RECOGNITION OF PROSODIC FACTORS AND DETECTION OF LANDMARKS FOR IMPROVEMENTS TO CONTINUOUS SPEECH RECOGNITION SYSTEMS

BY

Sarah Borys and Mark Hasegawa-Johnson

UNDERGRADUATE THESIS

Submitted in partial fulfillment of the requirements
for the degree of Bachelor of Science in Electrical Engineering
in the College of the
University of Illinois at Urbana-Champaign, 2004

Urbana, Illinois

This thesis examines the usefulness of including prosodic and phonetic context information in the phoneme model of a speech recognizer. This is done creating a series of prosodic and phonetic models and then comparing the log likelihoods of each model. The comparison of log likelihoods shows that both prosodic and phonetic context information improve the phoneme model for most phonemes. The prosodic and phonetic context information is then modeled in two separate recognizers to show that while phonetic context modeling does provide some useful information for the recognizer, prosody provides much more. This thesis also experiments with the detection of landmarks, the transitions between two phonemes, using binary classification. The experiments presented show that landmarks can be detected with high accuracy and are useful for phone classification.

Table of Contents

Chapter 1

Introduction

Humans can interpret and understand spoken language with relative ease. Computers, however, struggle with the task of recognizing continuous speech. Part of the problem is that speech recognizers implemented on computers are not robust to environmental and acoustic variations. For example, a speech recognizer trained under quiet conditions fails miserably when any amount of background noise is added. A speech recognizer trained on a specific person's voice will not perform as well for a different talker. Even the choice of microphone affects the performance of a recognizer. The accuracy of a recognizer trained on speech recorded with one microphone will decrease if the original microphone is replaced with a different one. Another problem is that speech itself is variable. No word or sound is produced in exactly the same way twice.

Another factor that plays into the poor performance of speech recognizers is that natural human conversational speech contains thousands of words; many of which have similar pronunciations. When a speech recognizer is trained and tested on a database containing only a few isolated, discrete words, recognition rates are high. Liu et al. [1], Samouelian [2], and Li-Peng et al. [3] each use different methods to implement isolated word recognition systems and each report that their different recognizers can identify over 90% of the words correctly. However, the word error rate (WER) increases significantly when attempts are made to recognize continuous conversational speech. Liu et al. [4] and Eide et al. [5] each report WER's of 50% or more when performing recognition experiments on corpora of

conversational telephone speech.

The research presented in this thesis attempts to address many of the previously mentioned problems of modern speech recognition. The recognition rates of a continuous speech recognizer can be increased through the use of prosody and landmark detection. Prosodic factors, when incorporated into the phoneme model, can increase the accuracy of a continuous speech recognizer and landmark detection can be used to distinguish between different linguistic features with high accuracy.

Chapter 2 will give a detailed background on prosody and landmarks. Chapter 2 will also describe hidden Markov models (HMM's) and support vector machines (SVM's), which are used for experiments involving prosody and landmarks respectively. Experiments will be described in chapter 3 and chapter 4 presents experimental results. Chapter 5 contains the conclusion and ideas for future work.

Chapter 2

Background

Prosody and landmarks are described in sections 2.1 and 2.2 respectively. The modeling of phones and prosodic factors is accomplished using HMMs. A brief overview of HMMs is given in section 2.3. Section 2.4 describes the acoustic features used for speech recognition in this thesis. Landmark detection is implemented via SVMs. SVMs are described in detail in section 2.5.

2.1 Prosody

Prosody is the study of versification and metrical structure. Some examples of prosodic features include intonation, accentuation, pitch, duration, loudness, pause, and voice quality. Prosodic features are important for two reasons. First of all, prosody changes the meaning of words. As an example, consider the word "right." "Right" can be used as either a question or confirmation of correctness. The difference between the two forms of the word when written is determined by the use of a "?" or a ".". The spoken difference between the question "right?" and the confirmation "right." is determined by the boundary tone at the word ending. A question will end in a rising tone whereas a statement will end in a falling tone. Prosodic features are also important because they affect and change the quality of phones in ways that are detectable.

Only two aspects of prosody will be considered in this thesis, accentuation and phrase

boundary. Specifically, experiments involving prosody will examine the effect of these two factors on the different phones that make up speech segments. Many linguists have studied accentuation and phrase boundary and its effects on speech segments. Several such studies are summarized below.

De Jong [6] studied the supraglottal correlates of linguistic prominence in English speech. In his experiments, De Jong observed an increase in duration of prevoicing in initial voiced stops in stressed syllables. Also observed was an increase in the duration in the closure and in the aspiration of initial voiceless stops. In [6], De Jong also suggests that stress involves a localized shift toward hyperarticulated speech.

Wightman et al. observed the effects of phrase boundaries on speech in [7]. In [7], the authors examine the effect on duration and pause of prosodic phrase boundaries and their experiments show that phrase boundary depth effects the distribution of phoneme duration. Wightman observes that there is segmental lengthening in the rhyme of a syllable when it directly precedes a phrase boundary. The lengthening effects of pre-boundary syllables can be used to distinguish several different types of phrase boundaries.

Fougeron and Keating [8] report that on the edges of prosodic phrase boundaries, final vowels and initial consonants have less reduced lingual articulation. The differences in articulation were manifested in the linguopalatal contact of boundary consonants and vowels. The linguopalatal contact of both consonants and vowels relates directly to the type and size of phrase boundary. Boundary type and size also appear to effect the acoustic duration of post-boundary consonants.

The effect of final lengthening at prosodic boundaries is examined by Edwards et al. in [9] by studying articulator movement patterns. It was found in their study that decreasing intragestural stiffness slows down the syllable. This, in turn, affects the tempo of the spoken word and causes the syllable to be lengthened. Changing intergestural phrasing also affects the syllable duration by decreasing the overlap of a vowel gesture with a consonant gesture. This not only increases the duration of accented syllables but also causes the accented syllable

to be strengthened comparatively to unaccented syllables.

Cho [10] investigates how phonetic features are conditioned by prosodic factors by examining accented, phrase final, and phrase initial syllables. In [10] Cho hypothesizes that accented syllables are characterized primarily by sonority expansion. His experiments show that an accented vowel is usually not affected by coarticulation with a neighboring vowel. Cho also notes that boundary induced articulatory strengthening occurs in phrase final vowel positions and phrase initial consonant positions. Phrase initial vowels are also more susceptible to coarticulation than phrase final vowels. Strengthening effects caused by boundaries and accents cannot be considered the same and Cho discusses several differences between boundary and accent strengthening effects.

Linguists have shown through many studies that the effects of prosody on speech segments produce observable changes on those segments. Other researchers have shown that these changes induced by prosodic factors, specifically accentuation and the intonational phrase boundary, can be detected and recognized with very high accuracy. In [11], Wightman and Ostendorf present two algorithms that can detect and label prosodic phrase boundaries. Prosodic cues used in the algorithms include breaths, pauses, pre-boundary lengthening, boundary tones, and rhythm changes. The first algorithm in [11] was used to detect breaths and silences and could detect 91.3% of the breaths and silence segments when the test set included data from the training set. The second algorithm labeled the phrases with prosodic labels. Wightman and Ostendorf created their own labeling system for phrase boundaries. The digits 0-6 were used to represent different types of boundaries with 0 corresponding to cliticization and 6 corresponding to a sentence boundary. The labeling algorithm only labeled 43% of breaks correctly, but when the authors split the labels into two groups (0-3 as one group, and 4-6 as the second), Wightman and Ostendorf found that the algorithm labeled 91% of the breaks correctly.

In [12] Wightman and Ostendorf use a modification of the algorithm used in [11] to detect prominences and boundary tones in syllables. A syllable could be identified as being in one

of four prosodic groups. A syllable could be a default syllable (containing no pitch accents or boundary tones), a pitch-accented syllable, a boundary-tone syllable, or both pitch accent and boundary tone syllable. Wightman and Ostendorf found that their algorithm could detect 77% of the boundary tones with a 3% rate of false alarms. The algorithm could also detect 86% of pitch accents with a 14% rate of false alarms.

Linguistic studies have shown that accentuation and intonation have profound effects on speech and speech sounds. Wightman and Ostendorf have shown that accentuation and intonation can be detected. The goal of this thesis is to take these ideas one step further for the purposes of speech recognition by exploiting the variation between prosodic and non-prosodic cues along with the ability of those cues to be detected by incorporating prosody into the phoneme model to increase the detectability of speech sounds.

2.2 Manner Features and Landmarks

A knowledge based approach to speech recognition is an approach that assumes a speech signal is comprised of articulatorily-motivated phonological features that have canonical acoustic properties. In order to detect and recognize the speech correctly, the different manner features that make up a speech signal must be correctly recognized and classified. In order to classify these manner features correctly, landmarks, or changes in the signal that correspond to a change in manner features, need to be correctly identified.

Distinctive features are useful because they allow for an economical way of classifying phone segments and because they also allow for a better understanding of allophonic variation. Each phone can in fact be classified by a unique set of distinctive features. Keyser et al. [13] developed a hierarchical distinctive feature model of co-articulation and phonology by combining information about the human vocal tract and manner based features.

