AUTOREGRESSIVE HIDDEN MARKOV MODELS AND THE SPEECH SIGNAL

BY

JACOB DANIEL BRYAN

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Adviser:

Professor Stephen E. Levinson

ABSTRACT

This thesis introduces an autoregressive hidden Markov model (HMM) and demonstrates its application to the speech signal. This new variant of the HMM is built upon the mathematical structure of the HMM and linear prediction analysis of speech signals. By incorporating these two methods into one inference algorithm, linguistic structures are inferred from a given set of speech data. These results extend historic experiments in which the HMM is used to infer linguistic information from text-based information and from the speech signal directly. Given the added robustness of this new model, the autoregressive HMM is suggested as a starting point for unsupervised learning of speech recognition and synthesis in pursuit of modeling the process of language acquisition.

To Ann, for her love and support.

ACKNOWLEDGMENTS

I would like to express my deep gratitude to California State University, Fresno, and the professors who took the time to provide me with the challenges and opportunities that have set me on my current path, namely Dr. Youngwook Kim, Dr. Gregory Kriehn and Dr. Chulho Won. I would like to thank my graduate advisor Dr. Stephen E. Levinson for his support, guidance, and encouragement in my work on this project. I would also like to thank the Beckman Institute and the Italian Institute of Technology for their support. Lastly, I would like to thank my colleagues in the Language Acquisition and Robotics Group (LARG) for stimulating my interest in language acquisition and machine intelligence.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Language Acquisition

Much of this work is motivated by the problem of developing an artificially intelligent system. Thus far, the human species is the most comprehensive example of intelligence and so the process of developing an intelligent system often involves attempting to imitate some human capability in the spirit of Alan Turing's *imitation game* [1]. Ideally, one would be able to directly construct a model of the human brain as a means of developing a system with intelligence comparable to that of humans. In practice, however, the level of complexity of the human nervous system carries a prohibitive computational cost. An alternative to direct brain modeling is to approach the problem from the perspective of functional equivalence. More specifically, if a model of intelligence exhibits behavior indistinguishable from that of a human, then the goal has been reached.

While there are many important characteristics to human intelligence, among the most ubiquitous is language. In addition to this observation, it should be noted that humans are not born with a language nor must they be explicitly trained in basic aspects of their native language. Instead, children perform a process of automatic language acquisition in the first years of life. Furthermore, a person's native language strongly affects the structure of their internal model of the world in which they exist. These observations indicate that great insights into the nature of intelligence might be made by understanding and modeling the process of language acquisition. The work in this thesis is done in pursuit of developing a system that is able to automatically acquire language in a manner similar to that of a human child. In particular, we will begin this process at the level of sensory signals and attempt to utilize insights about linguistic structure in order to make incremental steps toward language.

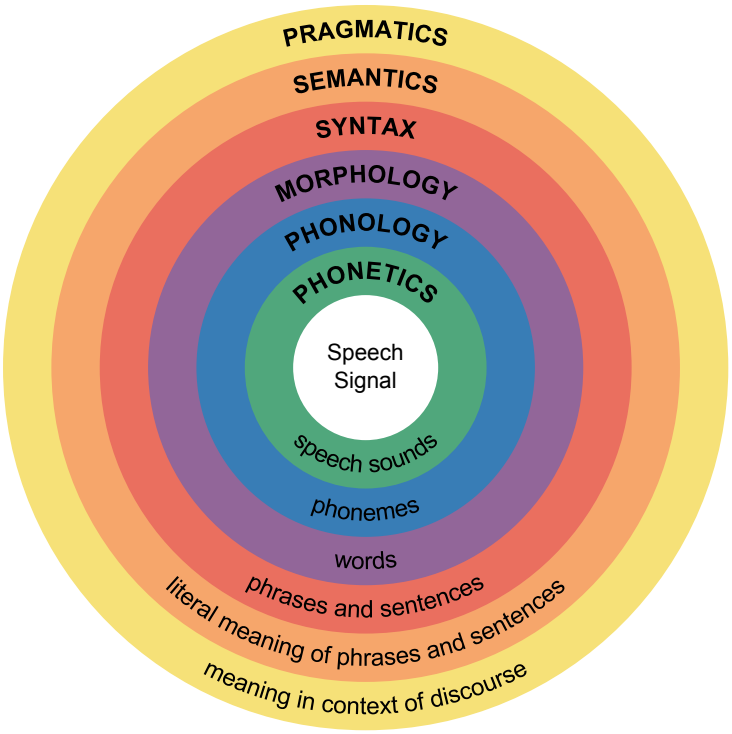

Figure 1.1: Hierarchy of linguistic structure in the speech signal.

1.2 Language and Embodiment

Generally speaking, linguistic structure defines a framework within which symbols may be related to one another. The symbols within this framework may correspond to objects in the physical world, actions, thoughts, or more abstract concepts. If language is to be acquired, these symbols must be related to sensory information and motor function, and the process by which this relationship is established is of particular importance. In order to develop a system capable of establishing such a symbolic relationship with sensorimotor information, self-organizing and adaptive models must be incorporated.

One approach to modeling this process of language acquisition is to consider the process in a hierarchical manner. An illustration of the hierarchical linguistic structure for the speech signal is shown in Figure 1.1. In addition to this, a similar hierarchy is used for each of the other sensory motor channels. As language is acquired, certain pieces of sensory and motor information are then related to one another at different levels of the hierarchy.

1.3 Speech Processing

In order to gain insight into how symbolic representations of the physical world might be established within this linguistic framework, we will first narrow the scope of the problem to one mode of sensory input and begin at the signal level. In particular, we will focus on the speech signal and consider a method by which information at higher levels of the linguistic hierarchy may be inferred, prior to the existence of working vocabulary. We now pursue the development of such a method by first considering capabilities of human children in the early stages of language acquisition.

In 1993, Bijeljac-Babic, Bertoncini, and Mehler conducted a study on how infants categorize multisyllabic phrases [2]. Among other implications, this study indicates that infants are capable of differentiating the number syllables from the speech signal within four days of birth. This indicates that a model of language acquisition should similarly be capable of determining information such as syllable or word boundaries without a working vocabulary of the language. While the energy envelope of the speech signal may be utilized in this process of detecting word and syllable boundaries, it does not generally provide enough information to perform these detections with high fidelity. Alternatively, the linguistic structure that is inherent in the speech signal offers a means by which this fundamental step might be accomplished.

1.4 Approach

In general, the task of inferring linguistic structure from speech data may be broken into two parts. The first is the issue of characterizing the speech signal at any given point in time. Specifically, we must use a signal model that can sufficiently characterize the speech signal and distinguish between the various speech sounds while remaining computationally tractable. In order to strike this balance, an all-pole filter model of the speech signal is used to accomplish this task.

The second issue is the task of characterizing the low level linguistic information within the speech signal. In terms of the linguistic hierarchy, we seek to infer phonetics and phonotactics within the signal. Given its proven utility in speech recognition and synthesis, the hidden Markov model (HMM) is a very good candidate for such a task. In addition, we will see that the HMM can be used to

characterize linguistic information.

Based on these two mathematical tools, we will utilize the all-pole filter model as a method of characterizing the observations of the HMM, resulting in a new linearly predictive HMM. Using this model, the filter parameters and HMM parameters may be inferred simultaneously using a modified Baum-Welch algorithm applied to the speech signal. The inferred parameters, as we will find, directly relate to coarse phonetic and phonotactic structures within the speech signal. Ultimately, an approach such as this represents the first fundamental steps toward unsupervised speech recognition and ultimately language acquisition.

This thesis details the development the linearly predictive HMM, the corresponding Baum-Welch algorithm and the mathematical tools used to develop this model. Chapter 2 explores the speech signal from a signal processing perspective and develops the mathematical justification for using an all-pole filter model of the signal known as an autoregressive model. Chapter 3 details the basic structure of the hidden Markov model, the Baum-Welch algorithm, and experimental results that indicate the relationship between linguistic structure and language. Chapter 4 is a derivation of a modified version of the Baum-Welch algorithm such that the algorithm incorporates the filter model into the HMM parameters. Chapter 5 details experimental results achieved by applying this new autoregressive HMM to several sets of speech data. Lastly, Chapter 6 outlines useful conclusions that may be drawn from this work as well as directions for future research.

CHAPTER 2

THE SPEECH SIGNAL

Given the time-varying nature of the speech signal and its spectral characteristics, it is often used as a canonical example of a nonstationary signal. In order to represent the information carried by the signal, we must develop a model that can represent the spectral information at a given moment in time and its variations over time.

2.1 The Source-Filter Model of Speech

Let us consider an idealized model of the vocal apparatus as a time-varying signal source followed by a time-varying filter as shown in Figure 2.1. In this model, the signal source corresponds to the processes by which vibrations are generated within the vocal apparatus. In general, the vibrations generated with the vocal tract may be idealized as one of three excitation signals: a periodic impulse train, a single impulse or white noise. In addition, the vocal tract itself has resonant frequencies that are dependent on the physical properties of the tissues of the vocal tract and the shape of the vocal tract at a given instant in time. These resonances can then be naturally represented using a time-varying all-pole filter [3].

Given that the excitation signal and resonances of the vocal tract vary regularly with time, the speech signal is definitively non-stationary. Furthermore, these variations of the frequency content of the vocal tract are the means by which information is carried within the speech signal. Fortunately the vocal tract is only capable of moving or re-configuring at a much slower speed than the frequencies found in the speech signal, which allows the signal to be treated as stationary over a sufficiently short period of time. Using this short-time stationary assumption, the speech signal can be segmented into analysis windows over which it may be treated as stationary. In practice, the speech signal may be treated as short-time stationary for analysis windows of length 10 ms to 30 ms. The drawback

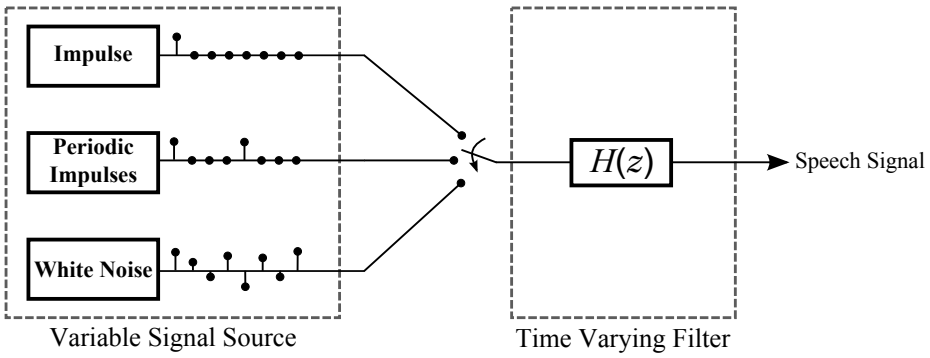

Figure 2.1: The source filter model of the human vocal apparatus.

to this assumption is that decreasing the length of the analysis window increases the time-resolution in our analysis, while increasing the length of the window increases frequency resolution. In order to maximize the trade-off between time and frequency resolution under this constraint, the analysis windows are allowed to overlap, which improves the time resolution without negatively affecting frequency resolution.

