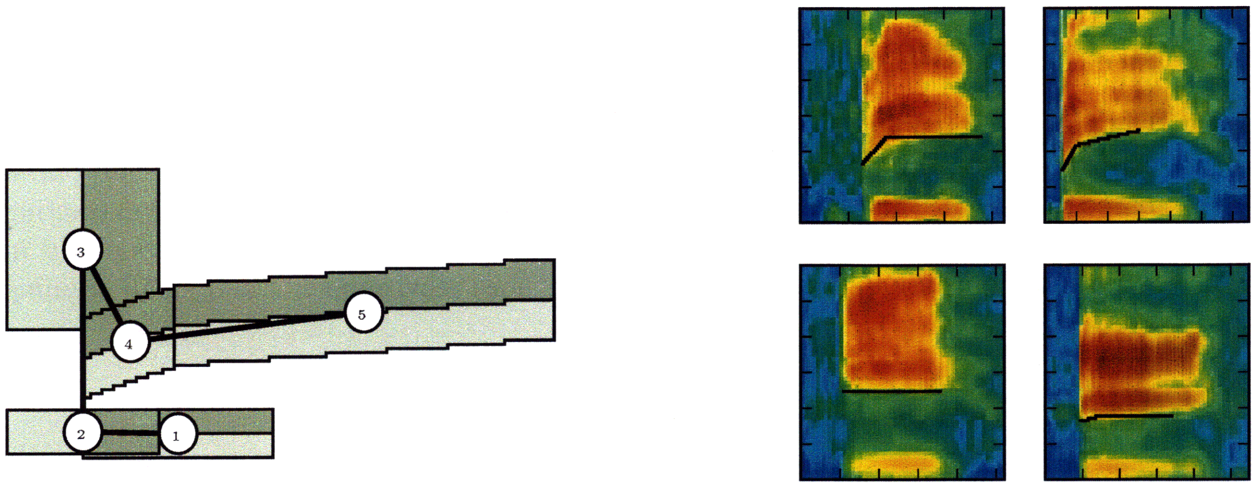


Figure 5-1: (a) The model used for detecting the second formant transition. (b) Some examples of the model fit to data (*B* above, *E* below), with a black line showing the center edge of parts 4 and 5.

Some additional experiments with this F2-slope classifier are given in the section on noise robustness later in this chapter.

5.1.2 Voice-onset time

The voice onset time (VOT) is the amount of time between the burst of a stop consonant and the onset of voicing of the following vowel. VOT has long been known as a perceptual cue for stop consonant identity [57], with unvoiced stops tending to have a longer VOT than voiced stops [87]. VOT measurements are essentially duration measurements and thus not well modeled by HMMs (see Section 3.2.6), but can be modeled explicitly in the proposed framework.

We demonstrate VOT modeling by comparing ISOLET utterances of the letters *D*

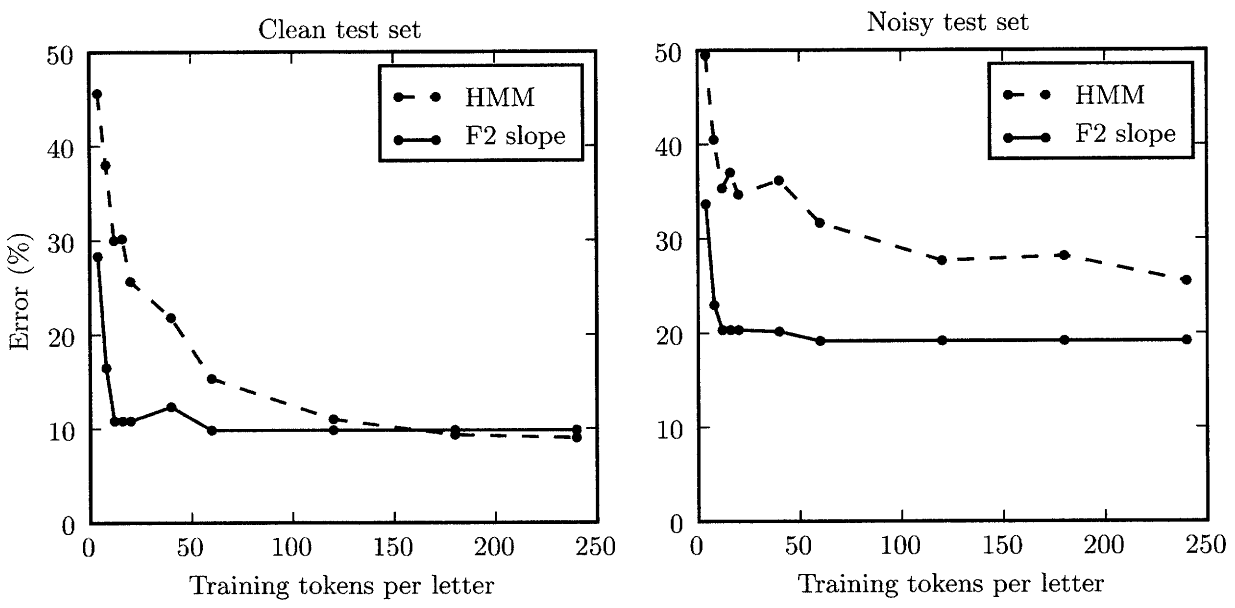


Figure 5-2: ISOLET *B*-*E* classification based on formant slopes derived from a parts-based model (PBM), compared to the MFCC/HMM baseline recognizer.

and *T*. Again, the setup will consist of fitting a deformable parts model to an utterance and extracting the VOT measurement as a feature to use for classification. The PBM used is similar to the model used for *B* and *E* shown in Figure 5-1a. The VOT measurement is extracted as the time difference between a wideband onset detector at middle to upper frequencies (as in part 2 in Figure 5-1a), and a low frequency onset detector meant to detect the onset of F1 (as in part 1 in Figure 5-1a). Note that while the extracted VOT is derived from the parameters of only two parts, all of the parts will have an effect by helping to correctly fit the overall template.

A histogram of extracted VOTs for *D* and *T* are shown in Figure 5-3. This qualitatively matches other *D*-*T* VOT measurements [16]. Figure 5-4 shows classification results using the same setup as the *B*-*E* classifier. Unlike the more difficult *B*-*E* case, the HMMs for *D* and *T* in the clean condition converge rather quickly and the accuracy reaches 99.0%. Classification with just a single VOT parameter achieves an accuracy of 94.3%, and converges almost immediately because "training" consists of choosing a single threshold parameter. In noise, this single VOT measurement consistently outperforms the full HMM recognizer.

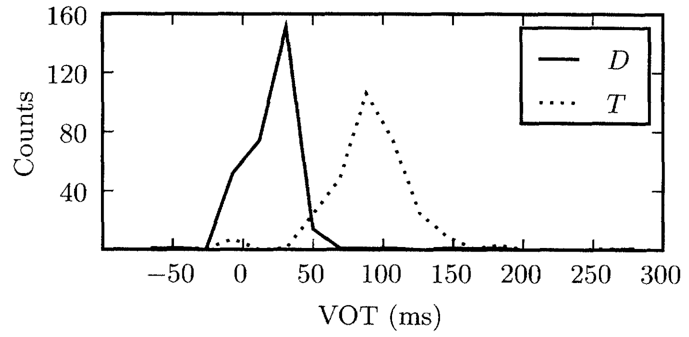


Figure 5-3: Histogram of voice-onset times extracted from the parts-based model for letters *D* and *T*.

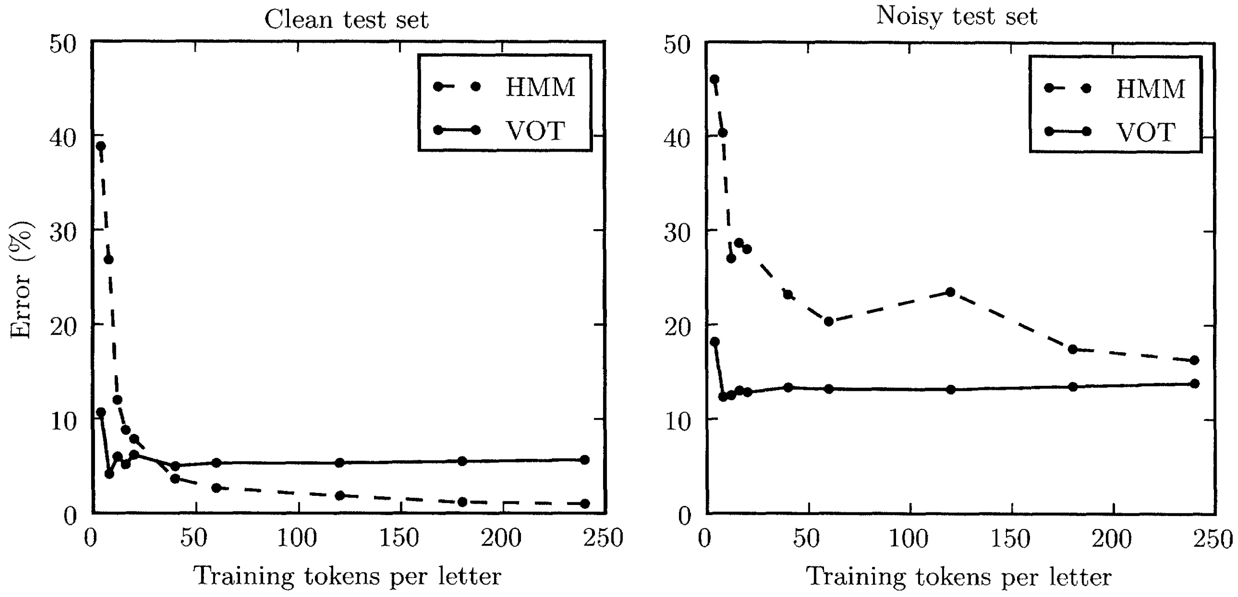


Figure 5-4: Classification of *D* and *T* with a single VOT parameter in comparison to the full HMM baseline recognizer.