A manner feature is a parameter of phonological structure that encodes a perceptually salient, articulator independent aspect of speech production. *Speech*, *continuant*, *sonorant*,

	aa	ae	ah	ao	eh	el	em	en	eng	er	ey	ih	iy	ow	uh	uw
sonorant	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
consonantal	-	-	-	-	-	-	+	+	+	-	-	-	-	-	-	-
syllabic	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
speech	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

Table 2.1: Manner features and their values for the vowels of English

	b	ch	d	dh	f	g	jh	k	p	s	sh	t	th	v	z	zh
continuant	-	-	-	+	+	-	-	-	-	+	+	-	+	+	+	+
sonorant	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
consonantal	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
speech	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

Table 2.2: Manner features and their values for the consonants of English

syllabic and *consonantal* are the five manner features that are considered in this thesis. Features are binary valued as being either positive (+) or negative (-). *Speech* takes on a value of + if a segment is a spoken phone and a value of - if the segment contains non-speech. The feature *continuant* describes the free airflow through the oral cavity. *Sonorant* describes whether a sound is resonant. *Syllabic* is a feature used to distinguish whether or not a sound occurs in the nucleus of a syllable and the feature *consonantal* specifies whether or not the oral cavity is narrowly constricted. Tables 2.1, 2.2 and 2.3 list the five considered features along with the feature values for each phone.

How then are manner features useful? Stevens [14] proposes that the variability between acoustic correlates can be reduced if those acoustic correlates are examined by means of phonetic features. Stevens examines vowels in a set of perceptual experiments designed to examine the effect of manipulation of features such as f0 and breathiness. A similar set of

	hh	l	m	n	ng	r	w	y
continuant		+	-	-	-	+		
sonorant		+	+	+	+	+	-	-
consonantal	-	+	+	+	+	-	-	-
syllabic		-	-	-	-	-	-	-
speech	+	+	+	+	+	+	+	+

Table 2.3: Manner features and their values for the glides and nasels of English

experiments is performed for consonants using the manner features *sonorant*, *anterior*, and *coronal*. Based on his experimental results, Stevens suggests that there might be an advantage in using a representation of phones based on acoustic events and nearby information. Stevens et al. in [15] then takes this idea one step further by using distinctive features to access words stored in a database. Each word is given a manner feature vector representation in the form of a matrix. Acoustic features in the form of landmarks are extracted from a spoken utterance. These landmarks are the classified as specific types of acoustic events and compared to stored word feature matrices.

Other researchers have performed phone classification experiments. Niyogi and Burges [16] investigate the detection of stop consonants using support vector machines. Niyogi and Burges are able to detect stops with a minimum error rate of 16.5

In [17], Juneja performs binary feature classification for the features *speech*, *sonorant*, *continuant* and *syllabic* for purposes of speech segmentation and isolated word recognition. The performed experiments were all done using support vector machines. To determine the best model, Juneja used a variety of different acoustic features that included MFCC and MFCC with deltas and acceleration and also tested various support vector machine parameters. The landmark detection experiments performed in this thesis are very similar to the classification experiments preformed in [16].

2.3 Hidden Markov Models

A hidden Markov model (HMM) is a finite state machine that transitions between states at every time t. Each time the HMM enters a new state, an observation is generated. The observation o_i is known, however, the state i at which o_i was generated is not known or is hidden. The following simplified explanation of HMM algorithms is based on the one given by Rabiner and Juang in [18].

For any observation sequence $O = [o_1, o_2, ...o_t]$ given a model M, the probability $P(O|M)$

can be computed using the forward-backward algorithm. Let $\alpha_t(i)$, the forward variable, be defined as the probability of observing a partial observation sequence given a state q_i until time t. Then $\alpha_t(i)$ can be calculated as follows:

$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N \tag{2.1}$$

where π_i is the initial state distribution, b_i is the probability density of the observation symbol in state i and N is the number of states. Once the initial forward probability is calculated, the forward probability at time $t+1$ can be calculated using the formula

$$\alpha_{t+1}(i) = b_j(o_{t+1}) \sum_{i=1}^{N} \alpha_t(i) a_{ij}, \quad t = 1...T-1, 1 \leq j \leq N \tag{2.2}$$

where a_{ij} is the probability of transitioning from the current state i to the next state j. Given the forward probabilities, the probability $P(O|M)$ can then be found by as follows

$$P(O|M) = \sum_{i=1}^{N} \alpha_t(i) \tag{2.3}$$

The backwards variable $\beta_t(i)$ is defined as the probability of the partial observation sequence from time $t+1$ until the end time T, given a state q_i at time t and a model M. The calculation of β is shown by the two equations below

$$\begin{aligned} \beta_t(i) &= 1 \\ \beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}), \quad t &= T-1, T-2, ...1, 1 \leq i \leq N \end{aligned} \tag{2.4}$$

Given the forward and backward probabilities and the observation sequence O, the optimal state sequence I can be calculated in the following manner. Define $\gamma_t(i)$ to be the probability of being in state q_i at time t given the observation sequence O. Then $\gamma_t(i)$ can

also be written as

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|M)} \tag{2.5}$$

Therefore, the most likely state i_i at time t is

$$i_i = \arg\max_{1 \le i \le N}[\gamma_t(i)], \quad 1 \le t \le T \tag{2.6}$$

The initial state distribution, the transition probabilities and the observation probability densities are all model parameters that need to be optimized in order to optimize $P(O|M)$. Optimization of these parameters can be done using the Baum-Welch algorithm.

Before discussing Baum-Welch re-estimation the quantity $\xi_t(i,j)$ needs to be defined. Let $\xi_t(i,j)$ represent the probability being in state i at time t and transitioning to state j at time $t+1$. $\xi_t(i,j)$ can be written as

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{P(O|M)} \tag{2.7}$$

Baum-Welch re-estimation can now be used to find optimal parameters. The initial state distribution of the current state π_i is re-estimated by the following formula

$$\pi_i = \gamma_1(i), \quad 1 \le i \le N \tag{2.8}$$

The transition probabilities can be re-estimated as

$$a_{ij} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)} \tag{2.9}$$

The observation probability densities can be re-estimated as

$$b_i(k) = \sum_{t=1}^{T}\sum_{k=1}^{K}\gamma_t(i)\delta(o_t = k) \tag{2.10}$$

where

$$\delta(x) = \begin{cases} 1 & x = true \\ 0 & otherwise \end{cases}$$

Baum-Welch re-estimation can be used to optimize HMM parameters for a given set of training data. The Viterbi algorithm is an algorithm that can then be used to determine the optimal state sequence once model parameters have been optimized.

The Viterbi algorithm is implemented as follows. For a given model M, let $\Phi_j(t)$ be equal to the maximum likelihood of observing sequence of speech feature vectors $O = [o_1, o_2, o_t]$ at a given time t in a state j. Then

$$\Phi_j(t) = \max_i [\Phi_j(t-1)a_{ij}]b_j(o_t), \quad 1 \leq j \leq N \tag{2.11}$$

Where

$$\Phi_j(1) = b_j(o_1)\pi_j \tag{2.12}$$

The maximum likelihood $P(O|M) = \max_i \Phi_N(T)a_{iN}$.

2.4 Speech Recognition Acoustic Observations

Modern speech recognition involves processing an input speech signal and detecting the sequences of sounds that make up the speech signal. The individual unique sounds that make up a language are referred to as phones. Table 2.4 shows a list of the different phones in English. The detected sequences of phones are then used to determine the recognized word via a beam search of the recognized sequence with predetermined sequences contained in a dictionary.

The speech signal is usually processed first by separating the signal into segments (usually around 10ms in length) and then converting those segments into feature vectors. In this thesis, the window used was a hamming window. A commonly used set of features are the

aa f*a*ther	ae *a*ct	ah *o*f	ao dr*a*wn	aw *ou*t
ax *a*part	ay *I*	eh s*ai*d	ey th*ey*	ih k*i*d
iy thr*ee*	ow g*o*	oy b*oy*	uh b*oo*k	uw bl*ue*
b *b*at	ch *ch*air	d *d*og	dh *th*ose	f *f*rog
g *g*rill	hh *h*i	jh *j*ob	k *c*an	l *l*ight
m *m*ouse	n *n*ice	ng alo*ng*	p *p*ot	r *r*at
s *s*ix	sh *sh*ed	t *t*en	th *th*row	v *v*ery
w *w*ater	y *y*es	z *z*one	zh mea*s*ure	

Table 2.4: The different sounds in the English language.

Mel frequency cepstral coefficients (MFCC). These features can be generated by first filtering the segment through a mel filterbank centered at Mel frequencies given by the equation:

$$Mel(f) = 2595 \log \frac{1+f}{700} \tag{2.13}$$

where f is frequency. MFC coefficients can be then calculated by taking the Discrete Cosine Transform (DCT) of the log amplitude of the filter outputs. The DCT is computed as follows:

$$c_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{N} m_j \cos\left(\pi i \frac{j-1/2}{N}\right) \tag{2.14}$$

where c_i is ith cepstral coefficient, N is the number of filters in the filterbank, m_j is the log of the filterbank amplitude ot the ith filter and $1 \leq$ i $\leq$ the number of cepstral coefficients.

Once the MFCC features are extracted, a probabilistic model and a decoding algorithm are used to determine which phones the features represent. The probabilistic model usually consists of hidden Markov models (HMM's) that have been trained on phonetic segments. HMM's are discussed in detail in section 2.3. The decoding algorithm used for the experiments in this thesis is the Viterbi algorithm, which is also discussed in detail in section 2.3.

2.5 Support Vector Machines

This section provides a brief overview of support vector machines (SVMs) for the case of binary classification. SVMs are described in detail in [19].