2.2 All-Pole Filter Model

As stated previously, the frequency response of the vocal tract is naturally modeled by an all-pole filter. It should be noted that, to account for the nasal cavity, the filter model of the vocal tract often includes a single zero outside of the unit circle. Fortunately, a zero outside the unit circle is equivalent to an infinite number of poles within the unit circle, which is a basic property of the geometric series. Based on this, we can then approximate the zero in the filter by the addition of an arbitrarily large number of poles as represented by the p^{th} order all-pole filter in Equation 2.1.

$$H(z) = \frac{A}{1 - \sum_{k=1}^{p} \alpha_k z^{-k}} \tag{2.1}$$

Although the nature of the excitation signal generated within the vocal tract does have an effect on the spectral shape of the speech signal, a time history of the filter parameters is generally sufficient for the purposes of speech recognition [4]. Based on this precedent, this work will focus on characterizing the nature of the filter in our source filter model.

Consider the p^{th} order all-pole filter, as shown in Equation 2.1, corresponding to

the vocal tract configuration during a short-time stationary segment of the speech signal. In this filter equation, the gain of the signal is denoted by A, and the coefficients α_k are referred to as linear prediction coefficients (LPCs). In general, the inference of these filter parameters is referred to as linear prediction analysis. In order to infer the LPCs and signal gain, we must consider the time domain representation of the signal. Let us denote the excitation signal by $u[n]$ and the impulse of response of the all-pole filter by $h[n]$. Then we can express the speech signal $s[n]$ using Equation 2.2.

$$s[n] = h[n] * u[n] = \sum_{k=1}^{p} \alpha_k s[n-k] + Au[n] \tag{2.2}$$

It should be noted that this time domain expression of the speech signal indicates that the signal at time index n is dependent on a linear combination of the previous p values of the signal. If the excitation signal corresponding to voiced speech is idealized as a periodic impulse train or a single impulse, the excitation signal is identically zero for the majority of the period. Applying this observation to Equation 2.2 we find that the speech signal is dependent only on its past values for the majority of the time. Due to this property, this model of the speech signal is also known as the autoregressive model of speech.

Based on this model of the speech signal, we then define the linear predictor of the speech signal $\hat{s}[n]$ as defined in Equation 2.3.

$$\hat{s}[n] = \sum_{k=1}^{p} \alpha_k s[n-k] \tag{2.3}$$

Using this model of the speech signal, we will then examine two methods of estimating the linear prediction coefficients α_k for a given short-time stationary segment of speech.

2.2.1 Covariance Method of Linear Prediction

We first study a direct method of approximating the LPCs by minimizing the approximation error via the projection theorem. Suppose that the short time stationary assumption holds for a speech signal segment of length $N + p + 1$. Using the linear predictor as an approximation of the speech signal, we define the estimation error by Equation 2.4. We can then seek to determine the LPC values (α_k)

that minimize squared error over the length of the stationary window as shown in Equation 2.5.

$$e[n] = s[n] - \hat{s}[n] = s[n] - \sum_{k=1}^{p} \alpha_k s[n-k] \tag{2.4}$$

$$\min_{\alpha_k \in \mathbb{R}} \sum_{n=p}^{N+1} \left(e[n]\right)^2 = \min_{\alpha_k \in \mathbb{R}} \sum_{n=p}^{N+1} \left(s[n] - \sum_{k=1}^{p} \alpha_k s[n-k] \right)^2 \tag{2.5}$$

Alternatively, this minimization problem may be formulated as a vector equation in $l_2(\mathbb{R})$ where the norm of the error vector is minimized. We define each vector of length $N + p + 1$ as shown in Equation 2.6 where the segment under analysis begins at time index t. We many then apply the Project Theorem for Hilbert spaces to this minimization problem.

$$\mathbf{s}_t - \mathbf{S}_t \boldsymbol{\alpha} = \mathbf{e}_t \tag{2.6a}$$

$$\mathbf{s}_t = \begin{bmatrix} s[t+p] & s[t+p+1] & \cdots & s[t+N] \end{bmatrix}^\top \tag{2.6b}$$

$$\mathbf{S}_t = \begin{bmatrix} s[t+p-1] & s[t+p-2] & \cdots & s[t+0] \\ s[t+p] & s[t+p-1] & \cdots & s[t+1] \\ \vdots & & \ddots & \vdots \\ s[t+N] & \cdots & \cdots & s[t+N-p] \end{bmatrix} \tag{2.6c}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_p \end{bmatrix}^\top \tag{2.6d}$$

$$\mathbf{e}_t = \begin{bmatrix} e[t+0] & e[t+1] & \cdots & e[t+N-p] \end{bmatrix}^\top \tag{2.6e}$$

Let $\mathcal{S}$ be the subspace generated by the columns of $\mathbf{S}_t$ and note this space is the set of all linear predictors. Then we have that $\hat{\mathbf{s}}_t = \mathbf{S}_t \boldsymbol{\alpha} \in \mathcal{S}$. The Projection Theorem states that error has minimum norm if and only if the error is orthogonal to $\mathcal{S}$. Since $\mathcal{S} = \text{span}\{\mathbf{S}_t\}$, the orthogonality condition may be restated as $\mathbf{S}_t^T \mathbf{e}_t = 0$. By then substituting the expression for the error from Equation 2.4, we reach Equation 2.7 which can be readily solved for $\boldsymbol{\alpha}$. In general, linear prediction analysis is then conducted over a set of times t, and the history of LPC vectors ($\boldsymbol{\alpha}$) is then collected and used for various applications such as speech synthesis or speech recognition.

$$\mathbf{S}_t^T \mathbf{S}_t \boldsymbol{\alpha} = -\mathbf{S}_t^T \mathbf{s}_t \tag{2.7}$$

This approach to estimating the LPCs is generally known as the covariance method of linear prediction due to the proportional relationship between $\mathbf{S}_t^T \mathbf{S}_t$ and the sample covariance of $s[n]$. In general, the covariance method of linear prediction produces an optimal estimate of the LPCs provided that the signal under analysis was indeed generated by an autoregressive model. If the signal under analysis is not autoregressive, however, the covariance method carries no guarantee that the estimated filter parameters will correspond to a stable filter.

2.2.2 Autocorrelation Method of Linear Prediction

Now, let us consider an alternative method of linear prediction known as the autocorrelation method. In this approach, we will utilize a windowed segment of the speech signal rather than extracting points from the signal directly as is done in the covariance method. Let us define the window function and windowed segment of the signal as follows, given that $s[n]$ represents the speech signal as before. Note that a square window is defined here for the purpose of mathematical simplicity, but in general a tapered window such as a Hamming window is used in order to minimize the error generated by data points near the edges of the window [3].

$$w[n] = \begin{cases} 1 & \text{if } n \in \{0, 1, \ldots, N\} \\ 0 & \text{else} \end{cases}$$

$$s_t[n] = s[t+n]w[n]$$

Under this definition, we have that $s_t[n]$ is equal to the segment of speech signal of length $N+1$ beginning at time t and is identically zero for $n < 0$ or $n > N$. By applying the same analysis as is done with the covariance method, we then arrive at a set of normal equations similar to those shown in Equation 2.7. At this point, we can expand the normal equations into a summation format and rewrite them as shown in Equation 2.8a using the function Φ_t as defined in Equation 2.8b.

$$\sum_{k=1}^{p} \alpha_k \Phi_t[i,k] = \Phi_t[i,0] \quad \forall i \in \{1, 2, \ldots, p\} \tag{2.8a}$$

$$\Phi_t[i,k] = \sum_{n=0}^{N+P} s_t[n-i]s_t[m-k] \tag{2.8b}$$

We can then take advantage of the fact that $s_t[n] = 0$ outside the bounds of the analysis window to simplify the expression of Φ_t.

$$\Phi_t[i,k] = \sum_{n=0}^{N+P} s_t[n-i]s_t[m-k] = \sum_{n=i}^{k+N} s_t[n-i]s_t[m-k]$$

Applying a change of variables $m = n - i$, we have

$$\Phi_t[i,k] = \sum_{m=0}^{N+k-i} s_t[m]s_t[m+i-k] = r_t[i-k].$$

Here we replace Φ_t with $r_t[\tau]$, which is the autocorrelation function of the windowed segment of the speech signal at time t. Substituting this back into the normal equations, we arrive at a new set of matrix equations that can be solved for $\boldsymbol{\alpha}$, known as the Yule-Walker equations.

$$\mathbf{R}_t\boldsymbol{\alpha} = \mathbf{r}_j \tag{2.9a}$$

$$\mathbf{R}_t = \begin{bmatrix} r_t[0] & r_t[1] & \cdots & r_t[p-1] \\ r_t[1] & r_t[0] & \cdots & r_t[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ r_t[p-1] & r_t[p-2] & \cdots & r_t[0] \end{bmatrix} \tag{2.9b}$$

$$\mathbf{r}_t = \begin{bmatrix} r_t[1] & r_t[2] & \cdots & r_t[p] \end{bmatrix}^\top \tag{2.9c}$$

It is important to note that $\mathbf{R}_t$ is a Toeplitz matrix, which allows the solution to be computed efficiently using the Levinson-Durbin recursion. In addition, since this method of linear prediction is performed on a windowed version of the speech signal, i.e. the signal goes to zero beyond the boundaries of the analysis window, the estimation error is always higher than the covariance method. Fortunately, the Toeplitz matrix found in the normal equations guarantees that the poles of the filter model will fall within the unit circle. In addition, the error induced by the autocorrelation method can be made arbitrarily small by simply increasing the number of poles in the filter.

CHAPTER 3

THE HIDDEN MARKOV MODEL

In this chapter, we will first examine the mathematical structure of a simple case of the hidden Markov model wherein the model is comprised of finitely many discrete random variables. We will then examine the Baum-Welch algorithm as a means of determining the HMM parameters associated with a given sequence of observations. Using the Baum-Welch algorithm, an historic experiment conducted by Cave and Neuwirth [5] is reproduced and analyzed. The results of this experiment illustrate the close relationship between the HMM and linguistic structure.

3.1 The Hidden Markov Model

The HMM is a mathematical model used to characterize random processes wherein an observable sequence is dependent on an unobserved Markov chain. The Markov chain is comprised of sequence of states drawn from a finitely countable set $\mathcal{Q} = \{q_j : 0 \leq j \leq N\}$ where each q_j is an individual state. This Markov chain makes up the state sequence $S = (s_t)_{t=0}^{T}$, where $s_t \in \mathcal{Q}$ for all t. Each element of the observation sequence $\mathcal{O} = (O_t)_{t=0}^{T}$ is then generated by the corresponding state at time t. An illustration of the structure of this model is given by Figure 3.1. In the simplest case, every random variable O_t is discrete and shares the same finite range given by $\mathcal{V} = \{v_k \text{ for } k = 1, 2, 3, \ldots, M\}$. The set $\mathcal{V}$ is commonly referred to as the *vocabulary* of the HMM.