5.2 Additional benefits

In addition to explicitly modeling specific phonetic phenomena, there are several other potential benefits to parts-based modeling.

5.2.1 Speaker variability and adaptation

We have previously discussed the problems with phonetic modeling by *fixed* spectral templates (Section 3.2.2). In particular, this approach does not naturally model speaker variation. Differences in vocal tract length result in shifts of the formant frequencies. With the parts-based framework, these speaker differences can be encoded in a more natural way. To demonstrate this, two versions of the *E* model from Section 4.4.1 were trained: one using data from only male speakers, and one from only female speakers. An illustration of the resulting models is shown in Figure 5-5a. As would be expected, there is a considerable difference in the location of the parts which encode the frequency of the second formant. Note that most of the information in the two models is very similar. The primary difference is the F2-F1 distance, and this is *explicitly* represented in the model with a single parameter (the frequency dimension of the mean of the Gaussian modeling the distance between part 1 and part 2). If this one parameter in the male model is given the value the female model, the two models become very similar (Figure 5-5b). Thus, parts-based modeling techniques can more naturally separate speaker-dependent and speaker-independent information within the model.

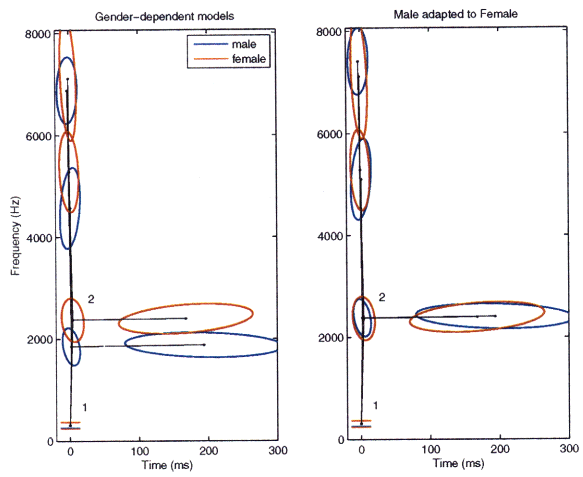


Figure 5-5: On the left is shown the configuration prior distributions for two separately trained models: one on only male speakers, the other on only female speakers. (see Section 4.4.1). On the right, the male model has been modified to closely match the female model by changing *only a single parameter*: $\mu_{2,1}$, the mean frequency distance between points 1 and 2. (Not shown here are the "filter" parameters for the cue detectors, which look subjectively similar in both cases.)

5.2.2 Generative ability

The speech schematic PBMs introduced in Section 4.4.2 essentially define what an example spectrogram should look like. Therefore, drawing samples from the prior can be used for generation. Figure 5-6 shows some examples for the letter Q comparing actual spectrograms with generated spectrograms from both an HMM and a PBM. The problems with the HMM generative model were discussed in Section 3.2.5, and those same issues can be seen here. The PBM in this case consisted of a set of edges and the proper constraints to detect a simple diphone to represent Q: a segment of high-frequency frication for the aspiration of /k/, followed by a vowel a constant F1 and downward sloping F2. To generate a full waveform, frication noise was generated based on the location of the horizontal edge during the first segment, and the voiced region was generated by filtering a set of harmonics based on the edge detectors' parameters during the vowel. While this does not constitute a high-quality speech synthesizer,[1] this simple model captures the basic properties of this diphone and does so with very few parameters. Higher quality synthesis could be performed within this framework.

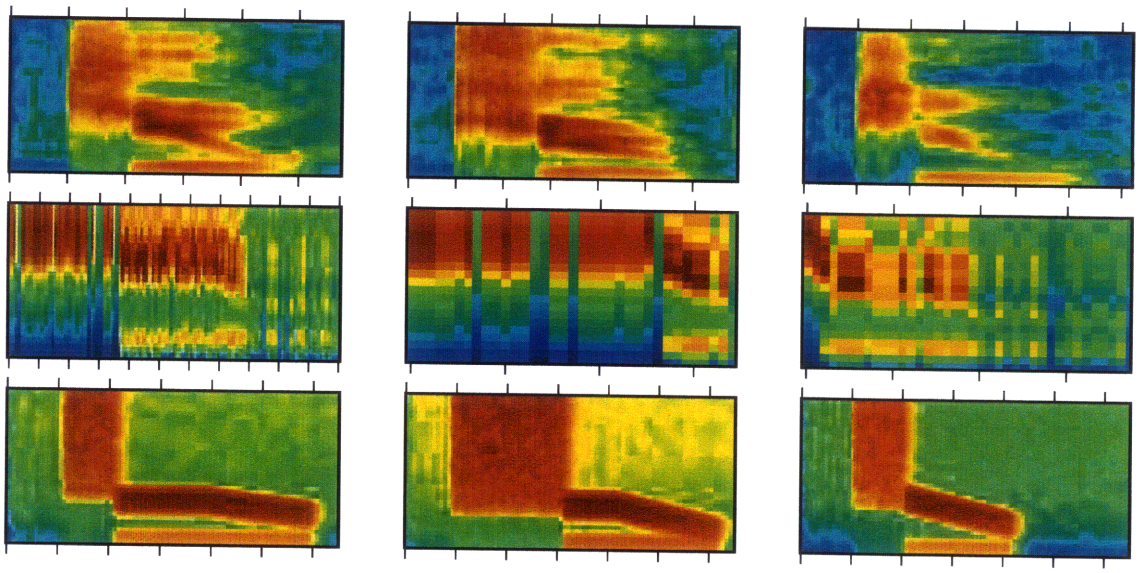


Figure 5-6: *Top row:* Example spectrograms of the letter Q. *Middle row:* Mel-spectra synthesized from an HMM model of Q. *Bottom row:* Spectrograms synthesized from an parts-based model of Q. For all images, ticks are drawn on the horizontal axes every 100ms. The PBM generates spectrograms which are much more similar to typical utterances than those of the HMM. Also, notice the inappropriate time-scale (durations) of those synthesized from an HMM.

5.2.3 Coarticulation (continuity)

Figure 5-7 shows a PBM constructed for the phoneme /ey/ (i.e. the spoken letter A). This vowel is a *diphthong*. Through the course of its duration, there is an inherent changing of it spectrum – the tongue moves higher and more fronted, causing the first formant to fall and the second formant to rise [87]. These movements can be seen in the latter part of the spectrogram in Figure 5-7.

For diphthongs, typical HMM modeling would likely utilize multiple states to model this phenomenon, so that moving through the stages of the diphthong correspond to moving through the states in the HMM. There are several problems with this. First, it uses discrete states of fixed spectral shapes to model a continuously changing spectrum (although GMMs and delta features could help with this to some

[1]In this example, the modeling of an aspirated /k/ with simple high-passed noise is particularly unnatural sounding.

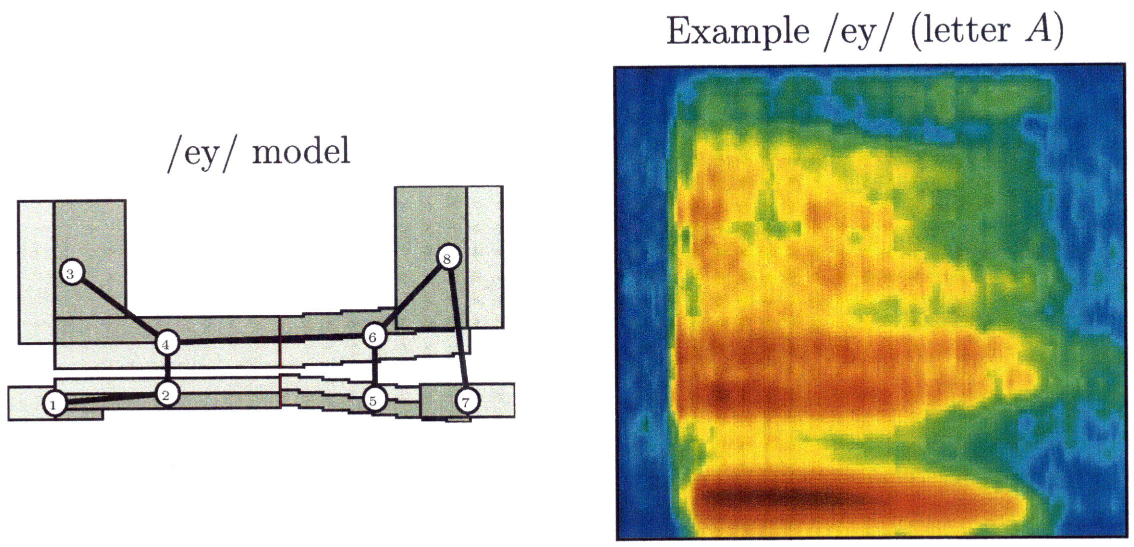


Figure 5-7: A PBM for /ey/ and a spectrogram of an example utterance of the letter *A*. Notice the two segments in the model to represent the diphthong.

extent). Second, even if this phoneme consisted of two stationary segments, there is no mechanism for *continuity between states.*

Now, consider the PBM shown for /ey/. This PBM uses two distinct segments (similar to having two states in an HMM), but notice that the edge between part 4 and part 6 enforces a *continuity* constraint. Parts 4 and 6 detect F2, which will be continuous through the course of the diphthong. And as described in the previous chapter, an "edge prior" can enforce the prior knowledge that the edges of these two parts must align. This is an example of "passing information" in the edges of the graph. In a typical HMM, on the other hand, the edges only dictate what state will be visited next, and not any "parameters" of that state. Building (or learning) models with this ability for continuity constraints could provide a more elegant mechanism for handling coarticulation effects than MFCC/HMM models, which solve these problems primarily with more models and more training data.