Suppose there is a given observation set consisting of N observations. Each observation, $y_i (i = 1, 2, N)$, is associated with a $\mathbf{x_i} \in R^n$. The observation y_i can take on two different values for the case of binary distinction. Those values are 1 and -1, often called positive and negative examples respectively. Observations sets in the form of $(\mathbf{x_i}, y_i)$ are drawn from an unknown distribution $P(\mathbf{x_i}, y_i)$. Is there an optimal function $f(\mathbf{x_i}, \alpha)$, where α is a set of adjustable model parameters, that can learn the mapping $\mathbf{x_i} \rightarrow y_i$?

The expected error is given by the equation

$$R(\alpha) = \frac{1}{2} \int |y - f(\mathbf{x}, \alpha)| \, dP(\mathbf{x}, y) \tag{2.15}$$

The expected error is also referred to as the expected risk or simply risk.

The quantity $R_{emp}(\alpha)$ is defined as the empirical error. It can be calculated using the formula

$$R_{emp}(\alpha) = \frac{1}{2N} \sum_{i=1}^{l} |y_i - f(\mathbf{x_i}, \alpha)| \tag{2.16}$$

The quantity $\frac{1}{2}|y_i - f(\mathbf{x_i}, \alpha)|$ is referred to as the loss. The loss can only have a value of either 0 or 1 for the case of binary classification. For a loss taking on these values, with probability $1 - \eta$, an upper bound for $R(\alpha)$ can be defined with the equation

$$R(\alpha) \leq R_{emp}(\alpha) + \frac{\sqrt{d(\log(\frac{2N}{d} + 1) - \log(\frac{\eta}{4}))}}{N} \tag{2.17}$$

The quantity η is defined such that $0 \leq \eta \leq 1$ and d is a positive integer called the Vapnik Chervonenkis (VC) dimension. The VC dimension is a property of the function set $f(\mathbf{x}, \alpha)$. The right hand side of equation 2.17 is called the risk bound. The square root term is referred

to as the VC confidence. The VC confidence depends on d and so therefore it also depends on the choice of $f(\mathbf{x}, \alpha)$. The risk and the empirical risk only depend on the specific function from the set of $f(\mathbf{x}, \alpha)$ that is chosen during the SVM training process. The goal is to find a function f such that the risk bound is minimized. This process is referred to as structural risk minimization (SRM).

The observations in a binary dataset can be thought of as points in a p-dimensional space, where p is the dimension of the observation vector. For any given set of observations, there exists a function that can divide the different observations into two optimal sets and thus minimizing the empirical risk. This function is called a hyperplane.

A set of data is considered separable if a hyperplane can be drawn that completely isolates one observation type from the other. Furthermore, they are said to be separable with " margin" $\frac{2}{\|\mathbf{w}\|}$ (the distance between the clouds $y_i = 1$ and $y_i = -1$ is $\frac{2}{\|\mathbf{w}\|}$) under the following conditions. If the vector $\|\mathbf{w}\|$ is defined to be a vector normal to the hyperplane and $\frac{|b|}{\|\mathbf{w}\|}$ is the perpendicular distance from the hyperplane to the origin, where $\|\mathbf{w}\|$ is the Euclidean norm of $\mathbf{w}$, then the grouping conditions for all observable examples in the data set are:

$$\mathbf{x_i} \cdot \mathbf{w} + b \geq 1 \; for \; y_i = 1 \tag{2.18}$$

$$\mathbf{x_i} \cdot \mathbf{w} + b \leq -1 \; for \; y_i = -1 \tag{2.19}$$

Equations 2.18 and 2.19 can be combined and rewritten as

$$y_i(\mathbf{x_i} \cdot \mathbf{w} + b) - 1 \geq 0 \;\; \forall \; i \tag{2.20}$$

The ith constraint is satisfied if and only if the ith example is correctly classified by the hyperplane.

Vapnik has shown that d in equation 2.17 is proportional to $\frac{1}{2}\|\mathbf{w}\|^2$. Therefore, if $R_{emp}(\alpha) = 0$, $R(\alpha)$ can be minimized by minimizing $\|\mathbf{x}\|$ subject to constraints i. For

each constraint, a Lagrange multiplier, α_i, is introduced. We minimize

$$L = \frac{1}{2}\|\mathbf{w}\| + \sum_i \alpha_i \times constraint_i \tag{2.21}$$

That is, the Lagrangian is then given by

$$L = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i y_i(\mathbf{x_i} \cdot \mathbf{w} + b) + \sum_{i=1}^{l} \alpha_i \tag{2.22}$$

To find the optimal L, the Lagrangian must be minimized with respect to $\mathbf{w}$ and b. Setting $\frac{dL}{d\mathbf{w}} = 0$ and $\frac{dL}{db} = 0$, we get

$$\begin{aligned} \sum_i \alpha_i y_i \mathbf{x_i} &= \mathbf{w} \\ \sum_i \alpha_i y_i &= 0 \end{aligned} \tag{2.23}$$

Inserting equation 2.23 into equation 2.22 yields:

$$L = \sum_i a_i + \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j} \tag{2.24}$$

which satisfies all conditions and therefore, provides an optimal Lagrangian.

More often than not, the data will be non-separable. In other words, there will not be an optimal hyperplane with which to divide the data into two distinct sets. In the case of non-separable data, equation 2.24 is no longer true so minimizing 2.22 is no longer equivalent to minimizing $\|\mathbf{w}\|^2$. Non-separable data can be accounted for by defining the positive slack variable, ξ_i $(1 \leq i \leq L)$, and adding it to the constraints proposed for the separable case in equations 2.18 and 2.19. $\xi_i = 0$ for correctly classified x_i, but $\xi_i > 0$ for incorrectly classified $\mathbf{x_i}$. Equations 2.18 and 2.19 become

$$\mathbf{x_i} \cdot \mathbf{w} + b \;\geq\; 1 - \xi_i \; for \; y_i = 1$$

$$\begin{aligned} \mathbf{x_i} \cdot \mathbf{w} + b &\leq -1 + \xi_i \; for \; y_i = -1 \\ \xi_i &\geq 0 \end{aligned} \tag{2.25}$$

For a classification error to occur, ξ_i must exceed unity. An upper bound on the number of training errors can be determined to be $R(\alpha) \leq \sum_i \xi_i$. Thus, equation 2.17 becomes

$$L = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i - \sum_i \alpha_i y_i(\mathbf{x_i} \cdot \mathbf{w}_b) + \sum_i \alpha_i \tag{2.26}$$

where C is the error penalty. The Lagrangian calculated in equation 2.24 needs to be minimized for the non-separable case. The optimal Lagrangian must satisfy the following conditions.

$$\begin{aligned} 0 \leq \; \alpha_i \; &\leq C \\ \mathbf{w} \; &= \; \sum_i \alpha_i y_i \mathbf{x_i} \\ \sum_i a_i y_i \; &= \; 0 \end{aligned} \tag{2.27}$$

The optimal Lagrangian is given by the equation

$$L = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i} \cdot \mathbf{x_j} \tag{2.28}$$

Until this point, it has been assumed that the hyperplane is a linear function. This is not always the case. A non-planar separatrix can be constructed by using $K(\mathbf{x_i}, \mathbf{x_j})$ in place of $\mathbf{x_i}$ and $\mathbf{x_j}$ in the equations above, where $K(\mathbf{x_i}, \mathbf{x_j})$ is a positive definite "kernel" function. The mathematical function that determines the shape of the hyperplane is called the kernel. Aside from the linear kernel, other common kernels are the radial basis function, polynomials and sigmoid functions. Optimization for each different kernel follows the same general procedure as the optimization for the linear kernel.

Chapter 3

Experiments

Section 3.1 describes the experiments involving the incorporation of prosody into the phonetic model. Section 3.2 describes landmark detection experiments.

3.1 Prosody

3.1.1 ToBI Labeling

Tone and Break Indices (ToBI) is a convention of labeling used for marking pitch accents and prominence as well as phrase boundary. Seven types of pitch accent were labeled in the speech corpus used for experimentation. The corpus, Radio News, is described in detail in the next section. The seven accent types labeled are H*, L*, !H*, L+H*, L+!H*, L*+H, and H+!H*. The H and L labels represent high and low tones respectively. A "*" is used to represent tone alignment. The ! is used to mark the occurrence of a downstep. Boundary tone labels were also labeled in the corpus. The labels used for intonational phrase boundary labeling were H% and L%. In ToBI labeling, word boundaries are also assigned an index to determine the degree of decoupling between a given word pair. Indices can range from 0-4 where an index of zero means no word boundary, 1 represents a normal word boundary, 2 and 3 represent different levels of intermediate phrase boundary and an index of 4 represents an intonational phrase boundary. A more detailed description of the ToBI labeling system

is given in [20].

3.1.2 Radio News

Boston University's Radio News Corpus [21] was used to the study of the effects of prosodic factors on speech recognition. This corpus, originally designed for speech synthesis applications, was chosen because it is one of the largest prosodically labeled English databases.

Experiments were run using speakers F1A, M1B, F2B, M2B, and F3A. Utterances from these speakers were only included in the dataset if they were accompanied by the two transcription files that had the extensions ".ton" and ".brk." The files labeled as ".ton" transcription files mark the times that pitch accents occur. The labeling in the ".ton" files was done using the ToBI labeling system, described in section 3.1.1. The ".brk" transcription files label the times at which boundaries occur and the strength of the boundary (0-4). The labeling of these files was also done according to ToBI conventions. Radio News also contained word transcription files, denoted with the ".wrd" extension and dictionaries in the form of ".prn" files. The ".prn" files also specify the lexically stressed vowel in each word.

The dataset contained over three hours of data divided between a training set and a test set. The test set was approximately 10% the size of the training set. The training set included 23,103 words in 272 files. 11,386 of the words were accented. The training set also included 3829 intonational phrase boundaries.