Based on this model structure, an HMM can be characterized using a set of probabilities which will be denoted in vector and matrix form. First we note that each element of the observation sequence is such that $O_t = v_i$ for some $v_i \in \mathcal{V}$ and each element of the state sequence is such that $s_t = q_i$ for some $q_i \in \mathcal{Q}$. We then define the initial state vector, $\boldsymbol{\pi} = [\pi_1, \pi_2, ...\pi_N]$, which corresponds to the probability distribution of O_1. Next we define the conditional state transition

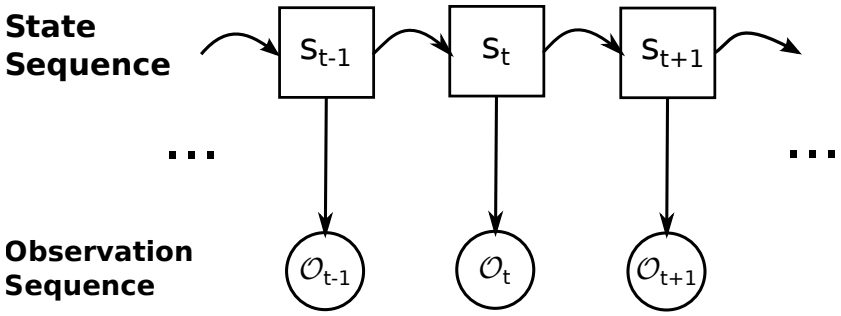

Figure 3.1: Observation sequence, $\mathcal{O}_t$, generated by an HMM.

probability $a_{ij} = P(s_{t+1} = q_j | s_t = q_i)$ and state transition matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$. Lastly we define the conditional probability of a given observation $b_{jk} = P(O_t = v_k | s_t = q_j)$ and the observable process matrix $\mathbf{B} = [b_{jk}] \in \mathbb{R}^{N \times N}$. We will now consider the task of estimating the set of model parameters $\boldsymbol{\lambda} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$ using a given sequence of observations.

3.2 Forward-Backward Algorithm

Consider the probability of a sequence of observations given the model parameters, as determined by Equation 3.1. In this equation, $\mathcal{S}$ is the set of all possible state sequences and the function $b_j(O_t)$ is defined in Equation 3.2.

$$P(\mathbf{O}|\boldsymbol{\lambda}) = \sum_{S \in \mathcal{S}} \left(\pi_{s_0} b_{s_1}(O_1) \prod_{t=1}^{T-1} a_{s_t, s_{t+1}} b_{s_{t+1}}(O_{t+1}) \right) \tag{3.1}$$

$$b_j(O_t) = \begin{cases} b_{jk} & \text{if } O_t = v_k \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

It is important to note that the length of this computation is exponential with the length of the observation sequence. This computation may be computed more efficiently by means of the forward-backward algorithm, which allows the probability to be computed on a timescale that is linear with the length of the observation sequence.

First, we define the "forward probability" as the joint probability of the given

observation sequence up until time t and $s_t = q_i$, given a set of model parameters.

$$\alpha_t(i) = P(O_1, O_2, ...O_t, s_t = q_i|\boldsymbol{\lambda})$$

The forward probabilities are initialized as $\alpha_1(j) = \pi_j b_j(O_t)$ and each subsequent probability is computed according to Equation 3.3.

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)a_{ij}\right] b_j(O_t), \quad 1 \leq t \leq T-1 \tag{3.3}$$

Now we define the "backward probability" as the probability of the sequence of observations after time t given that $s_t = q_j$.

$$\beta_t(j) = P(O_{t+1}, O_{t+2}, \ldots, O_T|s_t = q_j, \boldsymbol{\lambda}) \text{ for } 1 \leq j \leq N$$

Similarly, the backward probabilities are initialized as $\beta_T(j) = 1$ and computed recursively using Equation 3.4.

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}b_j(O_t)\beta_{t+1}(j), \quad T-1 \geq t \geq 1 \tag{3.4}$$

With these probabilities, we then note the following identity.

$$\begin{aligned}
\alpha_t(j)\beta_t(j) &= P(O_1, \ldots, O_t, s_t = q_j|\boldsymbol{\lambda}) \cdot P(O_{t+1}, \ldots, O_T|s_t = q_j\boldsymbol{\lambda}) \\
&= P(O_1, \ldots, O_T, s_t = q_j|\boldsymbol{\lambda}) \\
&= P(\mathcal{O}, s_t = q_j|\boldsymbol{\lambda})
\end{aligned}$$

Using this identity, we can then compute the probability of the observation sequence efficiently using any of the forms found in Equation 3.5.

$$\begin{aligned}
P(\mathcal{O}|\boldsymbol{\lambda}) &= \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j) \\
&= \sum_{j=1}^{N} \alpha_t(j)\beta_t(j) \\
&= \sum_{j=1}^{N} \alpha_T(j)
\end{aligned} \tag{3.5}$$

3.3 Baum-Welch Algorithm

Using Equation 3.5, the Baum-Welch algorithm can be used to recursively estimate $\boldsymbol{\lambda}$, the parameters of the HMM. Since the parameter matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{pi}$ are mutually disjoint, the parameters of each matrix may be estimated using separate reestimation formulas. The reestimation formulas are shown in Equation 3.6 where the new estimate of each parameter is denoted with an overline. Similarly, we denote the set of new parameters by $\overline{\boldsymbol{\lambda}} = \{\overline{\mathbf{A}}, \overline{\mathbf{B}}, \overline{\boldsymbol{\pi}}\}$.

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i)} \tag{3.6a}$$

$$\overline{b}_{jk} = \frac{\sum_{t \ni O_t = v_k} \alpha_t(j) \beta_t(j)}{\sum_{t=1}^{T-1} \alpha_t(j) \beta_t(j)} \tag{3.6b}$$

$$\overline{\pi}_i = \frac{\alpha_1(i) \beta_1(i)}{\sum_{j=1}^{N} \alpha_1(j) \beta_1(j)} \tag{3.6c}$$

Using these reestimation formulas, we have that $P(\mathbf{O}|\boldsymbol{\lambda}) \leq P(\mathbf{O}|\overline{\boldsymbol{\lambda}})$ where $\overline{\boldsymbol{\lambda}}$ is the set of newly estimated model parameters. Furthermore, equality holds if and only if $\boldsymbol{\lambda} = \overline{\boldsymbol{\lambda}}$, which provides a method of checking if the algorithm has converged on the optimal parameter values.

It can be shown that the Baum-Welch algorithm is a special case of the Expectation-Maximization algorithm. Alternatively, the proof of the Baum-Welch algorithm and its properties may be performed in a manner similar to the EM algorithm [6] or directly via optimization methods [7]. These proofs are not included in this thesis; however, the derivation via optimization methods is similar in nature to the derivation found in Chapter 4.

3.4 Practical Concerns

In order to implement the Baum-Welch algorithm, there are some practical concerns that must be addressed. Specifically, we must deal with the issue of representing very small probabilities in computer memory and the presence of zeros in the state transition matrix. The following sections outline methods of dealing with these issues when implementing the Baum-Welch algorithm.

3.4.1 Scaling

In the case of long observation sequences, it is possible that the values of the forward and backward probabilities will become so small that their floating point representation will "underflow," causing α_t to go to zero after some time t'. In order to keep the computations within the dynamic range of the computer, the scaling factor defined in Equation 3.7 is used to normalize the probabilities at each time step.

$$c_t = \frac{1}{\sum_{j=1}^{N} \alpha_t(j)} \tag{3.7a}$$

$$\alpha_t^*(j) = c_t \alpha_t(j) \quad \forall j \in \{1, \ldots N\} \tag{3.7b}$$

$$\beta_t^* = c_t \beta_t(j) \quad \forall j \in \{1, \ldots N\} \tag{3.7c}$$

It can then be shown that the scaled values of α^* and β^* can be used in each reestimation formula without any negative effects. It should be noted, however, that the scaling factors must be properly accounted for in order to correctly estimate the state transition matrix, $\mathbf{A}$. Specifically, we use the reestimation formula shown in Equation 3.8.

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \tag{3.8}$$

It should also be noted that, once scaled, α^* and β^* can no longer be used to compute $P(\mathcal{O}|\boldsymbol{\lambda})$. Alternatively, Equation 3.9 is used to compute the log probability.

$$\log P(\mathcal{O}|\boldsymbol{\lambda}) = -\sum_{t=1}^{T} \log c_t \tag{3.9}$$

3.4.2 Good-Turing Estimate

Although the presence of zeros in the state transition matrix is indicative of the underlying linguistic structure, they can also be a source of errors within the reestimation process. Specifically, once some parameter goes to zero, the output of the reestimation formula for that parameter becomes "stuck" at zero. In order to prevent this, the Good-Turing estimate is employed to keep parameters from going to zero [8]. The estimate is detailed in Equation 3.10 where $\epsilon = \frac{1}{T}$ for the purposes of this application. Although only the state transition probabilities are shown in Equation 3.10, the formula is applied to all the model parameters in $\boldsymbol{\lambda}$. In practice, this estimate is applied after each iteration of the Baum-Welch algorithm.

$$\tilde{a}_{ij} = \begin{cases} (1 - |M| \cdot \epsilon) \dfrac{a_{ij}}{\sum_{m \notin M} a_{im}} & \text{if } j \notin M = \{k \,:\, a_{ik} < \epsilon\} \\ \epsilon & \text{if } j \in M \end{cases} \tag{3.10}$$

3.5 The Cave-Neuwirth Experiment

In 1980 Cave and Neuwirth applied the Baum-Welch algorithm as described in Equation 3.6 to an observation sequence consisting of English text [5]. The text was taken from the New York Times newspaper and stripped of punctuation so that only the letters of the alphabet and spaces remained. The resulting observation sequence consisted of approximately 30,000 characters. The resulting model parameters, in particular the state transition matrix $\mathbf{A}$ and the observation probability matrix $\mathbf{B}$, were found to exhibit characteristics indicative of linguistic structure. In this section, we will examine results generated from a recreation of the Cave-Neuwirth experiment conducted using the same data set.

Algorithm 1 Algorithm used to reproduce the Cave-Neuwirth experiment.

while $\sigma > 10^{-4}$ **do**
  Compute scaled forward-backward probabilities
  Estimate $\overline{\boldsymbol{\lambda}}$ using Baum-Welch formulas
  Apply Good-Turing estimate to parameters below ϵ
  Compute new value of σ
end while

3.5.1 Experimental Setup

Each parameter of the HMM is initialized with random values and then re-estimated using the scaled reestimation formulas detailed in Section 3.4.1. Using σ as defined in Equation 3.11, convergence is determined by checking if σ falls below some desired threshold.

$$\sigma = \sum_{\lambda_i \in \boldsymbol{\lambda}} |\overline{\lambda}_i - \lambda_i| \tag{3.11}$$

Using this convergence test, the forward-backward algorithm, Baum-Welch algorithm and Good-Turing estimates are then applied iteratively until convergence is reached. An illustration of the algorithmic structure as used in this experiment is shown in Algorithm 1.

We will examine the results of four experiments conducted on the same training data set, each using an HMM with a state space of size $N = 2$ through $N = 5$. Each HMM was trained until each of its parameters converged to within a margin of 10^{-4}. In addition to the model parameters, a string of annotated text is also included. Each letter of this text is displayed above a number indicating which state had the highest probability at that instant in time as computed using Equation 3.12.

$$j^* = \arg\max_j \alpha_t(j)\beta_t(j) \tag{3.12}$$

It is significant to note that the states in each of the HMMs correspond to groupings of English letters that are linguistically significant. In the 2-state HMM, for example, consonants and vowels were clearly distinguished from one another. In a 3-state HMM consonants are further divided in to subgroups of constants that either precede or follow vowels. In the 4-state HMM, we find that spaces are now distinguished from all other letters. As the number of states in the HMM increases, the model parameters continue to represent finer details in the structure

of the text. These results are indicative of a close relationship between the hidden Markov model and linguistic structure. In fact, it can be shown that the HMM is equivalent to a right linear stochastic grammar [7]. Given this insight, it is expected that the HMM may prove highly useful in unsupervised methods of speech signal processing.