5.2.4 Noise Robustness

Several examples earlier in the chapter demonstrated the ability for features extracted from a PBM to robustly perform phonetic classification. Another simple example of a corrupted utterance can illustrate several other possible benefits. Consider the example in Figure 5-8. A simple patch-based model of *B* (the one discussed in Section 4.4.1) is applied to a clean utterance and an utterance artificially corrupted with noise covering narrow ranges of time and frequency. The outlines of the estimated locations of the parts are shown as black rectangles in both cases.

This example demonstrates how the flexibility of the parts' T-F location gives the ability detect parts in those regions which are uncorrupted by noise. In this example, the part at $(t, f) = (260\text{ms}, 2200\text{Hz})$ in the clean utterance is trying to detect the horizontal edge after the formant transition has stabilized. When a short-duration noise burst corrupts its preferred location, it can still detect this feature in an uncorrupted region slightly earlier. A well trained objective function will control this trade-off between searching for a strong detection of a part and finding it in the expected location.

Some parts may have all of their evidence completely corrupted by noise. If this is the case, the acoustic scores for these parts will be low; however, the other parts are unaffected. In the example of Figure 5-8, the two parts below 1000Hz are lost in the noise condition, but the others parts fit match in clean regions and the overall model is correctly fit to the example. This redundancy of cues (having more parts than strictly necessary) can aid in robustness. These properties have an interesting relationship to recent work in missing feature methods and the "speech fragment decoder" [3]. This is discussed in Section 6.1.2.

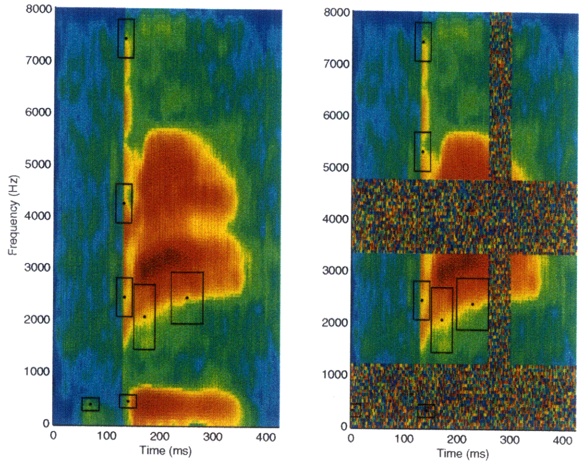


Figure 5-8: (a) An example *B* utterance. (b) The same utterance synthetically corrupted by scrambling portions of the spectrogram. Both are decoded with the *B* model described in Section 4.4.1. The hypothesized locations of its seven parts are shown as black outlines.

We have conducted a classification experiment to illustrate some of these benefits for noise robustness. The *B-E* classifier based on F2-slope (from Section 5.1.1) was tested on data corrupted in limited frequency bands in three different ways,

1. Lowpass filtering at 4500 Hz.

2. Adding a "siren" noise consisting of a slowly (6 Hz) frequency modulated tone ranging between 4500 Hz and 7500 Hz.

3. Highpass filtered at 1000 Hz.

These noises were chosen so that they would not interfere with the F2 cue (which is contained roughly in the range of 1500-2500 Hz). The resulting *B-E* classification error rates are summarized in Figure 5-9. This shows a drastic improvement in using the PBM over the HMM for the noise corrupted cases. Informal listening suggests that like the PBM, these corruptions will have essentially no effect on human classification. While some of these noise types are not conditions which must be dealt with often, finding this kind of data is a useful strategy for exploring alternative ASR models: conditions that cause large problems for a particular ASR system, but do not significantly affect human performance. Creating and working on these kinds of scenarios is a useful way to find better approaches.

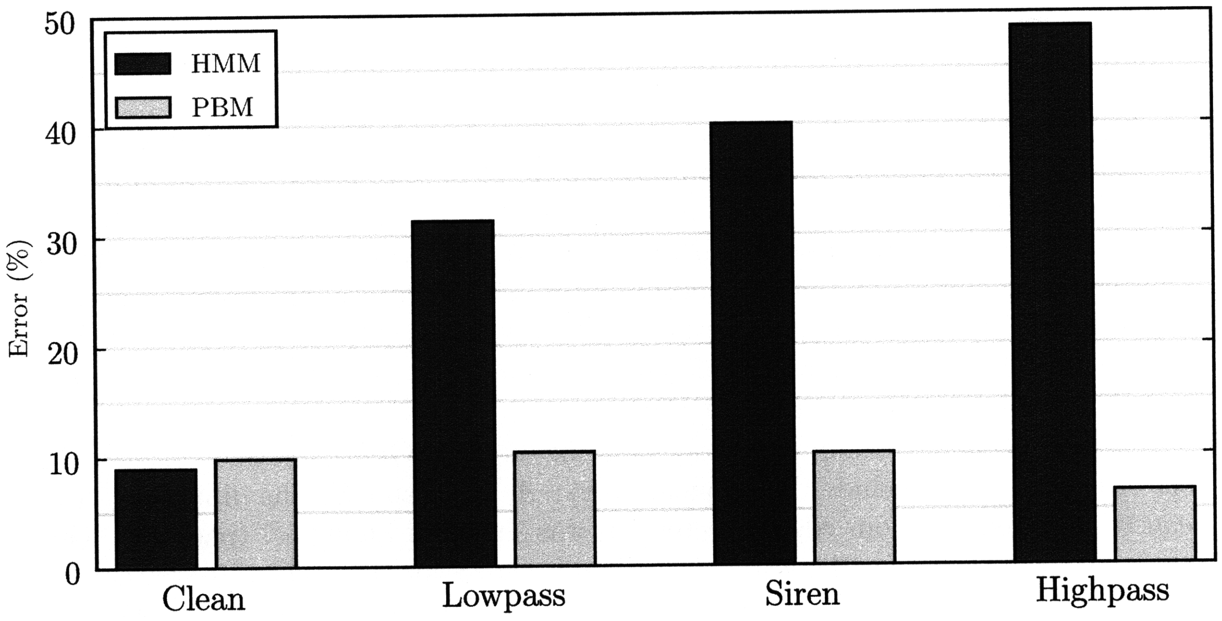


Figure 5-9: *B-E* classification performance in the synthetic noise conditions described in Section 5.2.4. The PBM is based on classification with an extracted F2 slopes. Corruption of uninformative frequency regions drastically degrades performance with a MFCC/HMM system, while the local parts detectors are largely unaffected.

5.3 Connected recognition

Thus far, PBM examples have consisted of fitting a single model to a given utterance – the equivalent of isolated word recognition. The parts-based techniques described in this thesis are primarily motivated for low-level acoustic-phonetic phenomena and building full ASR systems will have to combine these into larger units. This section will discuss several ways in which this is possible.

5.3.1 Direct combination

Two models, $\mathcal{M}_1$ and $\mathcal{M}_2$, can be directly combined into a larger model, $\mathcal{M}$, by simply taking the union of their parts, $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$, and joining the two graphs by adding a single edge between one node from $\mathcal{M}_1$ to one node from $\mathcal{M}_2$. The resulting graph will be a tree and will represent a valid model. Due to the factorization into local terms, the new joint objective function is simply the sum of the two individual scores, plus one extra term representing the newly added edge,

$$
\begin{aligned}
f(\mathbf{x}, \mathbf{o}) &= f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{o}) && (5.1)\\
&= f_1(\mathbf{x}_1, \mathbf{o}) + f_2(\mathbf{x}_2, \mathbf{o}) + f^E_{i,j}(x_i, x_j) && (5.2)
\end{aligned}
$$

where $x_i \in \mathbf{x}_1$ and $x_j \in \mathbf{x}_2$.

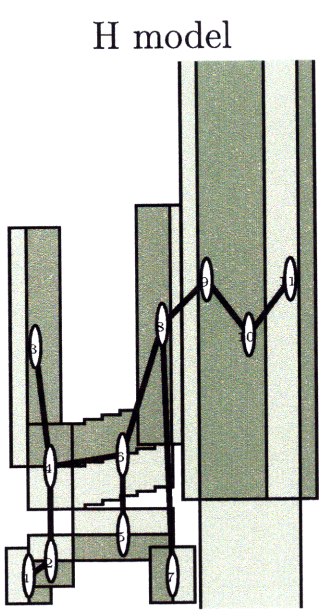
If $f^E_{i,j}$ is chosen to be 0, then the two models are independent and decoding will be equivalent to decoding each separately, which is not very useful. A typical reason one might want to combine models is to create a larger unit from two smaller units, e.g. a diphone from two phones. In this case the edge score, $f^E_{i,j}$, should be chosen to encode the constraint that $\mathcal{M}_2$ occurs after $\mathcal{M}_1$. To do so, the nodes chosen to connect the two graphs must directly encode the end of $\mathcal{M}_1$ and the start of $\mathcal{M}_2$, respectively. Unlike standard HMMs and segment-based models, this edge can also encode other constraints and information.

To give a concrete example of model-sharing on the ISOLET dataset, consider the set of letters which include the phoneme /ey/: *A*, *H*, *J*, and *K*. The diphthong /ey/ can be modeled with four edge detectors, as was shown in Figure 5-7 (page 79). This can serve as a model for the letter *A*. To build models for the other three letters, we can re-use the /ey/ model. For example, an *H* can be viewed as the /ey/ phoneme and a segment of frication (the affricate /ch/) separated by a short period of silence (where the tongue is creating an alveolar stop). Connecting the parts from an /ey/ model and an /ch/ model results in the model shown in Figure 5-10.

For some quantitative results, we performed binary classification between all letter pairs in the /ey/-set, *A*, *H*, *J*, and *K*. Binary linear RLS classifiers were trained on the set of part scores for each pair of models. The results are shown in Table 5.1. In clean speech, the baseline is nearly perfect, and the PBM does not reach this level of accuracy. However, in noise, the parts-based models provide large gains over the baseline in most cases.