3.1.3 The Hidden Markov Toolkit

The hidden Markov Toolkit (HTK) [22] is open source software designed for the creation and manipulation of HMMs. Of the individual tools included in the toolkit, only the tools HCopy, HRest, HERest, HVite, and HResults were used for experimentation. Experiments are described in sections 3.1.4 and 3.1.5. The individual tools are described below.

HCopy is a tool used for data extraction and file conversion. HCopy takes an audio file

as input and outputs a feature file. Supported audio input formats include HTK, Esignal, NIST, SCRIBE, SDES1, AIFF, SUNAU8, WAV, and OGI. The feature types HCopy can be used to output are HTK wave file, LPC, LPREFC, LPCEPSTRA, IREFC, MFCC, FBANK, MELSPEC, USER, and DISCRETE. Input and output file formats are described in detail in the HTK manual available at http://htk.eng.cam.ac.uk/.

The function of HRest is to perform Baum-Welch re-estimation for a single model given a specified set of training tokens. Baum-Welch re-estimation is described in detail in section 2.3. While performing parameter re-estimations, HRest can be forced to output the log likelihood of the model, which is a feature that is exploited in the experiment described in section 3.1.4. HRest can also be used to generate seed HMMs, which can be useful for purposes of building phoneme based recognition systems.

The tool HERest is similar to HRest in that it also implements the Baum-Welch algorithm. However, unlike HRest, HERest performs only a single iteration on a set of HMMs using an embedded re-estimation technique.

HVite is a general purpose word recognizer which implements the Viterbi algorithm. Aside from performing standard word recognition, HVite can also be used to create time aligned phonetic transcriptions for a given utterance. HVite also has the ability to generate word lattice networks.

HResults is a tool used to interpret the output of a recognizer that has been trained using HRest or HERest and then tested using HVite. HResults reports the number of insertions, deletions and substitutions along with the accuracy and correctness of a recognizer. Insertions are places where a word that was not spoken is placed between two words that were. A deletion occurs at a place where a word was supposed to have been output, but was not. A substitution is a place where the recognizer exchanges one word for a different, incorrect word. Correctness is calculated by the formula

$$\%Corr = \frac{T - S - D}{T} \tag{3.1}$$

```
abilities ax b ih l ax t iy z
borrowing b aa r ow ax ng
build b ih l d
expected ax k s p eh k t ax d
first f ah r s t
know n ow
reactive r iy ae k t ih v
thousand th aw z ax n d
yesterday y eh s t er d ey
```

Figure 3.1: An example of an HTK non-prosodic dictionary

where S is the number of substitutions, D is the number of deletions and T is the total number of tokens. Accuracy is calculated in the formula

$$Acc = \frac{T - S - D - I}{T} \tag{3.2}$$

where I is the number of insertions.

All HTK tools require a variety of auxiliary files in order to function correctly. All of these files are described in detail in the HTK manual and will not be discussed here. However, two of the file types, the dictionary and the master label file (MLF) are important for experimental purposes and as such, a brief description of each file type is given below.

The dictionary is a file that contains all the words plus the definitions of those words that are "known" to the recognizer. Word definitions are in the form of pronunciations which are symbolically represented through use of phones. An example of an HTK dictionary is shown in figure 3.1.

An MLF contains the transcriptions that are associated with the utterances in the specified dataset. There are two levels of MLF file, word and phone. A word level MLF contains labels for all the words spoken in an utterance while a phone level MLF contains phone labels for all the sounds created by a speaker in an utterance. Figure 3.2 shows examples of HTK format word and phone MLF files. The times of word and phone boundaries may or may not need to be included in an MLF. Certain tools, like HRest, require that times be

600000 2000000 in	600000 1700000 ih
2000000 5800000 nineteen	1700000 2000000 n
5800000 9300000 seventy	2000000 2300000 n
9300000 13600000 six	2300000 3500000 ay
13600000 19100000 democratic	3500000 3900000 n
19100000 21600000 governor	3900000 4500000 t
	4500000 5300000 iy
	5300000 5800000 n
(a)	**(b)**

Figure 3.2: **a**. An HTK format word level transcription. **b**. An HTK format phonetic level transcription. The words transcribed in the phonetic transcription are "in nineteen." The transcriptions in **a** and **b** are referred to as master label files (MLF's). The first two columns (the start and end times of the word or phone) are in 100 ns units.

included with the label. Other tools, such as HERest, do not require times. When times are included in the MLF transcription, they must be in units of 100 ns.

3.1.4 Phone Splitting

In order to determine by what amount prosody will improve recognition rates, it would first be useful to determine which prosodic factors, if any, will be helpful to include in the model and which, if any, will be harmful to include in the model. This can be accomplished by examining the log likelihoods of prosodically independent phones and comparing those likelihoods to log likelihoods of a phone model that has been split into two prosodic allophones.

The monophone set used for this experiment and the experiment described in section 3.1.5 was a version of the Sphinx monophone set described in [23]. The prosody independent set of monophones contained 47 distinct phones, including silence. The Sphinx monophone set merges closures with their stop consonants and also merges commonly occurring patterns of phones such as the "t" and "s" sounds at the end of the word "cats."

A series of allophone sets for purpose of binary comparison were created in order to be able to examine one distinct prosodic feature independently from another. The defined

	Stop	Fricative	Liquid	Nasal	Vowel
Left	LS	LF	LL	LN	LV
Right	RS	RF	RL	RN	RV

Table 3.1: The ten non-prosodic, context dependent allophone sets and their abreviations. The words "Left" and "Right" refer to which neighboring phone is being considered. The phonetic classes, stop, fricative, liquid, nasal or vowel, specify the type of the considered neighboring phone.

prosodic allophone sets will be referred to as the Accented (AC), Content-Function (CF), Phrase Final (PF) and Phrase Initial (PI) allophone sets. A series of ten non-prosodic allophone sets were also defined. These allophone sets are listed in table 3.1.

An independent allophone set (IND) was also defined for use as a control. The IND set contained only the 48 original phones and did not split them in to any subgroups. The dictionary and the MLF phone and word transcriptions for this set looked similar to those shown in figure 3.2.

The AC set split phones into two groups, accented and unaccented. Allophones contained in this set could not be distinguished on the basis of phrasal position. The criterion for an accented phone is defined separately for vowels and consonants. An accented vowel was defined to be the vowel in the lexically stressed syllable of a word that had been transcribed in the database as being accented. Accented consonants are not clearly defined, so therefore, in order to determine the full effect of accentuation on consonants, three sub-sets of accented consonants were created. These subsets were called Coda, Onset and All. The Coda subset defined an accented consonant to be any consonant occurring in the coda of the syllable containing an accented vowel. The Onset subset defined an accented consonant to be any consonant contained within the onset of the syllable containing the accented vowel. The All subset combined both Onset and Coda by defining an accented consonant as any consonant occurring in the syllable containing an accented vowel. Unaccented phones were specified to be any phones that were not accented. In the AC transcription and dictionary files, accented words are distinguished from unaccented words with a "!". For example, if the word "gerbil"

600000 2000000 in
2000000 5800000 nineteen!
5800000 9300000 seventy!
9300000 13600000 six!
13600000 19100000 democratic!
19100000 21600000 governor

Figure 3.3: An example of the accented word level MLF. All three subsets used the same word level MLF.

abilities! ax b! ih! l ax t iy z	abilities! ax b ih! l ax t iy z	abilities! ax b! ih! l ax t iy z
borrowing! b! aa! r ow ax ng	borrowing! b aa! r ow ax ng	borrowing! b! aa! r ow ax ng
build! b! ih! l! d!	build! b ih! l! d!	build! b! ih! l d
first! f! ah! r! s! t!	first! f ah! r! s! t!	first! f! ah! r s t
know! n! ow!	know! n ow!	know! n! ow!
reactive! r iy ae! k! t ih v	reactive! r iy ae! k! t ih v	reactive! r iy ae! k t ih v
thousand! th! aw! z ax n d	thousand! th aw! z ax n d	thousand! th! aw! z ax n d
yesterday! y! eh! s! t er d ey	yesterday! y eh! s! t er d ey	yesterday! y! eh! s t er d ey
(a)	**(b)**	**(c)**

Figure 3.4: **a**. An example dictionary from the AC All allophone subset. **b**. An example dictionary from the AC Coda allophone subset. **c**. An example dictionary from the AC Onset allophone subset. All three dictionaries also had definitions for unaccented words, such as those shown in figure 3.1

were spoken with a pitch accent, then it would be transcribed in both the word MLF and dictionary as "gerbil!". The unaccented version of the word would simply be transcribed as "gerbil." A sample accented word transcription is shown in figure 3.3. This same method was used to distinguish accented phones from unaccented phones. Each subset, Coda, Onset and All, had distinct phone level transcriptions and dictionaries. Examples of dictionaries and phone transcriptions from each subset are shown in figures 3.4 and 3.5 respectively.