3.5.2 Estimated Model Parameters when $N = 2$

In Table 3.1, we see the inferred state transition matrix for two-state HMM. Figure 3.2 is a scatter plot of the state dependent observation probabilities denoted by b_{jk}, which are color coded to indicate the corresponding state. Lastly, Table 3.2 shows the linguistic interpretation of each state and an annotated segment of text. It is significant to note that with only a two state HMM, vowels and consonants are distinctly separated.

Table 3.1: State transition matrix $\mathbf{A}$

C \ N	**1**	**2**
1	0.27414	0.72586
2	0.72337	0.27663

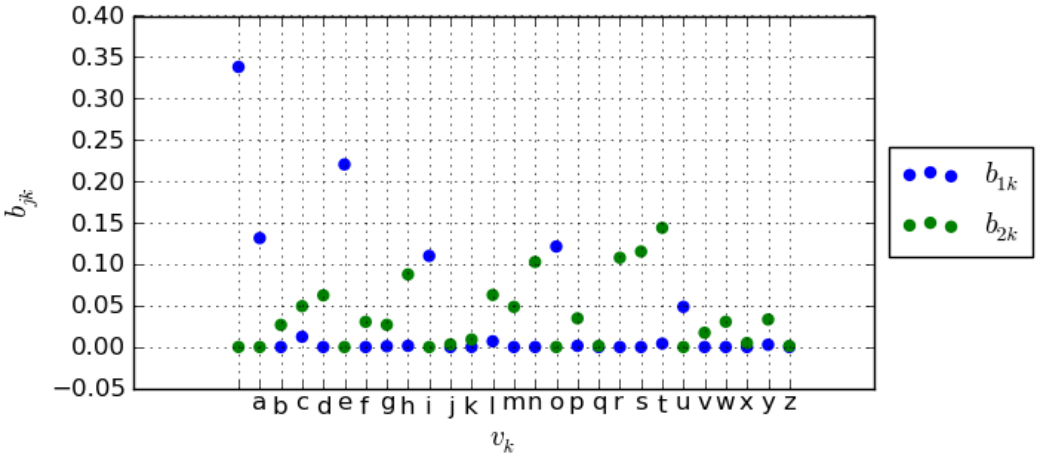

Figure 3.2: Observable process probabilities $\mathbf{B}$.

Table 3.2: Subjective interpretation of states

State	Interpretation
1	Vowels and spaces
2	Consonants
Annotated Text	$\underset{1}{\_}\ \underset{1}{i}\ \underset{2}{t}\ \underset{1}{\_}\ \underset{1}{i}\ \underset{2}{s}\ \underset{1}{\_}\ \underset{2}{n}\ \underset{1}{o}\ \underset{2}{t}\ \underset{1}{\_}\ \underset{2}{c}\ \underset{1}{o}\ \underset{2}{n}\ \underset{2}{s}\ \underset{1}{i}\ \underset{2}{d}\ \underset{1}{e}\ \underset{2}{r}\ \underset{1}{e}\ \underset{2}{d}\ \underset{1}{\_}\ \underset{2}{g}\ \underset{1}{o}\ \underset{1}{o}\ \underset{2}{d}\ \underset{1}{\_}\ \underset{2}{j}\ \underset{1}{u}\ \underset{2}{d}\ \underset{2}{g}\ \underset{2}{m}\ \underset{1}{e}\ \underset{2}{n}\ \underset{2}{t}$

3.5.3 Estimated Model Parameters when $N = 3$

The inferred state transition matrix is listed in Table 3.3 and a scatter plot of the state dependent observation probability is shown in Figure 3.3. In Table 3.4 the linguistic interpretation of each state shows that consonants have now been separated into two categories.

Table 3.3: State transition matrix $\mathbf{A}$

C \ N	1	2	3
1	0.04401	0.95599	0.0
2	0.11929	0.25435	0.62637
3	0.37165	0.55188	0.07646

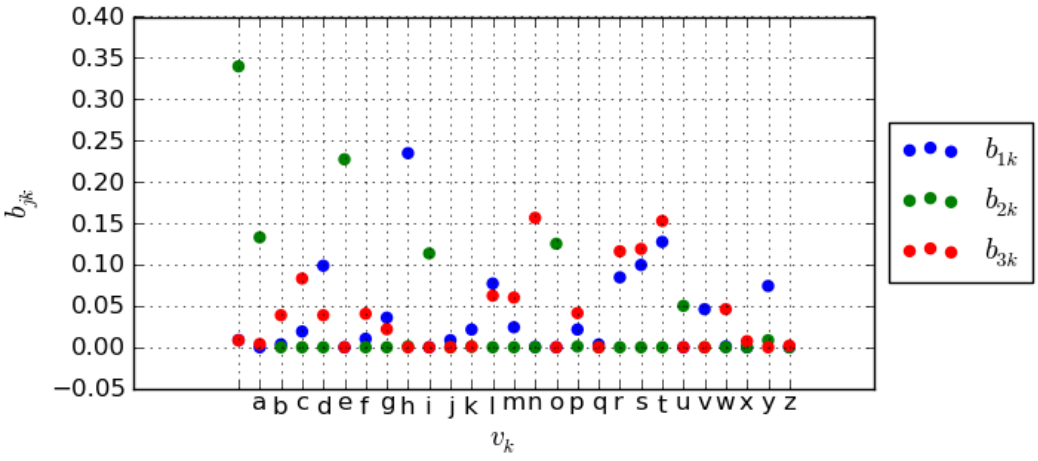

Figure 3.3: Observable process probabilities $\mathbf{B}$.

Table 3.4: Subjective interpretation of states

State	Interpretation
1	Consonants that follow consonants
2	Vowels and spaces
3	Consonants that follow vowels
Annotated Text	$\underset{2}{\_}\ \underset{2}{i}\ \underset{3}{t}\ \underset{2}{\_}\ \underset{2}{i}\ \underset{3}{s}\ \underset{2}{\_}\ \underset{3}{n}\ \underset{2}{o}\ \underset{3}{t}\ \underset{2}{\_}\ \underset{3}{c}\ \underset{2}{o}\ \underset{3}{n}\ \underset{1}{s}\ \underset{2}{i}\ \underset{3}{d}\ \underset{2}{e}\ \underset{3}{r}\ \underset{2}{e}\ \underset{3}{d}\ \underset{2}{\_}\ \underset{3}{g}\ \underset{2}{o}\ \underset{2}{o}\ \underset{3}{d}\ \underset{2}{\_}\ \underset{1}{j}\ \underset{2}{u}\ \underset{3}{d}\ \underset{3}{g}\ \underset{1}{m}\ \underset{2}{e}\ \underset{3}{n}\ \underset{1}{t}$

3.5.4 Estimated Model Parameters when $N = 4$

The inferred state transition matrix is listed in Table 3.5. A scatter plot of the state dependent observation probability is shown in Figure 3.4; however, it should be noted that the plot has been clipped because the probability of observing a space while in state 1 is above 0.9. In Table 3.6 the linguistic interpretation of each state shows that spaces have now been separated into a distinct category.

Table 3.5: State transition matrix $\mathbf{A}$

C \ N	1	2	3	4
1	0.0	0.00096	0.28461	0.71443
2	0.56256	0.37145	0.00113	0.06487
3	0.09908	0.59914	0.10256	0.19922
4	0.00247	0.0	0.84469	0.15284

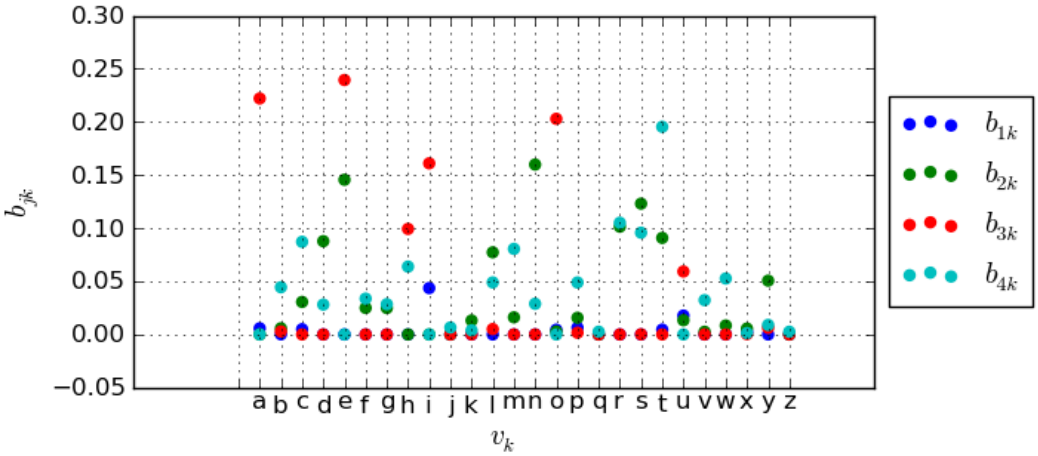

Figure 3.4: Observable process probabilities $\mathbf{B}$.

Table 3.6: Subjective interpretation of states

State	Interpretation
1	Spaces
2	Consonants that follow vowels
3	Vowels
4	Consonants that precede vowels
Annotated Text	$\underset{1}{\_}\underset{3}{i}\underset{2}{t}\ \underset{1}{\_}\underset{3}{i}\underset{2}{s}\ \underset{1}{\_}\underset{4}{n}\underset{3}{o}\underset{2}{t}\ \underset{1}{\_}\underset{4}{c}\underset{3}{o}\underset{2}{n}\underset{1}{s}\underset{4}{i}\underset{3}{d}\underset{2}{e}\underset{1}{r}\underset{4}{e}\underset{3}{d}\ \underset{1}{\_}\underset{4}{g}\underset{3}{o}\underset{2}{o}\underset{1}{d}\ \underset{4}{\_}\underset{3}{j}\underset{2}{u}\underset{1}{d}\underset{2}{g}\underset{4}{m}\underset{3}{e}\underset{2}{n}\underset{1}{t}$

3.5.5 Estimated Model Parameters when $N = 5$

The inferred state transition matrix is listed in Table 3.7. A scatter plot of the state dependent observation probability is shown in Figure 3.5, where the plot has been clipped because of the high probability of observing a space while in state 1. In Table 3.8, we see that the linguistic interpretation of each state provides a much more nuanced characterization of the data.