5.3.2 Hierarchical parts-based models

Another approach to build connected recognizers with PBMs is to run low-level models to get their scores as a function of time. These output can be combined by some higher-level mechanism, such as a standard HMM or segment-based model, or even used as input to another parts-based model. A "running computation" of a parts-based model can often be done efficiently, and in-fact, computing the score at all points in time often takes no extra computation than full decoding. Recall the de-



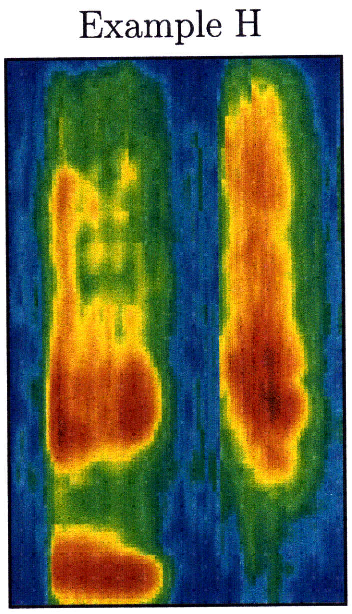


Figure 5-10: On the left is a visualization of a PBM of the letter *H* model built by reusing a common /ey/ model. On the right is a spectrogram of a typical *H*.

coding process described in Section 4.7.3. Messages are passed from the leaves of the tree to the root. At the root node, a maximum is taken over all the root node's parameters as the final step before the backtrace. The maximum value at the root is the total score, and the parameters which produce this maximum will be the hypothesized parameter settings for the root node. If one of the root node's parameters is a variable representing time (e.g. the time location of a T-F patch), then the final maximum can be taken over all parameters *except* time. The resulting array of scores will represent the model's score as a function of time (where the time is measured with respect to the time of the root node). If needed, all the backtrace information can be saved, and a backtrace can be performed from any point in time to reveal the optimal parameter values of all other parts.

	Clean		Noise		
Letter Pair	HMM	PBM	HMM	PBM	Improvement
A-H	1	2	88	26	70.5 %
A-J	1	2	69	40	42.0 %
A-K	1	7	98	121	-23.5 %
H-J	0	1	64	8	87.5 %
H-K	1	1	87	17	80.5 %
J-K	0	15	113	68	39.8 %

Table 5.1: Binary classification of letter pairs containing the phoneme /ey/, with results shown as the number of errors (of 600 total). The improvement column refers the percentage decrease in error rate in the noise condition when using the parts-based model (PBM) instead of the HMM.

Chapter 6

Discussion and future directions

6.1 Relationship to other work

Previous work with similar goals to ours was discussed in Section 3.5. Now that the proposed model has been described in detail, it is useful to revisit some of the previous work to compare and contrast with the details of the models presented in this thesis.

6.1.1 Patch-based models

As described in Section 3.5.2, the work of Ezzat and Poggio [28] described a patch-based word-spotting technique. Both their approach and the PBMs in this thesis are based on models constructed from a deformable template of spectro-temporal patches. There are several important differences. In the PBM, the flexible T-F positions of patches can be correlated, (e.g. the frequency of one patch can be tied to that of its neighbor, via the edge priors), and this correlation can be exploited for discrimination. Also, our work has explored models where the individual patches are not fixed, so that the "invariance" can be extended to dimensions other than time and frequency, such as the slope of an edge detector. Finally, we have considered patches with stronger prior relationships between parts, such as to encode continuity constraints. The patch-based models of [28] can be viewed as a special case of the proposed PBMs, where the priors place uniform distributions over a limited range of T-F, as well as constraints encoding the temporal positioning of the patches. Techniques used in [28] which could be beneficial to our work include learning patches by random selection, and trying to do more "oversampling" of patches. Using a much larger number of overlapping patches may be able to cover more variation than relying on single patches for each phonetic cue.

6.1.2 Missing feature methods

Section 3.5.3 discussed missing feature methods [18] and the speech fragment decoder [3]. In comparison to the speech fragment decoder, the models presented in this thesis also use only selected T-F regions, but in a different way. One way to view the difference between them is to consider the difference between *detection* and *classification*. The PBMs are better thought of as detectors. A model will attempt to detect its component parts, while simply ignoring any other spurious information. Missing feature methods operate more as classifiers. All the incoming data is classified and must either be fit to the hypothesis or assumed corrupt and marginalized out.

While both approaches have advantages, patch-based models are potentially useful to combine with missing feature methods, which require local features and often build models directly on spectral features (i.e., directly on T-F "pixels", for example from T-F representations such as the *ratemap* [63]). Encouraging examples of using PBMs on data with subband corruption (such as the additive siren noise in Section 5.2.4) suggest that they might be useful in approaches which view speech as just one possible sound source in the input, e.g., in computational auditory scene analysis (CASA) [93, 12, 27].

6.2 Future directions

The goals of this thesis were to explore alternative ASR acoustic modeling techniques. Developing these methods into a full recognition system which can compete with finely-tuned HMM-based recognizers will take additional work. There are many possible directions in which future work could proceed.

6.2.1 Learning

The models used in this thesis required a fair amount of work "by-hand", utilizing phonetic knowledge to construct the structure of the models. For reasons of both reducing labor and improving performance, we would like to have more data-driven approaches to building models. Like many problems involving graphical models, data-driven training can be roughly split into two problems:

1. Learning the structure of a model.

2. Training the parameters of a model.

Learning the structure

Learning the structure of models is often a more difficult task than setting the parameters of a given model. However, these PBMs have various levels of structure, and short-term future work will likely consist of hand-written rules for high level structures and only focusing on learning at the lower levels. For example, for the "speech schematic" models, we may define several broad phonetic classes (vowels, fricatives, etc.), the types of patches associated with each (such as horizontal and sloped edges for formants of vowels, rules for where onsets/offsets can occur, etc.), and some simple rules for connecting patches. Then, data-driven approaches could attempt to fit models to some labeled phonetic data.

We have done some previous work on learning the shapes of edge-like T-F detectors for discrimination suitable for parts-based modeling [84]. An Adaboost-based algorithm [80] was used for greedy feature selection, which managed to learn phonetically interpretable T-F patches (Figure 6-1). In [84] the patches could not be shifted in frequency, and the time dimension was handled with a simple warping scheme. In principle, a similar greedy part selection scheme could be utilized within the PBM framework. Additional work could attempt to combine these approaches.

Previous work done by others also gives options for learning the parts and structure of models. In [28] good performance was achieved by simply extracting patches randomly from training examples. Also, there is ongoing work in machine vision to learn parts in parts-based models [88, 78] which could serve as inspiration for methods to learn model structure.

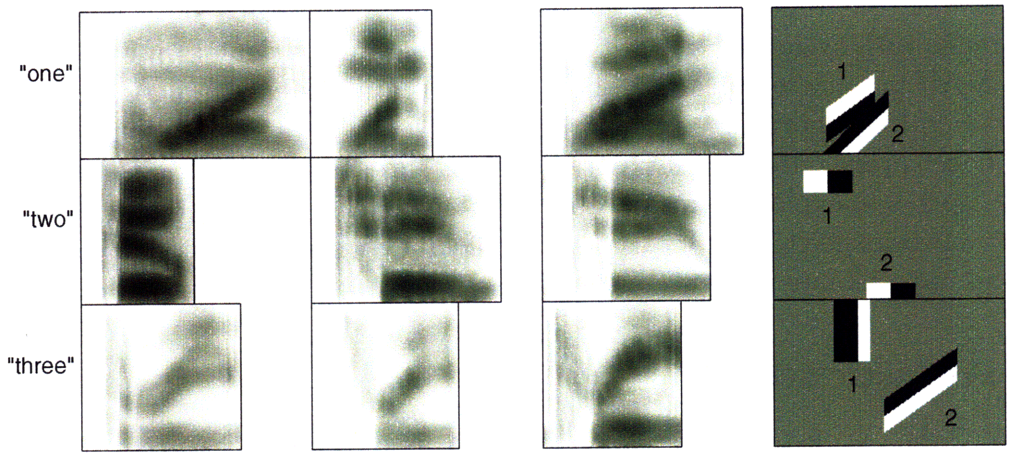


Figure 6-1: A figure from [84] showing the results of learning patches with greedy feature selection. The left three columns show three example spectrograms of the words "one", "two", and "three".The column on the right shows the first two features that the algorithm chooses to distinguish the respective words (white=-1, black=$+1$). Note the recognizable phonetic interpretations: for "one", the upper and lower edges of a rising F2 for $/w\,ah/$; for "two", the onset of high-frequency aspiration for $/t/$, followed by low-frequency voicing onset (note the implicit encoding of voice-onset-time); for "three", high-frequency offset for the end of $/\theta/$, followed by the lower edge of a rising F2/F3 for $/r\,i^y/$.

Training the parameters

Most of the results presented in the previous chapter were obtained by running a fixed PBM, and then extracting the fit parameters to use as features in a classifier. A more principled approach would be to train weights on the components of our objective function (the functions f^A, f^E, f^N), such that the decoded score, $f(\mathbf{x}^*, \mathbf{o})$ from Equation 4.3 (page 47) is used directly for classification.

There are many possibilities for performing this kind of training, including applying the same methods used for HMM parameter tuning. Viterbi training, Baum-Welch training, and various discriminative training methods all have analogues for the case of the tree-structured graphs of PBMs. A promising method would be to apply the very general techniques of gradient-based methods [50], such as energy-based methods on factor graphs discussed in [51].