The CF allophone set distinguished phones as being either a content phone or a function phone. A function phone is a phone that occurs in a function word such at "the" or "a." The phones used to represent the pronunciation of a function word were given the suffix "_f." Figure 3.6 shows all the function words that were considered in this experiment. Any phone that occurred in a content word, such as "Massachusetts" or "court", was considered to be a

2400000 2500000 n	2400000 2500000 n	2400000 2500000 n
2500000 3600000 ay	2500000 3600000 ay	2500000 3600000 ay
3600000 3900000 n	3600000 3900000 n	3600000 3900000 n
3900000 4500000 t!	3900000 4500000 t	3900000 4500000 t!
4500000 4800000 iy!	4500000 4800000 iy!	4500000 4800000 iy!
4800000 5800000 n!	4800000 5800000 n!	4800000 5800000 n
5800000 6800000 s!	5800000 6800000 s	5800000 6800000 s!
6800000 7500000 eh!	6800000 7500000 eh!	6800000 7500000 eh!
7500000 7700000 v	7500000 7700000 v	7500000 7700000 v
7700000 8000000 ax	7700000 8000000 ax	7700000 8000000 ax
8000000 8300000 n	8000000 8300000 n	8000000 8300000 n
8300000 8400000 t	8300000 8400000 t	8300000 8400000 t
8400000 9200000 iy	8400000 9200000 iy	8400000 9200000 iy
(a)	**(b)**	(c)

Figure 3.5: **a**. An example of a phone level transcription for the AC All allophone subset. **b**. An example of a phone level transcription for the AC Coda allophone subset. **c**. An example of a phone level transcription for the AC Onset allophone subset. The transcribed words are "nineteen seventy."

content phone. Content phones were indicated with a "_c" suffix. An example dictionary for the CF allophone set is shown in figure 3.7a. The word level MLF for the CF allophone set was indistinguishable from the IND word level MLF. The CF phone MLF, created from the dictionary using HVite, is shown in figure 3.7b. The purpose of the CF allophone set was to examine phone duration. Function words are spoken at a much faster rate than content words as is described in [23].

a	did	in	or	where
all	find	is	show	what
and	for	it	than	why
any	from	list	that	will
are	get	many	the	with
at	give	more	their	would
be	have	of	to	
been	has	on	use	
by	how	one	was	

Figure 3.6: A list of function words.

any eh_f n_f iy_f	600000 1700000 ih_f
borrowing b_c aa_c r_c ow_c ax_c ng_c	1700000 2000000 n_f
build b_c ih_c l_c d_c	2000000 2300000 n_c
first f_c ah_c r_c s_c t_c	2300000 3500000 ay_c
know n_c ow_c	3500000 3900000 n_c
reactive r_c iy_c ae_c k_c t_c ih_c v_c	3900000 4500000 t_c
the th_f ax_f	4500000 5300000 iy_c
thousand th_c aw_c z_c ax_c n_c d_c	5300000 5800000 n_c
would w_f uh_f d_f	
(a)	**(b)**

Figure 3.7: **a**. An example dictionary from the CF allophone set. **b**. An example phone transcription from the CF allophone set. The transcribed words are "in nineteen." The word level transcription for this allophone set would look the same as the transcription in figure 3.2

The PF allophone set split phones into "phrase final" and "phrase medial" phones. A phone was considered to be phrase final if it occurred in the nucleus or coda of the final syllable in a word that preceded an intonational phrase boundary, otherwise it was considered to be phrase medial. In the transcriptions and dictionary, phrase final phones were distinguished from phrase medial phones with the suffix "B4" which indicated that these phones occurred before a boundary with a ToBI index of 4. This convention was also used to distinguish phrase final words from phrase initial words. A sample word transcription and phone transcription are shown in figure 3.8 and a sample dictionary is shown in figure 3.9a.

The PI allophone set separated phones into the groups "phrase initial" and "phrase medial." A phrase initial phone could occur only in the onset or nucleus of the first syllable in a word that followed directly after an intonational phrase boundary. All phones in the PI allophone set that were not considered to be phrase initial were labeled as being phrase medial. Phrase initial phones were distinguished with the prefix "B4" to indicate that these phones are ones that occur just after a boundary with ToBI index of 4. Phrase initial words were distinguished from phrase medial words using the same method. A sample dictionary is shown in figure 3.9b. Sample word and phone transcriptions are shown in figure 3.10.

600000 2000000 in
2000000 5800000 nineteen
5800000 9300000 seventy
9300000 13600000 sixB4
13600000 19100000 democratic
19100000 21600000 governor

(a)

9300000 10700000 s
10700000 11800000 ihB4
11800000 12300000 kB4
12300000 13600000 sB4
13600000 14700000 d
14700000 15400000 eh
15400000 15900000 m
15900000 16400000 ax
16400000 17100000 k
17100000 17500000 r
17500000 18000000 ae
18000000 18300000 dx
18300000 18600000 ih
18600000 19000000 k

(b)

Figure 3.8: **a**. An example word level PF transcription. **b**. An example phone level PF transcription. The transcribed words are "six democratic."

abilitiesB4 ax b ih l ax t iyB4 zB4
borrowingB4 b aa! r ow axB4 ngB4
buildB4 b ihB4 lB4 dB4
firstB4 f ahB4 rB4 sB4 tB4
knowB4 n owB4
reactiveB4 r iy ae k t ihB4 vB4
thousandB4 th aw z axB4 nB4 dB4
yesterdayB4 y eh s t er d eyB4

(a)

B4abilities B4ax b ih l ax t iy z
B4borrowing B4b B4aa r ow ax ng
B4build B4b B4ih l d
B4first B4f B4ah r s t
B4know B4n B4ow
B4reactive B4r B4iy ae k t ih v
B4thousand B4th B4aw z ax n d
B4yesterday B4y B4eh s t er d ey

(b)

Figure 3.9: **a**. An example PF dictionary. **b**. An example PI dictionary. Both dictionries also contain non-prosodic entries such as those in figure 3.1

```
600000 2000000 B4in
2000000 5800000 nineteen
5800000 9300000 seventy
9300000 13600000 six
13600000 19100000 B4democratic
19100000 21600000 governor
```

(a)

```
9300000 10700000 s
10700000 11800000 ih
11800000 12300000 k
12300000 13600000 s
13600000 14700000 B4d
14700000 15400000 B4eh
15400000 15900000 m
15900000 16400000 ax
16400000 17100000 k
17100000 17500000 r
17500000 18000000 ae
18000000 18300000 dx
18300000 18600000 ih
18600000 19000000 k
```

(b)

Figure 3.10: **a**. An example PI word transcription. **(b)**. An example PI phone transcription. The transcribed words are "six democratic."

The allophone sets listed in table 3.1 were not defined on a prosodic basis, but were instead defined on a phonetic context basis. Each set listed in the table split phones into two binary categories that were based on the position and the class of neighboring phones. For example, the Left Fricative (LF) set distinguishes between phones that are preceded by a fricative and all other phones. The Right Fricative (RF) set distinguishes between all phones that are followed immediately after by a fricative and all other phones. This allophone set served two purposes. First of all, it allowed for examination of co-articulatory effects between different classes of phones. Second, allophones from these sets that showed improvements when split were used as additional parameters for the prosody independent recognizer described in section 3.1.5.

Prosodic word level transcriptions were created for each allophone set using the ".wrd", ".brk" and ".ton" files (described in section 3.1.2) and the programming language Perl. Dictionaries were also created using Perl and the ".prn" files. HVite was then used, along with pre initialized HMM's, to create time aligned phonetic transcriptions for each allophone

set. Time aligned transcriptions were created for both the training dataset and the test dataset.

HCopy was used to calculate the MFCC coefficients, along with the delta and acceleration coefficients for both the training and the test set. HRest was then used to re-estimate the pre-initialized HMM definitions on the training dataset. HRest uses Baum-Welch re-estimation, which is described in section 2.3. The models were re-estimated until they had converged or until HRest had preformed 100 iterations of the algorithm. Once the models had converged, they were re-trained on the test dataset for a single iteration using HRest. During this single iteration, HRest was forced to output and save the log likelihood for each re-estimated phone model. This procedure was completed for every phone in all allophone sets.

Once all the model likelihoods had been found, the IND allophone set was used as a basis for comparison to determine which phones should be split according to what prosodic factors. A weighted average was calculated between the likelihoods of the prosodically split allophones. The weighted averages were then compared to the likelihoods of the IND phones. The following example illustrates the process for a phone *phn* that has been split into phrase final and phrase medial allophones, $phnB4$ and *phn* respectively.

Suppose that there are M occurrences of *phn* and N occurrences of $phnB4$. The log likelihoods of *phn* and $phnB4$ are L_{phn} and L_{phnB4} respectively. The weighted average is calculated as follows:

$$WA = \frac{M}{M+N} \times L_{phn} + \frac{N}{M+N} \times L_{phnB4} \tag{3.3}$$

Suppose now that the log likelihood of the un-split phone "phn" in the IND allophone set is LL_{phn}. The algorithm used to determine if *phn* should be split is as follows

$$\begin{aligned} if(WA &> LL_{phn});\ \textit{then split phn into prosodic allophones} \\ if(WA &\leq LL_{phn});\ \textit{then do not split into prosodic allophones} \end{aligned} \tag{3.4}$$

This simple algorithm was used for each prosodic split to determine if that split had a positive or negative impact on the model. Results are shown in table 4.1 and are discussed in section 4.1.

3.1.5 Fixed Parameter Recognizer

The experiment described in section 3.1.4 was used to determine which prosodic splits allowed for the most significant improvements to the phoneme model. This information can be directly incorporated into a speech recognizer. The experiment described in this section describes the building and comparison of both a prosody independent recognizer and a prosody dependent recognizer. The 2-best prosodic splits were determined for each allophone and built into the prosody dependent recognizer.

As mentioned in section 3.1.4, the prosody independent monophone set contained only 47 monophones, including silence. The result of splitting any of these monophones into prosodic allophones is that the number of phones, and thus the number of model parameters, will increase. An increase in model parameters will have a tendency to favor prosodic splitting, so therefore, in order to achieve an accurate comparison between prosody independent and prosody dependent recognizers, the number of phones in the prosody independent recognizer should be increased to match the number in the prosody dependent recognizer.