Table 3.7: State transition matrix **A**

C \ N	**1**	**2**	**3**	**4**	**5**
1	0.08653	0.00013	0.81359	0.09347	0.00628
2	0.71996	0.22909	0.0	0.0	0.05095
3	0.17084	0.13172	0.03607	0.33037	0.33100
4	0.28667	0.70704	0.00629	0.0	0.0
5	0.01562	0.0	0.0	0.74675	0.23763

Table 3.8: Subjective interpretation of states

State	Interpretation
1	Vowels
2	Consonants that precede vowels
3	Consonants that follow vowels
4	Spaces
5	End of word characters
Annotated Text	$\underset{4}{\_}\underset{1}{i}\underset{3}{t}\ \underset{4}{\_}\underset{1}{i}\underset{3}{s}\ \underset{4}{\_}\underset{2}{n}\underset{1}{o}\underset{3}{t}\ \underset{4}{\_}\underset{2}{c}\underset{1}{o}\underset{3}{n}\underset{5}{s}\underset{4}{i}\underset{2}{d}\underset{1}{e}\underset{3}{r}\underset{5}{e}\underset{5}{d}\ \underset{4}{\_}\underset{2}{g}\underset{1}{o}\underset{1}{o}\underset{3}{d}\ \underset{4}{\_}\underset{2}{j}\underset{1}{u}\underset{3}{d}\underset{2}{g}\underset{2}{m}\underset{1}{e}\underset{3}{n}\underset{5}{t}$

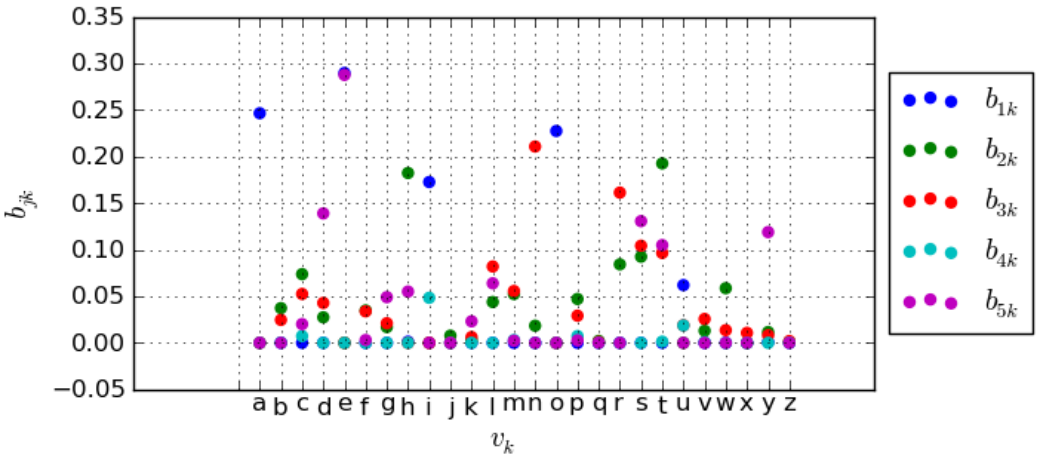

Figure 3.5: Observable process probabilities B.

CHAPTER 4

AUTOREGRESSIVE HMM

In the previous chapter, the discrete HMM was defined based on the idea that each observation was drawn from a finite vocabulary. While this particular model is very useful in applications involving discrete elements, there are many applications in which the observations are drawn from a continuum of possible values. In these cases, we must alter the HMM such that $b_j(O_t)$ becomes a state dependent probability density (or likelihood) function rather than a probability mass function as in the discrete case. Under this change, the probability of the observation sequence similarly becomes a likelihood function which is dependent on the set of model parameters $\boldsymbol{\lambda}$.

$$P(\mathcal{O}|\boldsymbol{\lambda}) \rightarrow \mathcal{L}(\mathcal{O}|\boldsymbol{\lambda})$$

In this chapter we will examine a variant of the continuous observation HMM in which the observation likelihood function is tailored specifically for application to the speech signal.

4.1 The Poritz Model

The results of the Cave-Neuwirth Experiment indicate that the HMM is well suited for applications involving human language, but these results apply only to applications where the observations are symbolic in nature. As an expansion on these results, Alan Poritz applied the HMM to speech data in an experiment similar to that of Cave and Neuwirth [9]. In the case of the Poritz experiment, however, each observation in the sequence consists of a segment taken directly from the speech signal. This was done by assuming that the observations are drawn from a p^{th} order autoregressive Gaussian probability density. This probability distribution is derived under the assumption that the error signal produced by the autoregressive model of the speech signal is Gaussian. In this formulation, the observations consist of a segment of the speech signal and are arranged into the array $\mathbf{Y}_t$ as defined

in Equation 4.1.

$$\mathbf{Y}_t = \begin{bmatrix} \mathcal{O}_t[p] & \mathcal{O}_t[p-1] & \cdots & \mathcal{O}_t[0] \\ \mathcal{O}_t[p+1] & \mathcal{O}_t[p] & & \\ \vdots & & \ddots & \vdots \\ \mathcal{O}_t[m] & \cdots & \cdots & \mathcal{O}_t[m-p-1] \end{bmatrix} \tag{4.1}$$

The corresponding observation likelihood function is then defined in Equation 4.2.

$$b_j(\mathcal{O}_t) = \frac{2}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{\mathbf{c}_j^\top \mathbf{Y}_t^\top \mathbf{Y}_t \mathbf{c}_j}{2\sigma^2} \right) \tag{4.2}$$

Based on this likelihood function, the parameter manifold that characterizes this model becomes $\boldsymbol{\lambda} = \{\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\sigma}, \mathbf{C}\}$ where each of the parameters are defined as shown in Equation 4.3.

$$\begin{aligned} \boldsymbol{\pi} &= (\pi_1, \pi_2, ...\pi_n)' \\ \mathbf{A} &= [a_{ij}] \\ \boldsymbol{\sigma} &= \{\sigma_1, \sigma_2, \ldots \sigma_n\} \\ \mathbf{C} &= \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n\} \\ &\quad \text{where } \mathbf{c}_j = [1, -c_{j1}, \ldots, -c_{jp}]^\top \end{aligned} \tag{4.3}$$

In this model, the parameters $\mathbf{c}_j$ and σ_j^2 correspond to the state dependent linear prediction filter shown in Equation 4.4. Because this filter controls the nature of the observation sequence generated by the HMM, this model is sometimes referred to as the hidden filter model [10].

$$H_j(z) = \frac{\sigma_j}{1 - \sum_{k=1}^{P} c_{jk} z^{-k}} \tag{4.4}$$

These filter parameters can then be estimated recursively using the following reestimation formulas for all j such that $1 \leq j \leq n$.

$$\mathbf{Y}_j^\top \mathbf{Y}_j = \sum_{t=1}^{T} \alpha_t(j)\beta_t(j)\mathbf{Y}_t'\mathbf{Y}_t = \begin{bmatrix} b_j & \mathbf{x}_j^\top \\ \mathbf{x}_j & \mathbf{D}_j \end{bmatrix} \tag{4.5a}$$

$$\overline{\mathbf{c}}_j = \left(1,\ -\mathbf{D}_j^{-1}\mathbf{x}_j\right) = (1,\ -c_{j1},\ \ldots\ , -c_{jp}) \tag{4.5b}$$

$$\overline{\sigma}_j^2 = \frac{\mathbf{c}_j \mathbf{R}_j \mathbf{c}_j'}{\sum_{t=1}^{T} \alpha_t(j)\beta_t(j)} \tag{4.5c}$$

Since this model makes no changes to the properties of the state transition probabilities, the reestimation formula shown in Equation 4.18 remains unchanged. It is significant to note that this model estimates the filter parameters corresponding to the autoregressive model of the speech signal as state dependent parameters. Under this construction, the state transition probabilities are expected to represent the phonetic structure of speech while the filter parameters model the short time stationary characteristics of the phonemes corresponding to each state.

Using this model with parameters $n = 5$ states and $p = 3$ filter, Poritz applied the Baum-Welch algorithm to approximately 80 seconds of speech data. The result of this experiment was that broad phonetic categories of speech, namely voiced phonemes, fricatives, nasals, plosives, and silence, were each associated with different states. In addition, the state transition matrix exhibited the phonotactic structure much like the letter ordering rules found in the Cave-Neuwirth experiment. These results extend those demonstrated by Cave and Neuwirth, in that linguistic structures may be directly inferred from speech data by the HMM.

4.2 The Autocorrelation Method

It is important to note that the model used by Poritz is directly analogous to the covariance method of linear prediction by assuming that the error, $\mathbf{e}_t$, is Gaussian. Since the covariance method is known to suffer from the instability based on the nature of the input signal as detailed in Chapter 2, we seek to develop an HMM based on the autocorrelation method of linear prediction in order to circumvent such drawbacks.

Starting with the observation probability density shown in Equation 4.6, we approximate the matrix $\mathbf{Y}_t^\top \mathbf{Y}_t$ using the short-time autocorrelation matrix $\mathbf{R}_t$ defined in Equation 4.7 where $r_t[n]$ is the autocorrelation function of a windowed version of the signal. With this approximation, we arrive at the new observation likelihood function shown in Equation 4.6, from which we may derive a set of reestimation formulas by which we may recursively estimate the parameters.

$$b_j(\mathcal{O}_t) = \frac{2}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{\mathbf{c}_j' \mathbf{R}_t \mathbf{c}_j}{2\sigma^2}\right) \tag{4.6}$$

$$\mathbf{R}_t = \begin{bmatrix} r_t[0] & r_t[1] & \cdots & r_t[p-1] \\ r_t[1] & r_t[0] & \cdots & r_t[p-2] \\ \vdots & & \ddots & \vdots \\ r_t[p-1] & & \cdots & r_t[0] \end{bmatrix} \tag{4.7}$$

It is significant to note that this new likelihood function is now dependent on a windowed version of the signal in the same manner as the autocorrelation method of linear prediction. In contrast, the likelihood function in Equation 4.2 is dependent on direct samples from the signal.

4.3 Reestimation Formulas

Using this new representation of the observable process, a new set of reestimation formulas must be derived. To do this, we follow the same process as used in [11] by Baum et al. It should be noted that this derivation and the derivation published by Baum et al. is based on optimization methods, however the Baum-Welch algorithm may also be derived as a special case of the Expectation-Maximization algorithm.