6.2.2 Models and algorithms

This thesis did not discuss the details of some of the considerations in designing the PBMs used in the previous chapter. However, in many cases the types of models desired (based on acoustic-phonetic knowledge) were limited by the requirement of tree-structured graphs governing the relationships between the parts. Trying to encode additional constraints and relationships between the various parts would often lead to graphs with cycles, for which the max-sum algorithm cannot, in general, produce the MAP parameter settings. There are alternative inference algorithms, such as the junction tree algorithm, and algorithms which find approximate solutions such as loopy belief propagation, variational methods, and sampling methods [6]. There are many aspects to explore in future work, particularly in understanding the various trade-offs between speed, memory usage, and accuracy when using more complicated models and approximate inference algorithms.

6.2.3 Acoustic scoring

The form of the "acoustic scores" for the PBMs used in this thesis were based on simple filters applied to a spectrogram, and the local filters were often limited to simple ± 1 edge detectors. These acoustic score terms (f_i^A terms in Equation 4.3) can be arbitrary scoring functions able to robustly provide "evidence" for a part in the model. Here, we will mention two potential directions for exploring alternative forms for these detectors.

In vocalic regions, we have assumed that the formant locations and trajectories are the important information to capture in the model. For the "speech schematic" models, we used T-F edge detectors to detect the formants. This worked well for some cases, primarily fronted vowels (particularly high-front vowels such as /iy/), for which there is a sizable gap between F1 and F2. In these cases, clear edges are present above F1 and below F2. However, in other vowel contexts where formants are closer together, the edges adjacent to formants are not as easily detected. An alternative approach could focus on *peak detection* rather than *edge detection*. While these are related, tracking the peaks may be more useful than tracking the edges. This also raises the possibility of directly performing explicit formant tracking within the context of PBMs of speech.

Another area which could be helpful is including pitch or periodicity information. All of our features were extracted from a single wideband spectrogram computed for a given utterance. Because the acoustic scores can be arbitrary, there is likely to be a benefit from using "heterogeneous" measurements [37]: different feature detectors can use different T-F representations, or any other information in the signal. In particular, it may be beneficial to explore methods utilizing information on pitch and periodicity. Often certain parts will always be in a voiced region, so the acoustic scores can take this into account. Some cues specifically related to voicing (such as pre-voicing of plosives) could use periodicity measurements directly. This could particularly be useful for feature detection in noise, for example using the autocorrelogram information often used in CASA systems [63, 93].

6.2.4 Other tasks

We used the ISOLET corpus in this thesis because it contains simple phonetic units (mostly diphones), and has easily confusable classes which offer a useful task for exploration. We would like to eventually use PBMs on more complicated and continuous speech data. Section 5.3 discussed two ways to build more complex models. One approach to move forward would be to build up a set of PBMs for diphones or triphones, each of which can output a score as a function of time. Possible data-driven approaches to building and training these models were discussed above in Section 6.2.1. The outputs of a bank of such detectors could be used as input to a more standard speech recognizer. A natural fit would be to use a segment-based recognizer [35], since diphone or triphone PBMs can output scores over potential segments. Ultimately, it would be desirable to use some method of global training of both the lower level and higher level parts in the recognizer (such as in [50]).

While there is always a tendency to move to more complete speech recognizers, we believe there is also a benefit to sticking with simple phonetic tasks. Often the rush to build useful applications neglects solving the problems of lower levels. One dataset we would like to explore is the consonant challenge corpus of VCV syllables [19]. This would have better phonetic coverage than ISOLET. Also, human performance has been evaluated on this task [82]. A useful road-map for future work would be to focus on simple phonetic discrimination on this task until human-level performance is achieved in all conditions.

6.3 Discussion

We conclude with two higher-level comments on future work in acoustic modeling and the models proposed in this thesis.

6.3.1 Phonetic cues as an intermediate representation

Typical ASR approaches try to learn a probabilistic mapping from acoustic observations (features) to phonetic units,

$$\text{acoustics} \longrightarrow \text{phonemes}$$

where "phonemes" represents some kind of abstract phonetic units, such as triphones, and not necessarily actual phoneme models. One way to view some of the benefits of the models presented in this thesis is that they add an intermediate representation,

$$\text{acoustics} \longrightarrow \text{phonetic cues} \longrightarrow \text{phonemes}$$

The "phonetic cues" are represented by the "parts" in a parts-based model. These cues correspond to actual acoustic-phonetic events such as formant transitions, bursts or onsets at particular frequencies, etc.

Not only are these cues localized and potentially robustly detected with specific feature detectors, but they also allow a "buffer" between the acoustics – which are highly variable and often corrupted by noise – and the speech itself. With this intermediate level between acoustics and speech, the training of speech models can concentrate on learning the properties of *speech*, rather than the acoustic realization of speech through a channel. When viewing the models in a generative sense, the traditional approach has to incorporate both a model of speech and a model of how speech appears in the current context,

$$\text{acoustics} \xleftarrow[\text{speech+channel}]{} \text{phonemes}$$

Using an intermediate level of phonetic cues keeps these processes distinct,

$$\text{acoustics} \xleftarrow[\text{channel}]{} \text{phonetic cues} \xleftarrow[\text{speech}]{} \text{phonemes}$$

For example, in the previous chapter, we showed a PBM which could robustly detect the letter B based on the slope of the second formant. Qualitatively, the model encodes the information, "/b iy/ has a rising second formant at the vowel onset". Note that this information is a property of speech itself, and it is true regardless of the channel and noise conditions. Models based on modeling spectral envelope (MFCCs, PLPs, etc), qualitatively encode information such as, "/b iy/ has a spectral shape that looks like *this*". Mixture models or adaptation can expand what *this* refers to, but nevertheless, it is a property of both speech as well as the channel, and thus inherently poorly suited to mismatched channel conditions.

Some work, such as articulatory feature (AF) based methods [44] employ a somewhat similar strategy of intermediate units. However, often the corresponding intermediate representation of AFs are still abstract units and the model must learn a mapping between AFs and full spectral features (such as MFCCs or PLPs). This mapping still confounds the effects of both speech and the environment.

6.3.2 Towards a more effective generative model of speech

We have mentioned several times in this thesis the desire for better *generative* models for ASR. In a broader sense, we believe that future work should aim to construct a more complete *model of speech*. The *source-filter* model of speech [73] has proved very useful for all areas of speech technology. However, this is only a first step. The source-filter decomposition may be useful, but what is the model for the source and what is the model for the filter?

Standard practice in ASR is to try to remove the influence of the source (e.g. by extracting features which are largely pitch-invariant) and model the filter (the spectral shape) as a fixed-rate sequence of spectral profiles (represented by the cepstrum). Much of this thesis has been devoted to arguing that this is a poor model of the "filter", and that the proposed parts-based modeling paradigm may offer a better alternative.

Regardless of the methods used, we believe ASR research could benefit from keeping in mind the broader goal of developing a more effective statistical generative model of speech (including the separation from channel effects discussed in the previous section). Pursuing a more complete model of speech will not only aid ASR systems, but benefit all areas of speech science and technology, including speech synthesis and speech coding.

Appendix A

ISOLET data and baseline recognizer

A.1 Description

The ISOLET corpus [15] is described in its documentation as follows,

> ISOLET is a database of letters of the English alphabet spoken in isolation. The database consists of 7800 spoken letters, two productions of each letter by 150 speakers. It contains approximately 1.25 hours of speech. The recordings were done under quiet, laboratory conditions with a noise-canceling microphone.

We follow the experimental setup proposed by Gelbart [33], which has the following modifications to the original corpus,

- **5-way test sets**

  To obtain more reliable results, all of the recognition results presented are the combination of five separate experiments over five training sets and five test sets. The original ISOLET dataset was separated into five sets (each having 30 speakers unique to that set). Each of the five sub-experiments consists of using one of these five sets as the test set and the other four as the training set.

- **Additive noise**

  Noisy versions of the data were created by adding a variety of noise types (taken from [90]) at a range of signal-to-noise ratios (clean, 20dB, 15dB, 10dB, 5dB, 0dB). A set of scripts is provided by [33] to enable researchers to generate the exact same datasets (using the FaNT tool [40]).

In addition to this setup, for some experiments we are interested performance on limited amounts of training data. The smaller training sets used in Chapter 5 were created by starting with a random list of speakers in the training set, separated by gender. From these lists, pairs of male and female speakers were added one pair at a time (both repetitions for each). This results in a series of test sets of increasing size, each consisting of a multiple of four utterances per letter, and balanced between male and female.

A.2 Baseline recognizer

For a baseline HMM recognizer, we also follow the setup provided by Gelbart [33]. It consists of an HTK recognizer using 39 dimensional MFCC+Δ+$\Delta\Delta$ features, four (strictly left-to-right) states per letter, and an single-state pause model optionally occurring at the utterance start and/or end. It is trained in an iterative fashion by successively splitting Gaussians, ending with six diagonal Gaussians per mixture (in a training procedure similar to the common Aurora baseline recognizer [70]).

A.2.1 Results

The results of the baseline HMM system are given in Table A.1.

		Testing condition	
		clean	noise
Training condition	clean	3.74% (292)	42.03% (3278)
	noise	9.78% (763)	19.92% (1554)

Table A.1: Baseline error rates on the ISOLET classification task (26 classes) in both clean and noise conditions. In parentheses is the number of errors (of 7800 total).

A.2.2 Pairwise results

Since we are interested in difficult phonetic discrimination tasks, all 325 letter pairs $(26 \times 25 \times \frac{1}{2})$ were evaluated to find the 10 binary classification tasks with the highest error rates (using clean training data). The results are shown in Table A.2.

Clean		Noise	
Letter-pair	Error	Letter-pair	Error
B – E	9.00	B – V	36.83
M – N	6.50	P – V	31.50
B – D	4.50	C – Z	29.83
D – V	4.00	B – D	28.17
B – V	3.33	G – T	27.33
D – E	3.00	D – V	26.33
P – T	2.83	B – E	25.50
C – Z	2.67	F – S	25.17
G – T	2.50	F – X	24.67
F – S	2.00	P – T	24.50

Table A.2: Binary classification error rates (%) for the 10 most difficult letter-pairs for the baseline HMM recognizer (trained on clean condition). Each pair has a total of 600 tokens (300 of each letter).