The context dependent allophone sets (described in section 3.1.4) were used to make the model parameters equal between the two recognizers. This was done by determining the 2-best context splits for each phone and then building these context splits into the prosody independent recognizer. The prosody dependent recognizer was built by replacing the context models for a given phone with prosodic models. If a phone had no prosodic splits that allowed for improvement in the log likelihood of the model, then the context parameters were not replaced.

As shown in table 4.1, different phones tend to favor different prosodic splits. Therefore, aspects of each prosodic effect needed to be integrated into the MLF and dictionary. Figure

600000 2000000 B4in
2000000 5800000 nineteen!
5800000 9300000 seventy!
9300000 13600000 sixB4!
13600000 19100000 B4democratic!
19100000 21600000 governor

(a)

600000 1700000 B4ih
1700000 2000000 n
2000000 2300000 n_rv
2300000 3500000 ay!
3500000 3900000 n_rs
3900000 4500000 t
4500000 5300000 iy
5300000 5800000 n

(b)

Figure 3.11: **a**. The prosody dependent word transcription. **b**. The prosody dependent phone transcription. When a phone had no prosodic allophones, then its context allophones were included in the recognizer instead. The "_rv" and "_rs" refer to the "Right Vowel" and "Right Stop" allophone sets respectively. Other labels are shown in table 3.1

.

3.11 shows examples of the word level MLF and of the dictionary. These files were created by using Perl to merge the ".brk" and ".ton" files with the ".wrd" and ".prn" files that were included in the Radio News corpus, described in section 3.1.2.

In order to train each recognizer, time-aligned phone transcriptions needed to be generated using HVite. Files containing the MFCC, delta and acceleration coefficients were already present from the previous experiment and did not need to be regenerated. The time-aligned prosody independent and dependent phone transcriptions were used to train independent and dependent recognizers respectively. Training was done using the tool HERest. Once the models finished training, the Viterbi decoder HVite was used to obtain results. The results are discussed in section 4.2 and are shown in table 4.2.

3.2 Landmarks

3.2.1 TIMIT

The TIMIT database [24] was designed to provide speech data for the studies of acoustics and phonetics and also for the building of speech recognition systems. TIMIT is a corpus of read

speech containing a total of 6300 sentences. The sentences were designed to be phonetically diverse, meaning that they contain examples of every possible sequence of phones, especially rare phones sequences or phones sequences that may be of particular interest to linguists. TIMIT includes 48 monophones. The data were collected from 630 speakers, both male and female, from 8 different dialect regions of the United States. TIMIT contains its own time aligned phonetic transcriptions. The training dataset contained an assortment of 140 wav files and the test dataset contained 40 wav files.

3.2.2 LIBSVM

LIBSVM [25] is a program useful for training support vector machines. Support vector machines are described in section 2.5. The program contains three tools, svm-scale, svm-train and svm-predict. The tool svm-scale is used to scale input vectors for a reduction of computation error, svm-train is the tool used to train the model and svm-predict is used to test the accuracy of the model.

The input data file has the format:

O_1 1:$value_1$ 2:$value_2$ 3:$value_3$...
O_2 1:$value_1$ 2:$value_2$ 3:$value_3$...
O_3 1:$value_1$ 2:$value_2$ 3:$value_3$...
...
O_N 1:$value_1$ 2:$value_2$ 3:$value_3$...

where O_1 through O_N are a series of observations and the lines "1:$value_1$ 2:$value_2$ 3:$value_3$..." represent the vector that corresponds to each observation. The numbers 1, 2, 3, in front of the colon are the index of the numbers, each represented by the "$value_{index}$" contained in the vector. An example input file is shown in figure 3.12.

```
-1    1:-0.100951 2:-0.0237191 3:-0.0392793 4:0.515768
1     1:-0.780494 2:0.667376 3:-0.0564404 4:0.530383
-1    1:0.508571 2:0.0834868 3:0.104343 4:0.475912
-1    1:0.171674 2:-0.522371 3:-0.348407 4:0.550595
-1    1:0.518083 2:-0.723123 3:-0.445102 4:0.189794
1     1:-0.204792 2:-0.00798731 3:-0.163721 4:0.441962
```

Figure 3.12: An example SVM input file. In this figure, the input vectors are length 4. The actual input files used contain vectors of length 429.

3.2.3 Support Vector Feature Classifiers

The purpose of these experiments was to train groups of SVM classifiers that could detect different landmarks.

The SVM input consisted of a concatenation of 11 individual frame vectors. An individual frame vector was in the form of MFCC, delta and acceleration coefficients for a single frame of speech. The first five vectors in the onset of the landmark were concatenated in chronological order. The frame containing the desired landmark was then concatenated to its onset vector. The first five frames in the offset of the landmark were then concatenated in chronological order and then added onto the end of the onset+landmark vector. The data for all classifiers was in this 11 frame concatenated format.

The first group of trained classifiers were designed to distinguish between a + and a transition for the features *speech*, *sonorant*, *continuant* and *syllabic*. For example, if the SVM was trained for the [+sonorant] vs [-sonorant] distinction, then an observation would have a value of 1 if the landmark of interest occurred in the transition to a nasal, vowel or glide. A value of -1 would be assigned if the landmark occurred in the transition to fricative or stop. Table 3.2 shows the experimental conditions and observation values for this set of SVM classifiers.

The second group of SVM classifiers was trained to recognize [+-feature] transitions and [-+feature] transitions where feature could be *speech*, *consonantal*, *sonorant*, *+consonantal/continuant*. In other words, there would be two SVMs for each manner feature distinction. The SVM for [-+sonorant] features would be trained to detect the transitions from

	+ Transitions	- Transitions
Speech	nasals, fricatives, liquids, stops, vowels	non-speech
Sonorant	nasals, liquids, vowels	fricatives, stops
Continuant	fricatives	stops
Syllabic	vowels	nasals, liquids

Table 3.2: The experimental conditions for the first set of SVM classifiers. Column 1 lists the considered feature. Transitions to the types of phones listed in column 2 were considered to be positive transitions. Transitions to the types of phones in column 3 were considered to be negative transitions.

	From	To
-+consonantal	vowel, liquid	nasal, fricative, stop
+-consonantal	nasal, fricative, stop	vowel, liquid
-+sonorant	fricative, stop	nasal, vowel, liquid
+-sonorant	nasal, vowel, liquid	fricative, stop
+consonantal/-+continuant	stop	fricative
+consonantal/+-continuant	fricative	stop

Table 3.3: The experimental conditions for the second set of SVM classifiers. Transitions listed in this table were labeled as being +1. Any other transitions were labeled as -1.

fricatives and stops to nasals, vowels and glides. The SVM for [+-sonorant] would be trained to detect the transitions from nasals, vowels and glides to fricatives and stops. Table 3.3 shows all the experimental conditions and observation values for this set of SVM classifiers.

Chapter 4

Results

4.1 Phone Splitting

Results from the phone splitting experiment, described in section 3.1.4, are depicted in table 4.1. In table 4.1, the first column indicates which monophone is being examined. The second column, labeled "Examples," specifies the number of examples of a particular monophone in the test data set. The "No Split" column indicates the log likelihood of a monophone when it has not split into any prosodic or contextual factors. Columns 4 through 10 correspond to the prosodic allophone sets described in section 3.1.4. The numbers in these columns specify the net change between the between the likelihood of the un-split monophone and the allophonic weighted average. A negative number indicates an improvement in the log likelihoods. The columns 11 through 20 in the table correspond to the context dependent allophone sets, also described in section 3.1.4. These columns specify the net change between the allophonic weighted average and log likelihood of the context independent monophone and can be read just as columns 4 through 10. A blank space in any column indicates that there was not enough data for the experiment to be successful.

In general, splitting monophones based on the context of neighboring phones allows for improvement to the phoneme model. The only major exception appears to be the case of a phone followed by a fricative, as can be seen in the "RF" column of table 4.1. The table suggests that knowing whether or not a phone is followed by a fricative improves the model