4.3.1 Likelihood Function

First, we note that the likelihood of the observation sequence, $\mathcal{L}(\mathcal{O}, \boldsymbol{\lambda})$, is defined in a manner identical to the discrete case with the exception that the $b_j(O_t)$ is now a likelihood function. Under this definition, we begin by rewriting the likelihood function in a matrix vector form. Let us assume that the HMM under consideration has N states, with state dependent linear prediction filters of order P. First we define the $N \times N$ state transition matrix $\mathbf{A} = [a_{ij}]$, and the observation matrix $\mathbf{B}_t$ to be the $N \times N$ diagonal matrix shown below.

$$\mathbf{B}_t = \begin{bmatrix} b_1(O_t) & 0 & 0 & \cdots & 0 \\ 0 & b_2(O_t) & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & b_{N-1}(O_t) & 0 \\ 0 & \cdots & 0 & 0 & b_N(O_t) \end{bmatrix}$$

Additionally, we define the forward and backward probability vectors

$$\begin{aligned}\boldsymbol{\alpha}_{t+1} &= [\alpha_{t+1}(1)\ \alpha_{t+1}(2)\ \ldots\ \alpha_{t+1}(N)]^\top \\ &= \mathbf{B}_{t+1}\mathbf{A}'\boldsymbol{\alpha}_t, \quad t = 1, 2, \ldots, T-1\end{aligned}$$

and

$$\begin{aligned}\boldsymbol{\beta}_t &= [\beta_t(1)\ \beta_t(2)\ \ldots \beta_t(N)]^\top \\ &= \mathbf{A}\mathbf{B}_{t+1}\boldsymbol{\beta}_{t+1}, \quad t = T-1, T-2, \ldots, 1\end{aligned}$$

which are initialized as

$$\boldsymbol{\alpha}_1 = \mathbf{B}_1\boldsymbol{\pi} \quad \text{and} \quad \boldsymbol{\beta}_T = \mathbf{1}$$

The initial state vector, $\boldsymbol{\pi} = [\pi_1, \pi_2, \ldots, \pi_N]^\top$, and $\mathbf{1} = [1, 1, \ldots, 1]$ are both of length N and the state transition matrix is defined as $\mathbf{A} = [a_{ij}]$. Under this matrix notation, we may now consider $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$ in a non-recursive format as shown here.

$$\boldsymbol{\alpha}_t = \mathbf{B}_t\mathbf{A}'\mathbf{B}_{t-1}\mathbf{A}'\cdots\mathbf{B}_2\mathbf{A}'\mathbf{B}_1\boldsymbol{\pi}$$

$$\boldsymbol{\beta}_t = \mathbf{A}\mathbf{B}_{t+1}\mathbf{A}\mathbf{B}_{t+2}\cdots\mathbf{A}\mathbf{B}_{T-1}\mathbf{A}\mathbf{B}_T\mathbf{1}$$

Now we may rewrite the likelihood of the observation sequence $\mathcal{O}$ given the parameter manifold $\boldsymbol{\lambda}$ as shown in Equation 4.8.

$$\mathcal{L}(\mathcal{O}, \boldsymbol{\lambda}) = \underbrace{\mathbf{1}'\mathbf{B}_T\mathbf{A}'\mathbf{B}_{T-1}\mathbf{A}'\cdots\mathbf{B}_{t+1}\mathbf{A}'}_{\boldsymbol{\beta}_t'}\underbrace{\mathbf{B}_t\mathbf{A}'\cdots\mathbf{B}_2\mathbf{A}'\mathbf{B}_1\mathbf{A}'\boldsymbol{\pi}}_{\boldsymbol{\alpha}_t} \tag{4.8}$$

Note that the matrix equation shown in Equation 4.8 may be expanded out to the canonical representation of the forward-backward algorithm as given by Equation 3.5.

4.3.2 Auxiliary Function

We define the auxiliary function Q as shown in Equation 4.9.

$$Q(\boldsymbol{\lambda}, \overline{\boldsymbol{\lambda}}) = \mathcal{L}(\mathcal{O}, \boldsymbol{\lambda}) \log \mathcal{L}(\mathcal{O}, \overline{\boldsymbol{\lambda}}) \tag{4.9}$$

Using this definition of Q, we may compute the gradient with respect to lambda and proceed to solve $\nabla_{\overline{\boldsymbol{\lambda}}} Q = 0$ for $\overline{\boldsymbol{\lambda}}$. The resulting solution represents a function of the form $\overline{\boldsymbol{\lambda}} = \mathcal{F}(\boldsymbol{\lambda})$, which maps from the parameter manifold back onto itself. Thus we have

$$\nabla_{\overline{\boldsymbol{\lambda}}} Q = \nabla_{\overline{\boldsymbol{\lambda}}} \left(\mathcal{L}(\mathcal{O}, \boldsymbol{\lambda}) \log \mathcal{L}(\mathcal{O}, \overline{\boldsymbol{\lambda}}) \right) = \frac{\mathcal{L}(\mathcal{O}, \boldsymbol{\lambda})}{\mathcal{L}(\mathcal{O}, \overline{\boldsymbol{\lambda}})} \nabla_{\overline{\boldsymbol{\lambda}}} \mathcal{L}(\mathcal{O}, \overline{\boldsymbol{\lambda}}) = 0$$

Here, we notice that this equation can be simplified further such that we must only consider the gradient of the likelihood function, $\mathcal{L}$ as shown in Equation 4.10.

$$\nabla_{\overline{\boldsymbol{\lambda}}} \mathcal{L}(\mathcal{O}, \overline{\boldsymbol{\lambda}}) = \frac{\mathcal{L}(\mathcal{O}, \overline{\boldsymbol{\lambda}})}{\mathcal{L}(\mathcal{O}, \boldsymbol{\lambda})} \cdot 0 = 0 \tag{4.10}$$

Lastly, we can consider each parameter independently based on

$$\nabla_{\overline{\boldsymbol{\lambda}}} \mathcal{L}(\mathcal{O}, \overline{\boldsymbol{\lambda}}) = 0 \quad \Rightarrow \quad \frac{\partial \mathcal{L}}{\partial \overline{\lambda}_i} = 0, \ \forall \overline{\lambda}_i \in \overline{\boldsymbol{\lambda}}$$

Furthermore, we may consider the elements of the parameter manifold as disjoint from one another; i.e., we may derive the reestimation formulas for the state transition probabilities and state dependent filter parameters independently from one another.

We now may compute $\nabla_{\overline{\boldsymbol{\lambda}}} \mathcal{L}(\mathcal{O}, \overline{\boldsymbol{\lambda}})$ to solve for the mapping $\overline{\boldsymbol{\lambda}} = \mathcal{F}(\boldsymbol{\lambda})$. For simplicity, let us assume that $\overline{\lambda}_j \in \overline{\boldsymbol{\lambda}}$ is a parameter upon which $b_j(O_t)$ is dependent. By computing the partial derivative of the likelihood function with respect to $\overline{\lambda}_j$, we arrive at

$$\frac{\partial \mathcal{L}}{\partial \overline{\lambda}_j} = \sum_{t=1}^{T-1} \boldsymbol{\beta}'_{t+1} \frac{\partial}{\partial \overline{\lambda}_j} (\mathbf{B}_{t+1}) \mathbf{A}' \boldsymbol{\alpha}_t + \boldsymbol{\beta}_1 \frac{\partial}{\partial \overline{\lambda}_j} (\mathbf{B}_1) \boldsymbol{\pi}$$

where we note that

$$\frac{\partial}{\partial \overline{\lambda}_j} (\mathbf{B}_t) = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & & & 0 \\ \vdots & & \frac{\partial}{\partial \overline{\lambda}_j}(b_j(O_t)) & & \vdots \\ 0 & & & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

Computing the matrix multiplications within the above summation, to reach Equation 4.11. By plugging this result into the summation, we have a direct representation of $\frac{\partial \mathcal{L}}{\partial \overline{\lambda}_j}$ as shown in Equation 4.12. Using Equation 4.12 as a template, we may now compute the partial derivatives of $b_j(O_t)$ with respect to the filter parameters σ_j and $\mathbf{c}_j$ as shown in Equations 4.13 and 4.14 respectively.

$$\boldsymbol{\beta}'_{t+1} \frac{\partial}{\partial \overline{\lambda}_j}(\mathbf{B}_{t+1})\mathbf{A}'\boldsymbol{\alpha}_t = \sum_{i=1}^{N} \alpha_t(i) a_{ij} \frac{\partial}{\partial \overline{\lambda}_j}(b_j(O_{t+1}))\beta_{t+1}(j) \tag{4.11}$$

$$\frac{\partial \mathcal{L}}{\partial \overline{\lambda}_j} = \sum_{t=1}^{T-1} \sum_{i=1}^{N} \alpha_t(i) a_{ij} \frac{\partial}{\partial \overline{\lambda}_j}(b_j(O_{t+1}))\beta_{t+1}(j) + \pi_j \frac{\partial}{\partial \overline{\lambda}_j}(b_j(O_1))\beta_1(j) = 0 \tag{4.12}$$

$$\frac{\partial}{\partial \sigma_j}\left(b_j(O_t)\right) = \left(\frac{1}{\sigma_j}\right)\left(\frac{\mathbf{c}_j \mathbf{R}_t \mathbf{c}'_j}{\sigma^2} - 1\right) b_j(O_t) \tag{4.13}$$

$$\frac{\partial}{\partial c_{jk}}\left(b_j(O_t)\right) = \left(\sum_{l=1}^{P} r_t[|l-k|]c_{jl}\right) b_j(O_t) \tag{4.14}$$

Plugging Equation 4.14 into Equation 4.12, we solve for $\overline{c}_{jk}$ to find the system of equations given by Equation 4.15. Here we note that, we may expand this system of equations into a matrix vector format and arrive at the reestimation formula in Equation 4.16.

$$-\sum_{l=1}^{p} \overline{c}_{jl} \sum_{t=1}^{T} r_t[|l-k|]\alpha_t(j)\beta_t(j) = \sum_{t=1}^{T} r_t[k]\alpha_t(j)\beta_t(j) \tag{4.15}$$

$$\overline{\mathbf{c}}_j = (1, -\mathbf{R}_j^{-1}\mathbf{r}_j) \tag{4.16}$$

Here we have that $\mathbf{R}_j$ and $\mathbf{r}_j$ are defined as follows.

$$\mathbf{R}_j = \begin{bmatrix} r_j[0] & r_j[1] & \cdots & r_j[p-1] \\ r_j[1] & r_j[0] & \cdots & r_j[p-2] \\ \vdots & \vdots & \ddots & \vdots \\ r_j[p-1] & r_j[p-2] & \cdots & r_j[0] \end{bmatrix}$$

$$\mathbf{r}_j = \begin{bmatrix} r_j[1] & r_j[2] & \cdots & r_j[p] \end{bmatrix}^\top$$

$$r_j[m] = \sum_{t=1}^{T} \alpha_t(j)\beta_t(j)r_t[m]$$

Similarly, we plug Equation 4.13 into 4.12 and solve for $\overline{\sigma}_j^2$ to get reestimation formulas shown in Equation 4.17.

$$\overline{\sigma}_j^2 = \frac{\overline{\mathbf{c}}_j \mathbf{R}_j \overline{\mathbf{c}}_j'}{\sum_{t=1}^{T} \alpha_t(j)\beta_t(j)} \tag{4.17}$$

Since the state transition probabilities, a_{ij}, are independent of $b_j(O_t)$, we find that reestimation formula shown in Equation 4.18 remains the same as in the discrete HMM.

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i)\beta_t(i)} \tag{4.18}$$

It is interesting to note that the reestimation formula seen in Equation 4.17 follows the same form that is found in Equation 4.5 as developed by Poritz. In addition, we observe that the matrix $\mathbf{R}_j$ is Toeplitz and therefore Equation 4.16 may be solved using the Levinson-Durbin recursion. Given this property, the resulting linear prediction coefficients share the guarantee of stability that comes with the autocorrelation method. It should also be noted that the reestimation formula for the $\mathbf{c}_j$, in Equation 4.16, is reminiscent of that used by Itakura in [4] wherein the autoregressive linear prediction coefficients are utilized to construct an approximate metric for the purpose of speech recognition. In particular, the formula posed by Itakura is demonstrated to correspond to the maximum of the likelihood function $b_j(O_t)$.

CHAPTER 5

EXPERIMENTAL RESULTS

Using the set of reestimation formulas developed in Equations 4.17 and 4.16, the Baum-Welch algorithm was applied to several recordings of speech data. Before examining the numerical results obtained from application of the autoregressive HMM, we will first detail the algorithmic structure of the HMM as it was used in this experiment. We will then consider the speech data used and detail the preprocessing applied to the data before application of the Baum-Welch algorithm.