A.3 Spectral representation

The parts-based models used in this thesis are based on detecting patterns in a wideband spectrogram. The spectrograms used were created with the following steps:

1. Pre-emphasize waveform with a factor of 0.97, $x_p[n] = x[n] - 0.97x[n-1]$.
2. Extract 4 millisecond (ms) windows every 2ms. Weight each segment by a Hamming window. (Since ISOLET is at 16kHz, that is 64 sample windows with a 32 sample step size).
3. Take an FFT, with zero padding to 256 points. Keep only the magnitude, and keep only first 129 points.
4. Apply a low-pass filter (in each channel) to remove the vertical striations due to the pitch periods. We use a single-pole filter with time constant of $\tau = 8$ms.
5. Take the logarithm (the base is not important since it will be normalized).
6. Normalize the entire time-frequency image such that it has zero-mean and unit variance.

For reference, Figure A-1 shows the resulting spectrograms for all letters from one speaker.

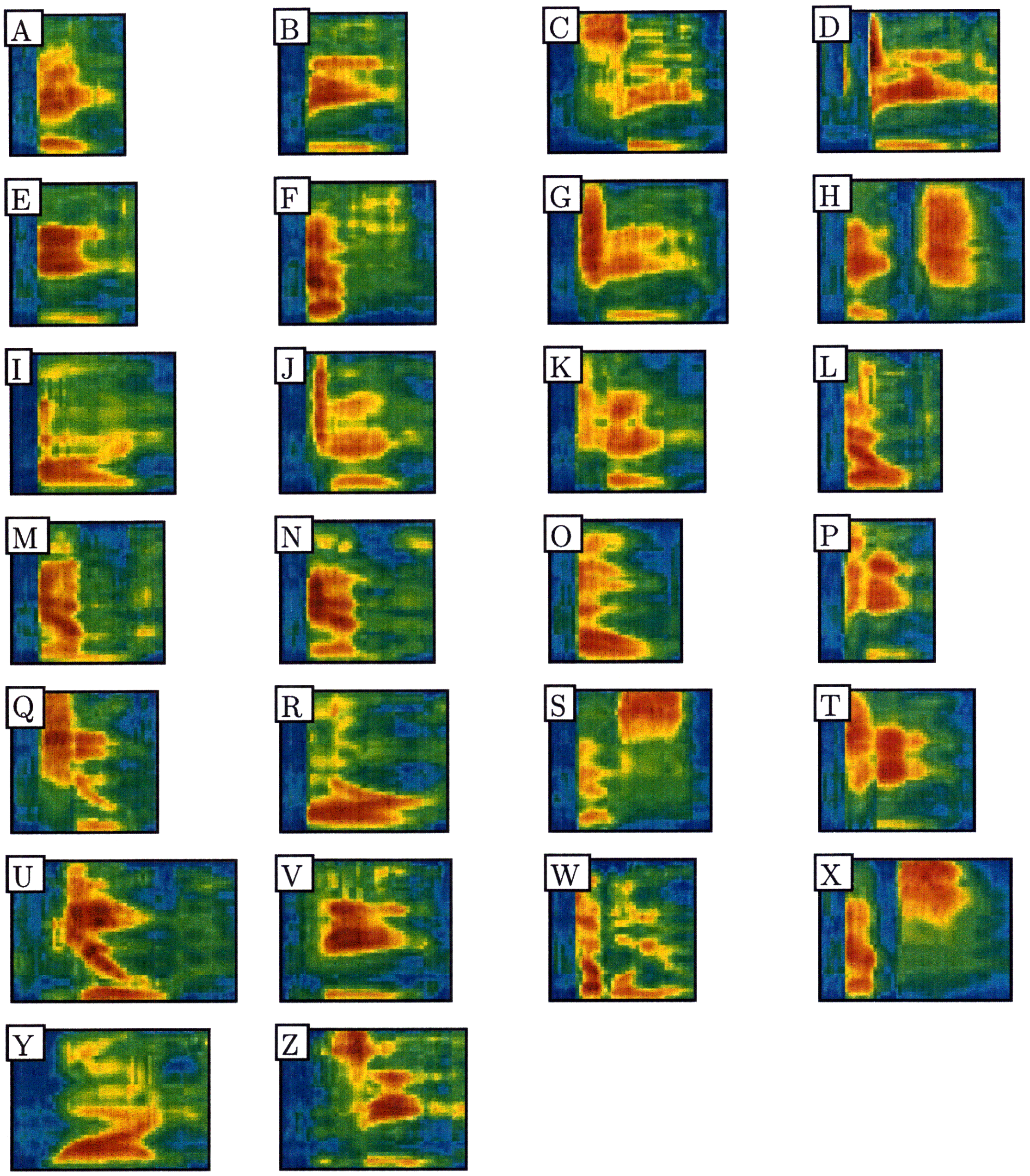


Figure A-1: Spectrograms as used in this thesis, showing all letters from a single ISOLET speaker (speaker fcmc0, token 1 of 2).

Appendix B

Graphical models

This chapter provides background information on the graphical modeling techniques required for the model proposed in Chapter 4. It also provides a description of the notation that will be used in this thesis. For more complete reviews on graphical models see [6, 43].

B.1 Overview

Graphical modeling offers a principled framework for dealing with multiple random variables by encoding the relationships between variables into a *graph*. A set of random variables $\mathbf{x} = \{x_1, ..., x_N\}$ is modeled with a graph, $G = (V, E)$, by associating each node of the graph with a variable, x_i, and each edge with a pair of variables, (x_i, x_j). The structure of the graph encodes relationships between the variables which allow the joint probability distribution, $p(\mathbf{x})$, to be factored into a product of simpler functions over only a subset of the variables,

For *undirected graphical models* (sometimes referred to as Markov random fields, or MRFs), the edges have no direction, and the factorization of the joint distribution is given by,

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{C} \psi_C(x_C) \tag{B.1}$$

C represents the *maximal cliques* of the graph. A clique is a subset of the nodes for which every pair of nodes in the subset is connected with an edge. A clique is called *maximal* if no other nodes can be added such that it remains a clique. x_C is set of variables represented by the nodes contained in clique C, and the *potential functions*, $\psi_C(x_C)$, are non-negative functions of x_C. Z is called the *partition function* and provides a normalization factor ensuring that $p(\mathbf{x})$ is a valid distribution, i.e., $Z = \sum_{\mathbf{x}} \prod_C \psi_C(x_C)$.[1]

[1]We will deal exclusively with x_i being discrete random random variables, and thus this is a summation over $\mathbf{x}$, rather than an integral.

In a *directed graphical model* (sometimes referred to as a Bayesian Network, or BN), the edges have a direction, and the graph has no directed cycles (i.e., it is a directed acyclic graph, or DAG). If the edge $x_i \to x_j$, is in the graph, then x_i is a *parent* of x_j and x_j is a *child* of x_j. For directed graphical models, the local functions (the factors of $p(\mathbf{x})$) are the conditional distributions of each node given that node's parents,

$$p(\mathbf{x}) = \prod_{k=1}^{N} p(x_k | x_{\pi(k)}) \tag{B.2}$$

where $x_{\pi(k)}$ represents the set of parents of x_k.

Figure B-1 shows an example for both undirected and directed graphs. For these graphs, the corresponding factorization of the distribution is given by,

$$\text{(a)} \qquad p(\mathbf{x}) = \frac{1}{Z}\psi_{1,2}(x_1, x_2)\psi_{2,3}(x_2, x_3)\psi_{3,5}(x_3, x_5)\psi_{3,4}(x_3, x_4) \tag{B.3}$$

$$\text{(b)} \qquad p(\mathbf{x}) = p(x_1|x_2)p(x_2|x_3)p(x_3|x_5)p(x_4|x_3)p(x_5) \tag{B.4}$$

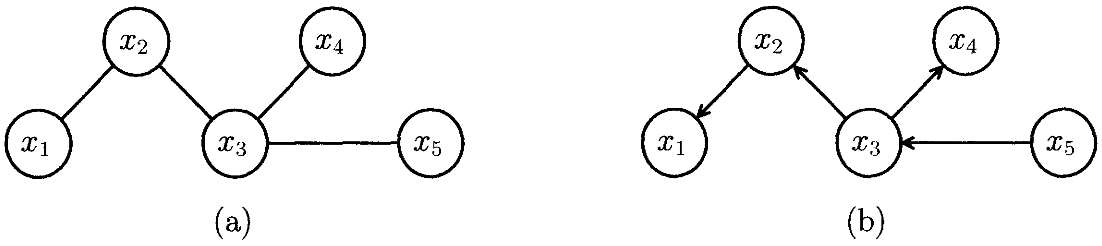


Figure B-1: (a) An undirected graph. More specifically, an undirected tree. (b) A directed tree determined from (a) by selecting x_5 as the root node.

B.2 Tree-structured graphs

For an undirected graph, a *tree* is a graph with exactly one unique path between any two nodes. Equivalently, a tree is a connected graph with no loops. From this definition, the graph must have maximal cliques of size two, and thus the joint distribution is of the form,

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{i,j \in E} \psi_{i,j}(x_i, x_j) \tag{B.5}$$

where the potential functions deal only with interactions of pairs of nodes.

For trees, both the directed and undirected forms can encode the same set of distributions (which is not the case in general) [6]. In the examples of Figure B-1, notice that the terms in their factorizations (Equations B.3 and B.4) can be directly equated. Each $\psi_{i,j}$ is equal to one of the conditional distributions. The extra term in Equation B.4 for the root node, x_5, can be incorporated into its pair-wise potential function, $\psi_{3,5}(x_3, x_5) = p(x_3|x_5)p(x_5)$. Because all potential functions are valid probability distributions, it is already properly normalized, thus $Z = 1$.