	CF	PF	PI	All	Onset	Coda	LL	RL	LS	RS	LN	RN	LF	RF	LV	RV
aa		13.3	18.1	15.1	18.9	14.5	-19.3	-27.0	-17.2	-20.7	-17.3	-26.4	-16.8	-0.7	-18.7	-18.1
ae	-17.2	-18.4	-30.5	-33.0	-32.0	-32.2	-13.7	-12.0	-12.5	-16.3	-14.1	-19.6	-14.1	-1.7	-12.1	-12.5
ah	-9.3	7.3	-18.1	-21.4	-21.4	-21.4	-9.8	-7.7	-9.7	-9.6	-8.0	-14.5	-7.8	-3.5	-6.4	-6.9
ao	-15.8	102.6	25.5	24.3	24.8	24.3	-16.6	-23.8	-15.6	-15.2	-15.2	-14.7	-14.9	-2.8	-13.8	-12.0
aw		-5.8	-30.0	-35.5	-35.1	-35.8	-23.3	-20.5	-22.0	-29.3	-23.2	-27.0	-28.2	-0.6	-22.7	-23.2
ax	-4.8	-46.5	10.9				-5.9	-8.0	-5.5	-7.6	-5.5	-8.7	-5.4	-1.7	-4.2	-4.0
ay		-63.6	-89.4	-88.8	-88.9	-89.9	-25.3	-28.1	-24.1	-25.3	-23.9	-28.4	-24.9	-2.6	-22.7	-25.1
b			30.7	33.5	31.1	31.7	-2.2	-5.1	-2.2	-2.2	-2.2	-2.2	-2.2		-2.2	-5.3
bd	-4.7	-43.5	-74.3	-74.9	-75.4	-75.1	-4.6	-3.1	-4.0	-3.1	-3.0	-3.1	-3.3	0.2	-4.8	-3.1
ch		-7.4	-51.3	-53.3	-54.7	-51.1	-4.1	-4.2	-5.2	3.2	-4.4	-4.7	-4.6	1.9	-6.0	-6.3
d		-2.6	-25.9	-27.9	-29.2	-25.7	-5.6	-12.3	-5.7	-4.7	-7.9	-5.1	-5.2	0.4	-6.0	-10.9
dd		38.7	50.8	50.7		49.6	-3.0	-3.3	-3.1	-4.2	-3.5	-3.1	-2.5	0.6	-3.7	-4.0
dh	-4.5		-1.4	-0.3	-0.3		-3.5	-3.2	-3.7	-3.2	-5.0	-3.2	-4.1	1.5	-5.0	-3.3
dx		101.1		96.6		96.6	-2.7	-3.0	-3.2	-3.2	-3.2	-5.2	-3.2	0.0	-2.6	-3.2
eh		-76.0	-86.2	-86.0	-82.8	-86.0	-14.2	-23.3	-12.6	-18.5	-12.2	-15.9	-15.8	-3.0	-11.2	-11.0
en		-54.6		29.2		29.2	-12.4	-11.1	-11.8	-13.8	-8.0	-11.0	-12.5	-1.8	-10.6	-12.2
er		-22.8	-36.5	-39.6	-44.2	-39.6	-13.1	-14.0	-12.9	-14.8	-12.5	-14.9	-12.0	-1.6	-11.5	-13.7
ey		-40.1	-64.3	-66.0	-65.4	-66.1	-21.9	-23.5	-20.4	-21.7	-20.3	-24.6	-21.2	-4.0	-17.9	-18.4
f	-8.7	27.5	25.0	24.5	23.2	25.5	-5.2	-6.9	-5.8	-4.6	-5.3	-4.6	-6.6	1.7	-6.9	-9.8
g		-5.2	6.1	4.7	4.4	6.9	-5.4	-7.3	-3.0	1.1	-4.6	-3.7	-4.8	0.0	-5.4	-10.1
hh			27.6	26.9	27.0	29.2	-4.9	-5.2	-4.9	-5.3	-4.8	-5.3	-6.8	0.4	-5.3	-5.2
ih	-6.5	-55.6	-24.4	-26.1	-23.9	-26.2	-7.5	-9.8	-6.6	-8.5	-6.8	-13.3	-7.2	-1.2	-6.3	-5.5
iy	-12.7	-39.3	-24.9	-25.7	-25.3	-25.7	-15.4	-14.6	-12.4	-14.8	-12.6	-14.7	-13.1	-4.2	-11.1	-16.1
jh		45.2	115.0	110.8	109.8	113.0	-4.1	-4.2	-4.3	-11.4	-4.1	-4.5	-4.1	0.4	-4.5	-6.6
k		-16.3	6.4	4.5	3.5	4.7	-3.3	-6.8	-5.1	-3.4	-3.9	-6.7	-5.6	3.2	-6.5	-11.4
kd		199.4		248.2		248.2	-1.3	-28.5	-1.3	-3.1	1.7	-3.9	-1.1	-1.2	-0.1	-1.2
l		52.7	89.9	88.1	72.2	87.8	-8.4	-9.7	-9.4	-12.1	-9.1	-9.3	-8.6	1.7	-10.1	-14.2
m		29.3	-8.2	-9.6	-12.0	-8.6	-7.8	-8.5	-8.8	-12.3	-7.8	-8.2	-9.9	3.2	-11.0	-15.3
n	-7.3	31.9	-0.3	-0.7	-1.4	-0.9	-7.3	-8.8	-7.3	-9.6	-7.1	-7.4	-7.7	2.0	-9.2	-12.4
ng		-3.9		10.6		10.6	-11.8	-12.8	-11.8	-16.3	-11.8	-11.7	-11.8		-11.8	-14.1
ow		-82.1	-46.7	-48.4	-43.7	-48.3	-18.8	-20.3	-17.8	-20.5	-20.8	-22.3	-18.8	-3.1	-15.7	-18.7
oy		26.3		67.3	67.3	61.9	-8.7	7.1	-13.6	3.2	-16.8	-17.7	-17.8	8.6	-17.3	-17.0
p	-6.6	-31.3	-10.5	-9.6	-7.5	-11.5	-5.5	-9.1	-6.0	-4.8	6.5	-4.3	-8.5	0.8	-6.3	-9.2
r	-7.6	42.1	54.2	53.6	60.8	51.3	-6.8	-7.2	-9.0	-10.0	-6.7	-7.1	-7.3	3.2	-9.9	-12.1
s		17.5	17.3	16.9	16.4	15.9	-6.1	-7.0	-6.3	-11.9	-5.8	-6.3	-5.6	1.6	-7.7	-14.7
sh		38.1	27.0	26.2	26.6	25.4	-8.5	-11.5	-10.8	2.1	-9.9	-11.9	-9.5	2.7	-11.2	-12.0
t	-5.0	-38.4	-29.9	-31.3	-31.7	-30.6	-3.8	-7.8	-4.3	-3.5	-5.2	-3.5	-7.8	1.7	-5.6	-6.7
td	-1.7	63.5		91.2		91.1	-1.6	-3.6	-1.5	-3.5	-2.0	-3.0	-2.2	-0.1	-1.5	-3.8
th		38.2		17.4	16.5	16.1	-0.6	-8.9	-2.6	-2.6	-3.6	-4.0	-3.2	3.9	-4.5	-7.2
ts		-16.8		-46.4		-46.2	-2.2	-4.0	-2.1	-7.3	-6.6	-3.3	-3.0	2.5	-4.7	-7.1
uh			14.5	7.8	7.8	7.8	-3.0	-8.6	-6.7	-5.6	-12.3	-5.5	-3.2	2.4	-5.4	-5.4
uw	-8.1	-2.7	2.1	-0.3	3.7	-0.3	-6.8	-8.7	-7.9	-8.8	-6.5	-6.9	-6.9	-1.4	-6.0	-7.5

		CF	PF	PI	All	Onset	Coda	LL	RL	LS	RS	LN	RN	LF	RF	LV	RV
	v	-6.1	6.8		14.0	13.4	14.3	-3.9	-6.4	-4.9	-5.9	-5.5	-5.3	-4.4	2.1	-6.0	-8.7
t	w	-10.2		18.2	17.3	17.3	19.3	-8.3	-7.6	-8.3	-7.6	-9.3	-7.6	-9.7	0.5	-8.8	-7.6
	y			126.2	125.2	125.2	127.1	-13.6	-9.9	-9.4	-10.1	-9.4	-13.2	-9.7	-3.6	-10.0	-10.0
	z	-5.1	39.6		64.9	64.0	65.3	-4.2	-6.4	-4.9	-6.6	-4.6	-6.7	-4.4	0.5	-4.7	-11.5

Table 4.1: Results from the phone splitting experiments. The first column lists all the phones used in the experiments. The top row lists the available splits. The table entries correspond to the difference between the independent log likelihood and the weighted average. A negative table entry indicated that the phone should be split into allophones. A positive table entry indicated that the phone should not be split. A blank entry indicates that there was not enough information to determine whether or not a phone sould be split.

for most vowels, but few consonants.

Because the majority of phone models tend to improve on any contextual split, it is impossible to state which contextual split is the best overall. It is not so difficult, however, to determine which split for any individual phone causes the greatest improvement in the model. Improvement can be measured by the negativity of the number in the column. A larger negative number indicates a greater improvement for a given split. When a comparison is made this way, there is a strong tendency for phones to favor splitting based on the context of the following phone.

By examining table 4.1, it can be seen that prosodic splits differ from contextual splits in two ways. First, there are fewer prosodic splits that allow for model improvement. Second, when there is an improvement due to prosodic splitting, the differences in likelihoods resulting from the prosodic splits are, in general, much more negative than the differences from contextual splits.

Different prosodic splits tend to favor certain groups of phones over others. Vowels, for example, show improvement for every defined prosodic split. The only exceptions are the vowels /aa/, /ao/, /ax/, /oy/ and /uh/. The phones /aa/ and /ao/ tend to only favor function-content splitting while the schwa, /ax/ favors both CF splitting and phrase final splitting.. The diphthong /oy/ and the vowel /uh/ tend not to favor any splits at all. Of all the consonants, the effects of prosodic splitting are most significant for plosives and nasals.

	% Correct	Accuracy
Independent	67.31	63.75
Dependent	77.16	74.18

Table 4.2: Recognition results for the prosody independent and prosody dependent recognizers.

The majority of plosives models are improved by phrase final splitting. Phrase initial and accent splits show improvements over prosody independent models for about half of the plosive consonants. Nasals appear to be improved by either phrase final splitting or phrase initial and accent splitting. Liquids and glides favor no prosodic splits with the exception of the content-function phonetic distinction. Fricatives are similar to liquids in the respect that they also only favor the CF distinction. The only two exceptions are the phones /ch/ and /dh/, which seem to favor every prosodic split.

4.2 Fixed Parameter Recognizer

Recognition results for the fixed parameter recognizers are shown in table 4.2. As seen in the table, when prosodic factors are used as model parameters, the number of words correctly identified by the recognizer increases by 9.85%. The accuracy of the prosody based word recognizer is greater than that of the prosody independent recognizer by 10.83%.