5.1 Experimental Setup

As with the discrete HMM, practical concerns such as scaling of the forward-backward probabilities (Section 3.4.1) and application of the Good-Turing estimate (Section 3.4.2) must be included. In addition, the convergence of the algorithm is estimated similarly to the case of the discrete HMM. Specifically, a convergence parameter is computed as shown in Equation 3.11 where λ_i now corresponds to an element of the new parameter manifold $\boldsymbol{\lambda}$. For the purposes of this experiment, the algorithm was considered to have converged once the convergence parameter was computed to be less than 10^{-4}. A brief outline of the algorithmic structure used in this experiment is shown in Algorithm 2.

The data used in this experiment was obtained from the TIMIT Acoustic-Phonetic Continuous Speech Corpus [12]. Specifically, the data consists of the collection of all recordings produced by an individual speaker selected from the database, making up approximately 30 seconds of speech data with a sample rate of 16 kHz. The speech data was then segmented into overlapping analysis windows using a 30 ms Hamming window at 5 ms step sizes. Within each analysis window, the autocorrelation function was then computed for each windowed segment of the signal. This sequence of short-time autocorrelation functions was then used as input to the autoregressive HMM detailed in Chapter 4. The model order of

Algorithm 2 Algorithm used to reproduce the Caive-Neuwirth experiment.

while $\epsilon > 10^{-4}$ **do**
  Compute scaled forward-backward probabilities
  Compute $\mathbf{R}_j \quad \forall j \in \{1, 2, \ldots, N\}$
  Re-estimate $\mathbf{A}$
  Re-estimate $\mathbf{c}_j \quad \forall j \in \{1, 2, \ldots, N\}$
  Re-estimate $\boldsymbol{\sigma}_j \quad \forall j \in \{1, 2, \ldots, N\}$
  Apply Good-Turing estimate to state transition matrix $\mathbf{A}$
  Compute convergence parameter ϵ
end while

the HMM, i.e. number of states and filter coefficients, was swept over a range of values and the resulting state transition matrices and filter parameters are included in the following results.

5.2 Results

The following results were generated by performing a parameter sweep on both the order of the state space N and the order of the linear prediction filter P. From this parameter sweep, two examples were selected and the results are show in this section. For each of these parameter settings, the inferred state transition matrices are shown in Tables 5.1 and 5.3 as well as the state dependent filter parameters in Figures 5.1 through 5.3 and Figures 5.5 through 5.9. A history of the probability of each state was then computed using Equation 5.1.

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\sum_{i=1}^{n} \alpha_t(i)\beta_t(i)} \tag{5.1}$$

Using the probability history, segments of the speech signal corresponding to a high state probability (greater than 80%) were extracted and saved to an audio file. These files were played back through a set of speakers in order to identify the phonetic category of the speech sounds corresponding to each state. The phonetic categories identified for each state are listed in Tables 5.2 and 5.4. In addition, the spectrogram and probability history of each state were plotted for a 3 second segment of the speech data and shown in Figures 5.4 and 5.10. This visualization of state probability history is particularly useful in demonstrating this model's ability to automatically detect different phonetic categories.

Interestingly, it was found that the order of the state space drove a requirement

on the order of the state dependent filters. This observation logically flows from the idea that each of the state dependent filters must be able to effectively reject speech sounds from other states. Conversely, it should be noted that arbitrarily increasing the order of the filters does not necessarily improve performance. This observation similarly follows from the fact that each filter must effectively pass speech sounds from a relatively broad set of phonetic categories given that the state space is relatively small. The result of applying too high a filter order for the given state space is that a large portion of the states will converge to a relatively specific phonetic category while the remaining states capture an overly broad set of phonetic categories.

5.2.1 $N = 3$ States, $P = 5$ LPCs

The inferred state transition matrix listed in Table 5.1 has a dominant diagonal component, indicating that each state tends to transition back onto itself. This observation is consistent with the justification of the short-time stationary assumption discussed in Chapter 2. The linguistic interpretation listed in Table 5.2 indicates that voiced speech and fricatives are most easily separated from the data. The distinctive frequency response of each filter in Figures 5.1 through 5.3 support this observation. Lastly, Figure 5.4 indicates that even a three state HMM is capable of identifying significant boundaries between speech sounds.

Table 5.1: State transition matrix **A**

From \ To	**1**	**2**	**3**
1	0.95641	0.04213	0.00146
2	0.00959	0.98189	0.00852
3	0.00857	0.04663	0.94480

Table 5.2: Phonetic categories represented by each state

State	**Interpretation**
1	Vowels and Semivowels
2	Plosives, Nasals and Silence
3	Fricatives

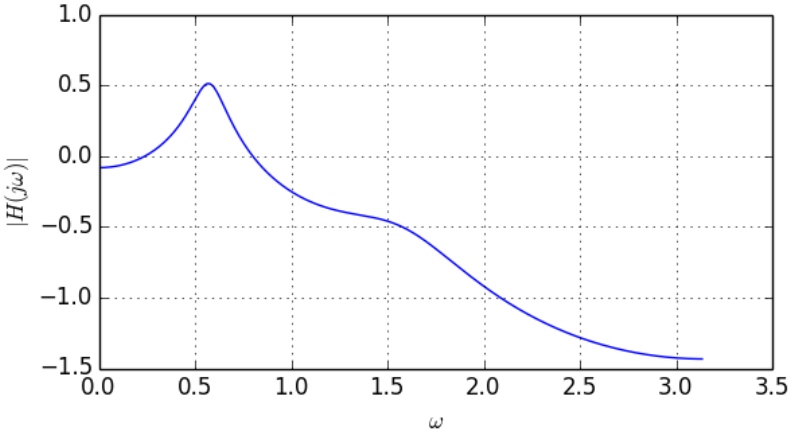

Figure 5.1: Frequency response of the filter for state 1.

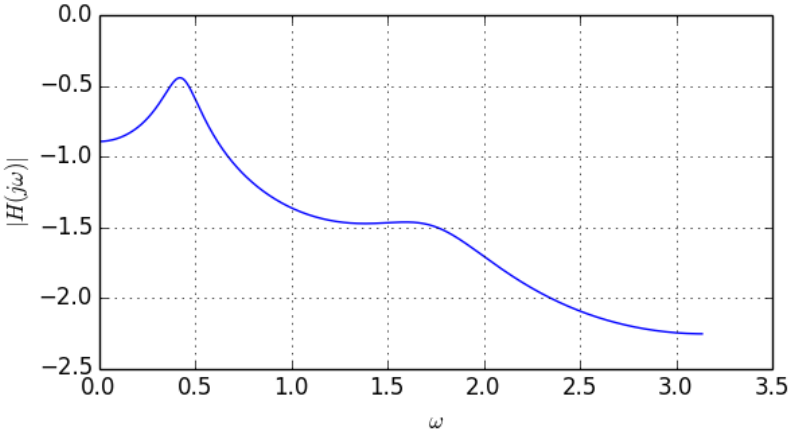

Figure 5.2: Frequency response of the filter for state 2.

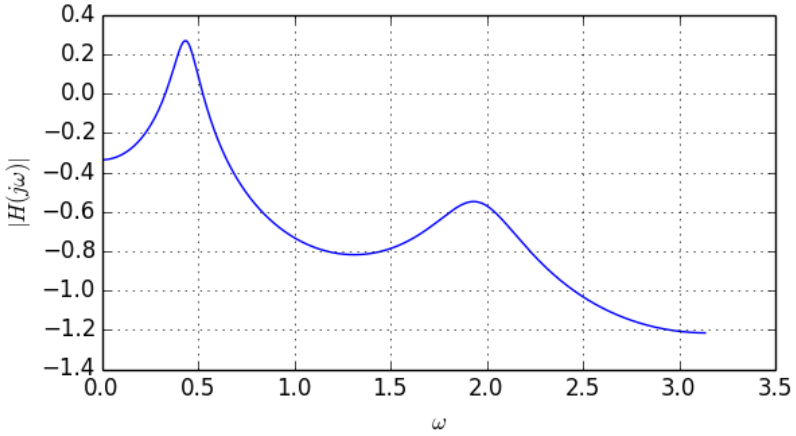

Figure 5.3: Frequency response of the filter for state 3.

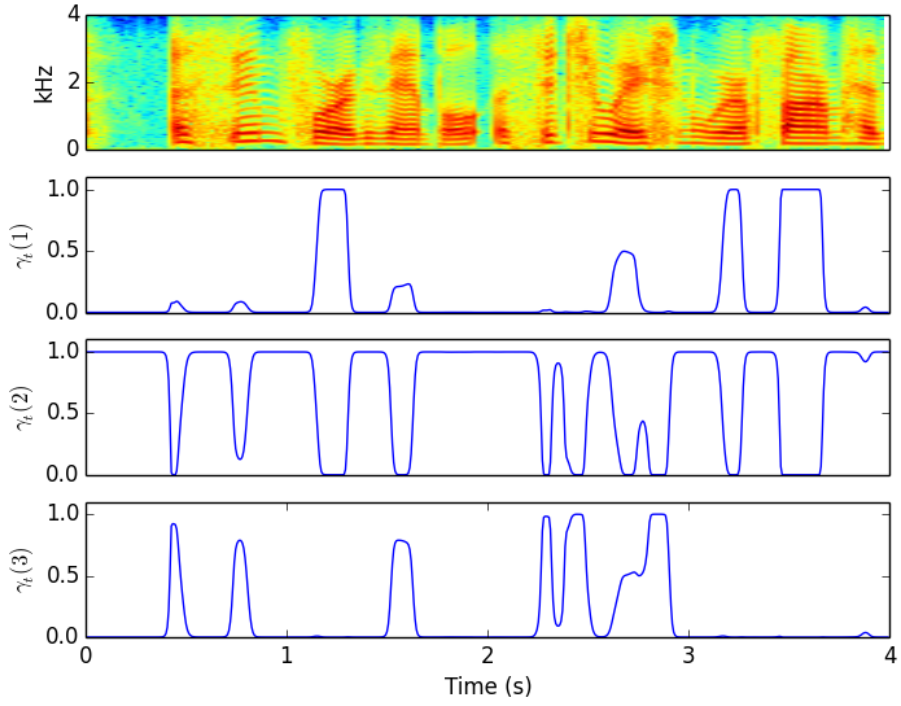

Figure 5.4: State probability history for N=3 and P=5.

5.2.2 $N = 5$ States, $P = 5$ LPCs

The state transition matrix listed in Table 5.3 has a structure similar to that of the three state HMM. Table 5.4 indicates an improved separation of the types of speech sounds. In particular, we notice that voiced speech is now separated into three separate states. In addition, Figures 5.5 through 5.9 demonstrate a more nuanced spectral characterization of each category of speech sound. Lastly, Figure 5.10 demonstrates a marked improvement in identification of important boundaries between speech signals.