The algorithms used in this thesis require the underlying graph to be a tree, thus all models and examples will use tree-structured graphs. Either directed or undirected graphs can be used, but because the local functions in the undirected case can be unnormalized, we will generally deal with undirected graphs.

B.3 Observed nodes

In practice, often the set of nodes can be partitioned into two groups: *hidden* nodes and *observed* nodes. For a given problem, the values of the observed nodes are known, while the values of the hidden nodes are unknown. Following [96], it is useful to consider the case where every hidden node, x_i, has a single corresponding observed node, o_i. The observed node, o_i, provides the evidence for x_i. For example, o_i can represent a noisy measurement of x_i. In this case, there is an edge (x_i, o_i) connecting each hidden node to its corresponding observed node, and this is the only edge connected to each o_i.[2] Note that if the original graph for the hidden nodes is a tree, then adding these observed nodes will still result a tree. In this case, the MRF joint distribution, $p(\mathbf{x}, \mathbf{o})$, is given by,

$$\begin{aligned} p(\mathbf{x}, \mathbf{o}) &= p(x_1, \ldots, x_N, o_1, \ldots, o_N) && \text{(B.6)} \\ &= \frac{1}{Z} \prod_{i,j \in E} \psi_{i,j}(x_i, x_j) \prod_{i \in V} \psi_i(x_i, o_i) && \text{(B.7)} \end{aligned}$$

Because all observed nodes are available, it can be helpful to just view the final term in Equation B.7 as simply $\psi_i(x_i)$, which represents the "evidence" for x_i [96]. For the models used in this thesis, the set of observations will not correspond to direct pairing with the nodes, but considered as a global set of information available to all variables. Thus, the "evidence" terms will be written $\psi_i(x_i, \mathbf{o})$. An instance is conditioned on globally observed information, from which arbitrary "features" can be computed.

[2] From the relationships between conditional independence and graph properties [6], the fact that x_i separates o_i from the rest of the graph, tells us that $p(o_i, \mathbf{x}_{\setminus i}|x_i) = p(o_i|x_i)p(\mathbf{x}_{\setminus i}|x_i)$, where $\mathbf{x}_{\setminus i}$ represents the set of $\mathbf{x}$ not including x_i.

B.4 Inference

For the case of observed and hidden variables as described above, the inference problem we are interested in this thesis is finding the setting of variables resulting the maximum a posteriori (MAP) probability,

$$\mathbf{x}^* = \operatorname*{argmax}_{\mathbf{x}} p(\mathbf{x}|\mathbf{o}) \tag{B.8}$$

Note that $\mathbf{x}^*$ also maximizes $p(\mathbf{x}, \mathbf{o})$, for a given $\mathbf{o}$. Using the factorization from Equation B.7, and performing the maximization in the log domain [6],

$$\begin{aligned}
\mathbf{x}^* &= \operatorname*{argmax}_{\mathbf{x}} p(\mathbf{x}, \mathbf{o}) && \text{(B.9)}\\
&= \operatorname*{argmax}_{\mathbf{x}} \log p(\mathbf{x}, \mathbf{o}) && \text{(B.10)}\\
&= \operatorname*{argmax}_{\mathbf{x}} \left[\sum_{i,j\in E} \log \psi_{i,j}(x_i, x_j) + \sum_{i\in V} \log \psi_i(x_i, \mathbf{o}) \right] && \text{(B.11)}
\end{aligned}$$

Notice that for finding the optimal $\mathbf{x}$, the partition function, Z, is not needed and thus was dropped. We will use $f(\cdot)$ to represent an overall objective function, and $f(\cdot)$ with subscripts to represent the log of the potential functions, so that the function we wish to maximize is,

$$f(\mathbf{x}, \mathbf{o}) = \sum_{i,j\in E} f_{i,j}(x_i, x_j) + \sum_{i\in V} f_i(x_i, \mathbf{o}) \tag{B.12}$$

In the case that the local functions are properly normalized, then the $f(\cdot)$ functions represent log-likelihoods, which are typical objective functions in HMM-based ASR methods (see Section 2.2.3).[3] Equation B.12 is the form of the objective function used for the parts-based models described in Chapter 4. The *max-sum* algorithm is a dynamic programming algorithm able to efficiently solve for the $\mathbf{x}$ which maximizes $f(\mathbf{x}, \mathbf{o})$. This algorithm is described in Section 4.7.3.

[3]Some use the negative of the log potential functions, which is known as as the *energy* [51], due to its history in statistical physics. Using our notation allows us to think in terms of maximizing log-likelihoods (as opposed to minimizing energy).

Bibliography

[1] J. B. Allen. How do humans process and recognize speech? *IEEE Trans. on Speech and Audio Proc.*, 2(4):567–577, Oct 1994.

[2] Y. Amit, A. Koloydenko, and P. Niyogi. Robust acoustic object detection. *Journal of the Acoustical Society of America*, 118:2634–2648, 2005.

[3] J. Barker, M. P. Cooke, and D. P. Ellis. Decoding speech in the presence of other sources. *Speech Communication*, 45:5–25, 2005.

[4] S. M. Bileschi and T. Poggio. Component-based face detection. In *In Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 657–662, 2001.

[5] J. A. Bilmes. Graphical models and automatic speech recognition. In *Mathematical Foundations of Speech and Language Processing.* Springer-Verlag, 2003.

[6] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[7] H. Bourlard and S. Dupont. A new ASR approach based on independent processing and recombination of partial frequency bands. In *International Conference on Spoken Language Processing*, pages 426–429, 1996.

[8] H. Bourlard, H. Hermansky, and N. Morgan. Towards increasing speech recognition error rates. *Speech Communication*, 18:205–231, 1996.

[9] H. Bourlard and N. Morgan. *Connectionist Speech Recognition - A Hybrid Approach.* Kluwer Academic Publishers, 1994.

[10] J. Bouvrie, T. Ezzat, and T. Poggio. Localized spectro-temporal cepstral analysis of speech. In *International Conference on Acoustics, Speech, and Signal Processing*, Las Vegas, Nevada, 2008.

[11] A. S. Bregman. *Auditory Scene Analysis.* The MIT Press, Cambridge, MA, 1990.

[12] G. J. Brown and M. P. Cooke. Computational auditory scene analysis. *Computer Speech and Language*, 8:297–336, 1994.

[13] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318, Morristown, NJ, USA, 1996. Association for Computational Linguistics.

[14] T. Chi, P. Ru, and S. Shamma. Multiresolution spectrotemporal analysis of complex sounds. *Journal of the Acoustical Society of America*, 118:887–906, 2005.

[15] R. A. Cole, Y. Muthusamy, and M. Fanty. The ISOLET spoken letter database. Technical report, 1990.

[16] R. A. Cole, R. M. Stern, and M. J. Lasry. *Performing fine phonetic distinctions: templates versus features.* Lawrence Erlbaum Assocs., Hillsdale, N.J., 1986.

[17] M. P. Cooke. A glimpsing model of speech perception in noise. *Journal of the Acoustical Society of America*, 119:1562–1573, 2006.

[18] M. P. Cooke, P. Green, L. Josifovski, and A. Vizinho. Robust automatic speech recognition with missing and unreliable data. *Speech Communication*, 34:267–285, 2001.

[19] M. P. Cooke and O. Scharenborg. The interspeech 2008 consonant challenge. In *Interspeech*, pages 1765–1768, Brisbane, Australia, September 2008.

[20] F. S. Cooper, P. Delattre, A. M. Liberman, J. M. Borst, and L. J. Gerstman. Some experiments on the perception of synthetic speech sounds. *Journal of the Acoustical Society of America*, 24:597–606, 1952.

[21] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition.* The MIT Press, September 2001.

[22] C. Cortes and V. Vapnik. Support vector networks. In *Machine Learning*, volume 20, pages 273–297, 1995.

[23] K. Davis, B. Biddulph, and S. Balashek. Automatic recognition of spoken digits. *Journal of the Acoustical Society of America*, 24:637–642, 1952.

[24] S. Davis and P. Mermelstein. Comparison of parametric representation for monosyllable word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28:357–366, Aug 1980.

[25] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[26] D. A. Depireux, J. Z. Simon, D. J. Klein, and S. A. Shamma. Spectro-temporal response field characterization with dynamic ripples in ferret primary auditory cortex. *Journal of Neurophysiology*, 85:1220–1234, March 2001.

[27] D. P. Ellis. *Prediction-driven computational auditory scene analysis.* PhD thesis, Massachusetts Institute of Technology, 1996.

[28] T. Ezzat and T. Poggio. Discriminative word-spotting using ordered spectro-temporal patch features. In *Proceedings of the 2008 SAPA Workshop*, pages 35–40, Brisbane, Australia, September 2008.

[29] L. Fei-Fei, R. Fergus, and A. Torralba. Recognizing and learning object categories. http://people.csail.mit.edu/torralba/shortCourseRLOC.

[30] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1), January 2005.

[31] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92, January 1973.

[32] J. Frankel, M. Wester, and S. King. Articulatory feature recognition using dynamic bayesian networks. *Comput. Speech Lang.*, 21(4):620–640, 2007.

[33] D. Gelbart. Noisy isolet and isolet testbeds. http://www.icsi.berkeley.edu/speech/papers/eurospeech05-onset/isolet.

[34] Z. Ghahramani, M. I. Jordan, and P. Smyth. Factorial hidden markov models. In *Machine Learning.* MIT Press, 1997.

[35] J. R. Glass. A probabilistic framework for segment-based speech recognition. *Computer Speech and Language*, 17:137–152, 2003.

[36] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt. Hidden conditional random fields for phone classification. In *Eurospeech*, 2005.