The main effect of incorporating prosody into the recognizer is that the number of substitution errors is drastically reduced. Substitutions are reduced by 35% between the prosody independent and prosody dependent recognizers. Insertions and deletions made by the prosody dependent recognizer are also reduced by 27% and 15% respectively from the number of insertions and deletions made by the independent recognizer.

	% Accuracy
Speech	97.65
Sonorant	95.35
Continuant	90.70
Syllabic	85.83

Table 4.3: Results from the first set of SVM binary classifiers described in section 3.2.3

	% Accuracy
-+consonantal	92.29
+-consonantal	87.38
-+sonorant	100.00
+-sonorant	94.03
+consonantal/-+continuant	95.37
+consonantal/+-continuant	95.09

Table 4.4: Results from the second set of SVM binary classifiers described in section 3.2.3

4.3 Manner Feature Detection

SVM feature classifiers are able to recognize binary distinctions of landmarks with high accuracy.

SVM feature classifiers are able to distinguish between transitions going to positive feature valued events and transitions going to negative feature valued events. Table 4.3 shows the results for this set of classifiers trained for *sonorant*, *continuant* and *syllabic* features to provide the positive vs negative transitional distinction. The SVM is able to detect *sonorant* and *continuant* features with an accuracy of over 90%. Because there are usually no distinct boundaries between vowels and semi-vowels, the detection rate for *syllabic* features is slightly lower.

When given more specific transition constraints as described in section 3.2.3, SVM classifiers are still able to accurately distinguish transitions, as shown in table 4.4.

Chapter 5

Future Work and Conclusion

5.1 Future Work

The prosody of read speech and the prosody of conversational speech are different. Radio News, the corpus used for the prosody recognition experiments, consists of speech read by professional radio announcers. Conversational speech, in general, is much more difficult to recognize than read speech. Prosody has a positive effect on recognition rates for read speech; how will it affect recognition rates for conversational speech?

To determine this, a set of experiments similar to those presented in this thesis will need to be run for a corpus containing conversationsl speech. The Switchboard [26] corpus contains recorded telephone conversations between pairs of strangers. The conversations are spontaneous; however, topic of conversation was pre-assigned. Switchboard does not contain any transcriptions with prosodic labels. Prosodic labeling will therefore have to be done for a subset of switchboard transcriptions. Prosodic labels will be generated for a subset of Switchboard, UI03, using an automatic parser [27]. UI03 is a database created specifically for prosody study by parsing the word transcriptions from Switchboard to find all segments that contain ten seconds or more of continuous dialog with no disfluencies. These ten second segments were then saved as their own separate transcriptions and corresponding sound files were chopped from the original Switchboard sound files. Experiments on the UI03 database will follow the procedure described in sections 3.1.4 and 3.1.5.

Landmark detection using SVMs has proven to be highly accurate. Should landmarks be prosody dependent? For example, will the detection of a [+-consonantal] transition improve if the SVM knows whether or not it is looking for a [+accent] or a [-accent] transition. On the other hand, can prosody be classified based on pitch, timing and MFCC's at [+-consonantal], [+-syllabic] and [+-sonorant] sequences of landmarks? This experiment will be performed on the Radio News Corpus described in section 3.1.2. The procedure will be similar to the procedure described in section 3.2.3.

Landmark detection is highly accurate for SVM binary classifiers. If these SVM binary classifiers can be incorporated into a continuous speech recognizer, how much will recognition rates improve over those of a standard HMM recognizer? A recognizer of this nature would first find the landmark and then classify it based on place of articulation and voicing information. This information would then be given to an HMM for further processing and recognition.

5.2 Conclusion

Prosody and co-articulation both affect the way different phones sound. Both prosodic and phonetic context information can be included in the phoneme model. When included, this information causes the log likelihood of the model to improve. Prosodic information is more useful to the model than is phonetic context information for most phonemes. This has been shown both in experiments that compare the log likelihoods of phonetic context-dependent models to those of prosodic context-dependent models and also in experiments that included both prosodic and phonetic context models in two different speech recognizers.

Support vector machine binary classifiers are able to detect landmarks with high accuracy. Because each phoneme can be represented by a unique set of distinctive features, the ability to identify transitions between different features correctly has the potential to provide for accurate phone and word recognition.

The research presented in this thesis has shown that incorporating prosody into the phoneme model not only improves the model, but recognition accuracy as well. This research has also shown that landmark detection of manner features has high accuracy and is useful for phone classification.

References

[1] L. Liu, Z. Li, B. Shi, *Segment matrix vector quantization and fuzzy logic for isolated-word speech recognition.* Proceedings, 25th International Symposium, May 1995, pp: 152-156.

[2] A. Samouelian, *Isolated voiced digit recognition using inductive inference.* TENCON '96. Proceedings, IEEE TENCON. Digital Signal Processing Applications , Volume: 1, Nov. 1996, pp: 119-124.

[3] Z. Li-Peng, L. Li-Mei, C. Chang-Nian, *Speech recognition using dynamic recognition neural network.* TENCON '93. IEEE Region 10 Conference Proceedings. Computer, Communication, Control and Power Engineering, Oct 1993, pp: 333-336.

[4] F. Liu, M. Picheny, P. Srinivasa, M. Monkowski, J. Chen, *Speech recognition on Mandarin Call Home: a large-vocabulary, conversational, and telephone speech corpus.* IEEE International Conference on Acoustics, Speech, and Signal Processing, May 1996, pp: 157-160.

[5] P. Eide, E. Chaudhari, U. McDonough, J. Ng, K. Siu, M. Gish, H. Jeanrenaud, *Reducing word error rate on conversational speech from the Switchboard corpus.* International Conference on Acoustics, Speech, and Signal Processing, May 1995, pp: 53-56.

[6] K. De Jong, *The supraglottal articulation of prominence in English: Linguistic stress as localized hyperarticulation.* Journal of the Acoustical Society of America, 1995. vol.97(1), pp: 491-504.

[7] C. W. Wightman, S. Shattuck-Hufnagel, M. Ostendorf, P. J. Price, *Segmental durations in the vicinity of prosodic phrase boundaries*. Journal of the Acoustical Society of America, 1992. vol. 91(3), pp: 1707-17.

[8] P. Fougeron, P. Keating, *Articulatory strengthening at the edges of prosodic domains*. Journal of the Acoustical society of America, 1997. vol 101(6), pp: 3728-3740.

[9] J. Edwards, M. Beckman, J. Fletcher. *The articulatory kinematics of final lengthening*. Journal of the Acoustical Society of America, 1991. vol 89(1), pp: 369-382.

[10] T. Cho, *Effects of Prosody on Articulation in English*. Ph.D. dissertation, UCLA, 2001.

[11] C. W. Wightman, M. Ostendorf, *Automatic recognition of prosodic phrases*. International Conference on Acoustics, speech, and signal Processing, April 1991, pp: 321-324.

[12] C. W. Wightman, M. Ostendorf, *Automatic recognition of intonational features*. IEEE International Conference on Acoustics, Speech, and Signal Processing, March 1992, pp: 221-224.

[13] S. Keyser, N. Stevens, *Feature geometry and the vocal tract*. Journal of Phonetics, 1999.

[14] K. Stevens, *Relational properties as perceptual correlates of phonetic features*. International Conference of Phonetic Sciences, 1987. 352-355

[15] K. Stevens, S. Manual, S. Shattuck-Hufragel, S. Liu, *Implementation of a model for lexical access based on features* International Conference on Spoken Language Processing, 1992.

[16] P. Niyogi, C. Burges, *Detecting and implementing acoustic features by support vector machines*. University of Chicago Tech Report TR-2002-02.

[17] A. Juneja, *Speech recognition using acoustic landmarks and binary phonetic feature classifiers*. PhD. Thesis Proposal, University of Maryland, 2003.

[18] R. L. Rabiner, B. H. Juang, *An introduction to hidden Markov models*. Acoustics, Speech, and Signal Processing Magazine, January 1986, pp 4-15.

[19] C. Burges. *A tutorial on support vector machines for pattern recognition*. Data Mining and Knowledge Discovery, Vol. 2, No. 2, 1998, pp 1-47.

[20] M. Beckman, G. Ayers, *Guidelines for ToBI labeling : the very experimental HTML version*. http:/www.ling.ohio-state.edu/research/phonetics/E_ToBI/singer_tobi.html

[21] M. Ostendorf, P. J. Price, S. Shattuck-Hufnagel, *The Boston University radio news corpus*. Linguistic Data Consortium, Philadelphia, PA. 1995.

[22] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, P. Woodland, *The HTK book*. Cambridge University Engineering Department, http://htk.eng.cam.ac.uk/, 2002.

[23] K. Lee, H. Hon, R. Reddy, *An overview of the SPHINX speech recognition system*. IEEE Transactions on Acoustics, Speech, and Signal Processing, 38(1). January 1990.

[24] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, *The DARPA TIMIT acoustic phonetic speech corpus*, NIST, 1993.

[25] C. Chang and C. Lin, *LIBSVM : a library for support vector machines*, 2001. Software available at http:/www.csie.ntu.edu.tw/c̃jlin/libsvm

[26] J. J. Godfrey, E. C. Holliman, and J. McDaniel, *SWITCHBOARD: Telephone speech corpus for research and development*, IEEE International Conference on Acoustics, Speech and Signal Processing, 1992, pp. 517-520.

[27] A. Cohen, MS thesis, in preparation.