Table 5.3: State transition matrix **A**

From \ To	**1**	**2**	**3**	**4**	**5**
1	0.93022	0.01309	0.00016	0.05637	0.00016
2	0.00723	0.94400	0.00852	0.04008	0.00016
3	0.01825	0.03291	0.94851	0.00016	0.00016
4	0.00193	0.01094	0.00016	0.97804	0.00893
5	0.01994	0.00016	0.00016	0.03609	0.94364

Table 5.4: Phonetic categories represented by each state

State	**Interpretation**
1	Vowels (eh)
2	Semivowels
3	Vowels (ah)
4	Plosives and Silence
5	Fricatives

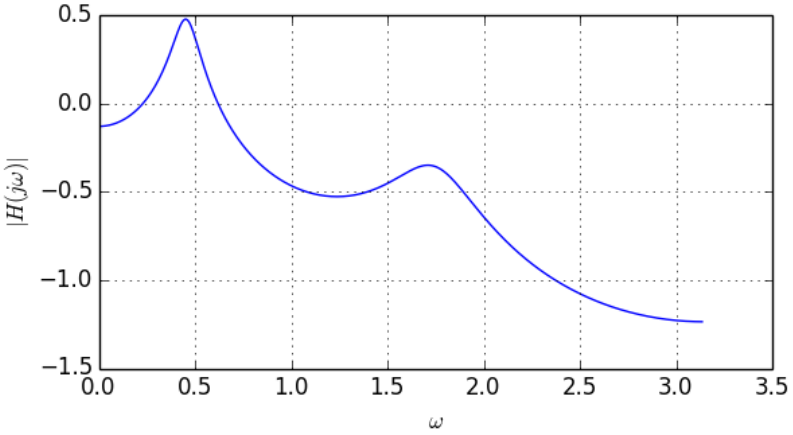

Figure 5.5: Frequency response of the filter for state 1.

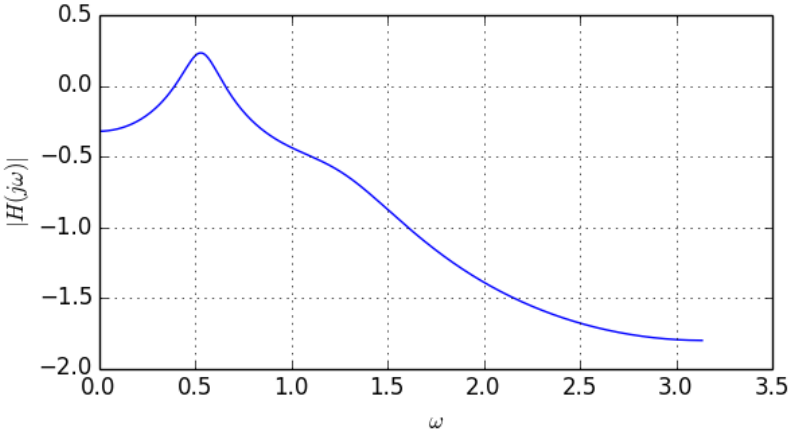

Figure 5.6: Frequency response of the filter for state 2.

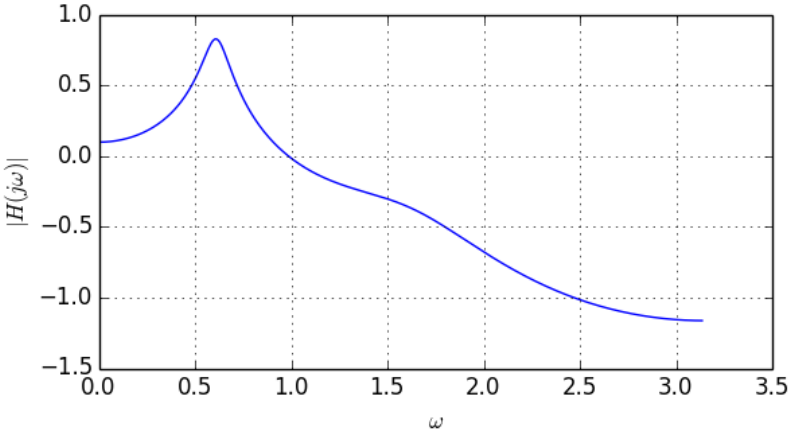

Figure 5.7: Frequency response of the filter for state 3.

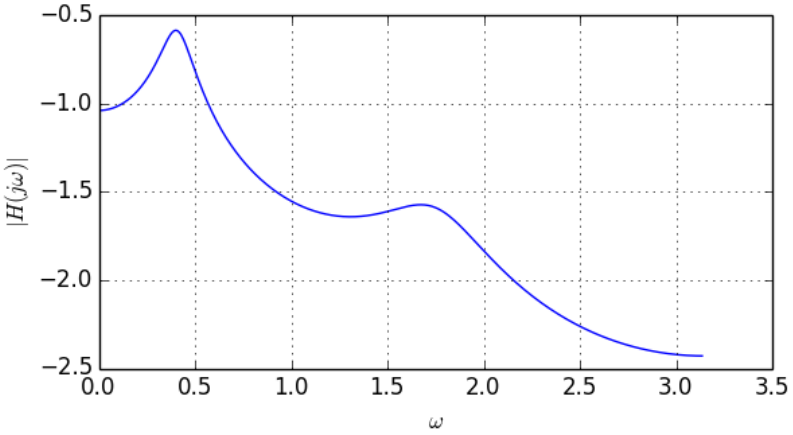

Figure 5.8: Frequency response of the filter for state 4.

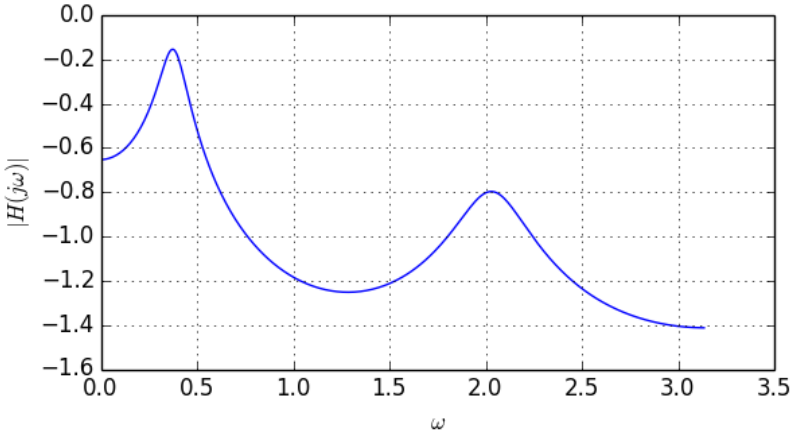

Figure 5.9: Frequency response of the filter for state 5.

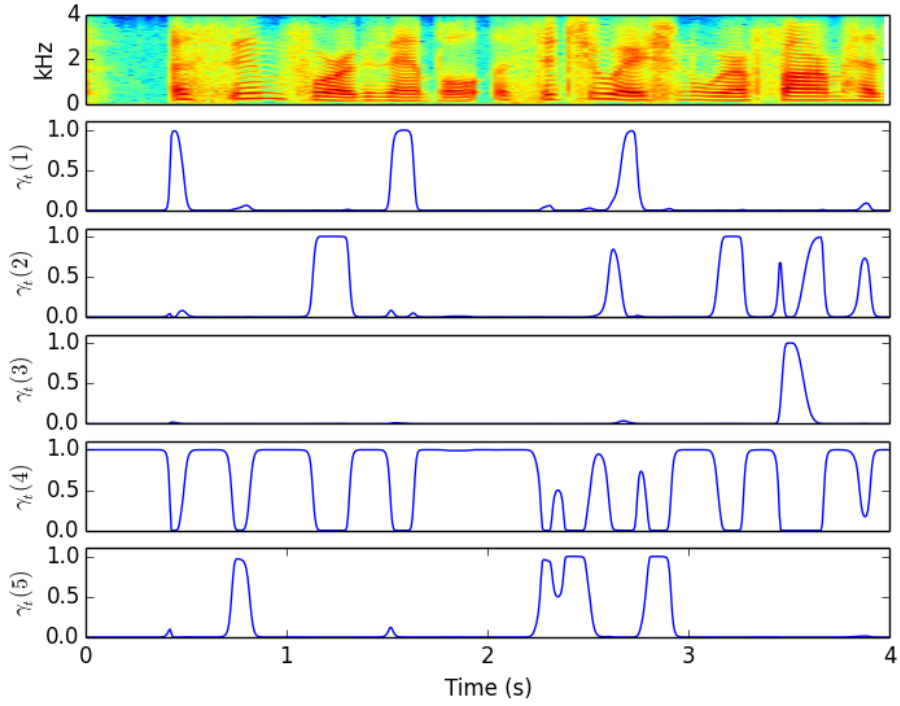

Figure 5.10: State probability history for N=5 and P=5.

CHAPTER 6

CONCLUSION

The autoregressive hidden Markov model based on the autocorrelation method of linear prediction represents a powerful tool for the purposes of speech signal processing. Qualitatively, experimental results showed that model parameters inferred by this autoregressive HMM represent broad phonetic categories found in the speech data in a manner similar to the model used by Poritz [9]. It is significant to note that this new model is capable of converging onto a set of parameters given a random initialization. As a result, this model represents a method of inferring the linguistic structure of the speech signal in an unsupervised way.

By placing these results in the broader context of language acquisition, the autoregressive HMM may prove useful as a first stage in unsupervised speech signal processing for speech recognition. In particular, this model would be particularly effective for the purpose of identifying syllable or phoneme boundaries without any prior knowledge of the language being spoken. Given that the state dependent filter parameters are guaranteed to be stable, this model could also be utilized for the purposes of speech synthesis. With this in mind, this autoregressive HMM may play an important role in the modeling process by which a machine may acquire language.

REFERENCES

[1] A. M. Turing, "Computing machinery and intelligence," *Mind*, no. 49, pp. 433–460, 1950.

[2] R. Bijeljac-Babic, J. Bertoncini, and J. Mehler, "How do 4-day-old infants categorize multisyllabic utterances?" *Developmental Psychology*, vol. 29, no. 4, pp. 711–721, 1993.

[3] T. F. Quatieri, *Discrete-Time Speech Signal Processing*, ser. Prentice-Hall signal processing series. Upper Saddle River, NJ: Prentice Hall PTR, 2002.

[4] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 23, no. 1, pp. 67–72, 1975.

[5] R. L. Cave and L. P. Neuwirth, "Hidden Markov models for English," in *Proc. Symp. on the Application of Hidden Markov Models to Text and Speech*, J. D. Ferguson, Ed., Princeton, NJ, 1980, pp. 16–56.

[6] L. E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of a Markov process," *Inequalities*, vol. 3, pp. 1–8, 1970.

[7] S. E. Levinson, *Mathematical Models for Speech Technology*, ser. Prentice-Hall signal processing series. The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England: John Wiley and Sons Ltd, 2005.

[8] I. J. Good, "The population frequencies of species and the estimation of population parameters," *Biometrika*, vol. 40, no. 3/4, pp. 237–264, 1953.

[9] A. B. Poritz, "Linear predictive hidden Markov models and the speech signal," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, 1982, pp. 1291–1294.

[10] A. B. Poritz, "Hidden Markov models: A guided tour," in *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*. IEEE, 1988, pp. 7–13.

[11] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[12] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, and N. Dahlgren, "TIMIT acoustic-phonetic continuous speech corpus LDC93S1," *Philadelphia: Linguistic Data Consortium*, 1993.