[37] A. Halberstadt. *Heterogeneous Acoustic Measurements and Multiple Classifiers for Speech Recognition.* PhD thesis, Massachusetts Institute of Technology, 1998.

[38] H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. 87(4):1738–1752, April 1990.

[39] H. Hermansky and S. Sharma. Temporal patterns (TRAPs) in ASR of noisy speech. In *International Conference on Acoustics, Speech, and Signal Processing*, 1997.

[40] H.-G. Hirsch. FaNT: Filtering and noise adding tool. http://dnt.kr.hs-niederrhein.de/download.html.

[41] X. Huang, A. Acero, and H.-W. Hon. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development.* Prentice Hall PTR, April 2001.

[42] T. Irino and R. D. Patterson. A dynamic compressive gammachrip auditory filterbank. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(6), November 2006.

[43] M. Jordan and Y. Weiss. *Graphical models: probabilistic inference.* MIT Press, 2002.

[44] S. Kinga, J. Frankel, K. Livescu, E. McDermott, K. Richmond, and M. Wester. Speech production knowledge in automatic speech recognition. *Journal of the Acoustical Society of America*, 121:723–742, 2007.

[45] K. Kirchhoff. *Robust Speech Recognition Using Articulatory Information.* PhD thesis, University of Bielefeld, Bielefeld, Germany, 1999.

[46] M. Kleinschmidt. Localized spectro-temporal features for automatic speech recognition. In *Eurospeech*, 2003.

[47] M. Kleinschmidt and D. Gelbart. Improving word accuracy with gabor feature extraction. In *International Conference on Spoken Language Processing*, 2002.

[48] F. R. Kschischang, B. J. Frey, and H. andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519, 2001.

[49] L. F. Lamel. *Formalizing knowledge used in spectrogram reading: acoustic and perceptual evidence from stops.* PhD thesis, Massachusetts Institute of Technology, 1988.

[50] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[51] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. J. Huang. A tutorial on energy-based learning. In *Predicting Structured Data.* MIT Press, 2006.

[52] L. Lee and R. C. Rose. Speaker normalization using efficient frequency warping procedures. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 353–356, 1996.

[53] C. J. Leggetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech and Language*, 9:171–185, April 1995.

[54] H. C. Leung and V. W. Zue. Visual characterization of speech spectrograms. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 2751–2754, Tokyo, 1986.

[55] A. M. Liberman. *Speech: A special code.* MIT Press, Cambridge, MA, 1996.

[56] A. M. Liberman, P. Delattre, and F. S. Cooper. The role of selected stimulus-variables in the perception of the unvoiced stop consonants. *The American Journal of Psychology*, 65(4):497–516, October 1952.

[57] A. M. Liberman, P. C. Delattre, and F. S. Cooper. Some cues for the distinction between voiced and voiceless stops in initial position. *Language and Speech*, 1:153–167, 1958.

[58] A. M. Liberman, F. Ingemann, L. Lisker, P. Delattre, and F. S. Cooper. Minimal rules for synthesizing speech. *Journal of the Acoustical Society of America*, 31:1490–1499, 1959.

[59] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *IEEE ICIP 2002*, pages 900–903, 2002.

[60] R. P. Lippmann. Speech recognition by machines and humans. *Speech Communication*, 22:1–15, 1997.

[61] K. Livescu. *Feature-Based Pronunciation Modeling for Automatic Speech Recognition*. PhD thesis, Massachusetts Institute of Technology, 2005.

[62] K. Livescu, Özgür Çetin, M. Hasegawa-Johnson, S. King, C. Bartels, N. Borges, A. Kantor, P. Lal, L. Yung, A. Bezman, S. Dawson-Haggerty, B. Woods, J. Frankel, M. Magimai-Doss, and K. Saenko. Articulatory feature-based methods for acoustic and audio-visual speech recognition: Summary from the 2006 jhu summer workshop. In *International Conference on Acoustics, Speech, and Signal Processing*, April 2007.

[63] N. Ma, P. Green, J. Barker, and A. Coy. Exploiting correlogram structure for robust speech recognition with multiple speech sources. *Speech Communication*, 49:874–891, 2007.

[64] J. McAuley, J. Ming, D. Stewart, and P. Hanna. Subband correlation and robust speech recognition. *IEEE Transactions on Speech and Audio Processing*, 13(5):956–964, September 2005.

[65] E. McDermott, T. Hazen, J. L. Roux, A. Nakamura, and S. Katagiri. Discriminative training for large vocabulary speech recognition using minimum classification error. *IEEE Transactions on Audio, Speech and Language Processing*, 15(1), January 2007.

[66] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:349–361, 2001.

[67] N. Morgan, Q. Zhu, A. Stolcke, K. Sonmez, S. Sivadas, T. Shinozaki, M. Ostendorf, P. Jain, H. Hermansky, D. P. Ellis, G. Doddington, B. Chen, O. Cretin, H. Bourlard, and M. Athineos. Pushing the envelope – aside. *Signal Processing Magazine, IEEE*, 22(5):81–88, 2005.

[68] M. Ostendorf. Moving beyond the "beads-on-a-string" model of speech. In *IEEE Automatic Speech Recognition and Understanding Workshop*, pages 79–83, Keystone, CO, USA, 1999.

[69] M. Ostendorf, V. Digalakis, and O. A. Kimball. From hmms to segment models: A unified view of stochastic modeling for speech recognition. 4:360–378, 1995.

[70] D. Pearce and H.-G. Hirsch. The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. 2000.

[71] D. Povey. *Discriminative Training for Large Vocabulary Speech Recognition.* PhD thesis, Cambridge University, 2003.

[72] L. Rabiner and B.-H. Juang. *Fundamentals of speech recognition.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[73] L. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals.* Prentice-Hall, 1978.

[74] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[75] B. R. Ramakrishnan. *Reconstruction of Incomplete Spectrograms for Robust Speech Recognition.* PhD thesis, Carnegie Mellon University, 2000.

[76] M. Reyes-Gomez, N. Jojic, and D. P. Ellis. Towards single-channel unsupervised source separation of speech mixtures: The layered harmonics/formants separation-tracking model. In *Proceedings of the 2004 SAPA Workshop*, Jeju, Korea, October 2004.

[77] R. M. Rifkin and R. A. Lippert. Notes on regularized least squares. Technical report, Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory. 2007.

[78] D. A. Ross and R. S. Zemel. Learning parts-based representations of data. *Journal of Machine Learning Research*, 7:2369–2397, 2006.

[79] L. K. Saul and M. I. Jordan. Boltzmann chains and hidden markov models. In *Advances in Neural Information Processing Systems 7*, pages 435–442. MIT Press, 1995.

[80] R. E. Schapire. *The Boosting Approach to Machine Learning – An Overview.* Springer, 2003.

[81] O. Scharenborg. Reaching over the gap: A review of efforts to link human and automatic speech recognition research. *Speech Communication*, 49:336–347, 2007.

[82] O. Scharenborg and M. P. Cooke. Comparing human and machine recognition performance on a vcv corpus. In *Proceedings of the workshop on Speech Analysis and Processing for Knowledge Discovery*, Aalborg, Denmark, June 2008.

[83] O. Scharenborg, V. Wan, and R. K. Moore. Towards capturing fine phonetic variation in speech using articulatory features. *Speech Communication*, 49(10-11):811–826, 2007.

[84] K. Schutte and J. R. Glass. Speech recognition with localized time-frequency pattern detectors. In *IEEE Automatic Speech Recognition and Understanding Workshop*, pages 341–346, Kyoto, Japan, December 2007.

[85] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[86] S. Srinivasan, N. Roman, and D. Wang. On binary and ratio time-frequency masks for robust speech recognition. In *International Conference on Spoken Language Processing*, pages 2541–2544, 2004.

[87] K. N. Stevens. *Acoustic Phonetics.* The MIT Press, Cambridge, MA, 1998.

[88] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S.Willsky. Describing visual scenes using transformed objects and parts. *International Journal of Computer Vision*, 77, March 2008.

[89] L. Tóth and A. Kocsor. Explicit duration modelling in HMM/ANN hybrids. *LNAI*, pages 310–317, 2005.

[90] A. Varga, H. J. M. Steeneken, M. Tomlinson, and D. Jones. The noisex-92 study on the effect of additive noise on automatic speech recognition. Technical Report, DRA Speech Research Unit, 1992.

[91] P. Viola and M. Jones. Robust real-time object detection. In *Workshop on Statistical and Computational Theories of Vision*, Vancouver, Canada, July 2001.

[92] D. Wang. *On ideal binary mask as the computational goal of auditory scene analysis*, pages 181–197. Kluwer Academic, Norwell MA, 2004.

[93] D. Wang and G. J. Brown. *Computational Auditory Scene Analysis: Principle, Algorithms and Applications.* IEEE Press, Wiley-Interscience, 2006.

[94] K. Wang and S. Shamma. Spectral shape analysis in the central auditory system. *IEEE Transactions on Speech and Audio Processing*, 3, 1995.

[95] L. Welling, S. Kanthak, and H. Ney. Improved methods for vocal tract normalization. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 761–764, 1999.

[96] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. Technical report, San Francisco, CA, USA, 2002.

[97] W. A. Yost. *Fundamentals of Hearing: An Introduction.* Elsevier, 5th edition, 2007.

[98] S. Young, G. Evermann, M. Gales, T. Hain, D. Kerhsaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book.* Cambridge University Engineering Department, 2002.

[99] V. W. Zue. Notes on spectrogram reading. Technical report, Massachusetts Institute of Technology, Dept of EECS. 1985.

[100] V. W. Zue. The use of speech knowledge in automatic speech recognition. In *Proceedings of the IEEE*, volume 73, pages 1602–1615, Nov 1985.

[101] G. G. Zweig. *Speech Recognition with dynamic bayesian networks.* PhD thesis, University of California, Berkeley, 1998.