**A COMPARATIVE ANALYSIS OF BAYESIAN NONPARAMETRIC VARIATIONAL INFERENCE ALGORITHMS FOR SPEECH RECOGNITION**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**A Thesis Proposal**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**in Partial Fulfillment**

**of the Requirements for the Degree of**

**Master of Science in Electrical Engineering**

**By**

**John Steinberg**

**December, 2012**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Dr. Joseph Picone**

**Thesis Advisor**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Dr. Chang-Hee Won Dr. Iyad Obeid**

**Professor of Electrical & Professor of Electrical &**

**Computer Engineering Computer Engineering**

**Committee Member Committee Member**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Dr. Alexander Yates Dr. Marc Sobel**

**Professor of Computer & Professor of Statistics**

**Information Science Committee Member**

**Committee Member**

# ABSTRACT

Nonparametric Bayesian models have become increasingly popular in speech recognition tasks such as language and acoustic modeling due to their ability to discover underlying structure in an iterative manner. Nonparametric methods do not require a priori assumptions about the structure of the data, such as the number of mixture components, and can learn this structure directly from the data. Dirichlet process mixtures (DPMs) are a widely used nonparametric method. These processes are an extension of the Dirichlet distribution which spans non-negative numbers that sum to one. Thus, DPMs are particularly useful for modeling distributions of distributions. Because DPMs potentially require infinite parameters, inference algorithms are needed to approximate these models for sampling purposes. The focus of this work is an evaluation of three of these Bayesian inference algorithms, which have only recently become computationally viable: Accelerated Variational Dirichlet Process Mixtures (AVDPM), Collapsed Variational Stick Breaking (CVSB), and Collapsed Dirichlet Priors (CDP).

Rather than conducting a complete speech recognition experiment where classification is affected by several factors (i.e. language and acoustic modeling), a simple phone classification task is chosen to more clearly assess the efficacy of these algorithms. Furthermore, this evaluation is conducted using the CALLHOME Mandarin and English corpora, two languages that, from a human perspective, are phonologically very different. This serves two purposes: (1) Artifacts from either language that influence classification will be identified. (2) If no such artifacts exist, then these algorithms will have strongly supported their use for future multi-language recognition tasks. Finally, Mandarin misclassification error rates have consistently been much higher than those from comparable English experiments. Thus the last goal of this work is to show whether these three inference algorithms can help reduce this gap while maintaining reasonable computational complexity.

# TABLE OF CONTENTS

[ABSTRACT ii](#_Toc341711156)

[TABLE OF CONTENTS iii](#_Toc341711157)

[LIST OF TABLES v](#_Toc341711158)

[LIST OF FIGURES vi](#_Toc341711159)

[CHAPTER 1 1](#_Toc341711160)

[INTRODUCTION 1](#_Toc341711161)

[1.1 Speech Recognition Overview 2](#_Toc341711162)

[1.2 Previous Work 4](#_Toc341711163)

[1.2.1 Generative, Discriminative, Neural Network, and Exemplar Models 4](#_Toc341711164)

[1.2.2 Applications of Nonparametric Models 7](#_Toc341711165)

[1.3 Thesis Overview 8](#_Toc341711166)

[CHAPTER 2 9](#_Toc341711167)

[INFERENCE ALGORITHMS 9](#_Toc341711168)

[2.1 Dirichlet Distributions & Processes 10](#_Toc341711169)

[2.1.1 Dirichlet Distributions 10](#_Toc341711170)

[2.1.2 Dirichlet Processes 13](#_Toc341711171)

[2.2 Variational Inference 15](#_Toc341711172)

[2.2.1 Accelerated Variational Dirichlet Process Mixtures (AVDPM) 16](#_Toc341711173)

[2.2.2 Collapsed Variational Stick Breaking (CVSB) and Collapsed Dirichlet Priors (CDP) 18](#_Toc341711174)

[CHAPTER 3 20](#_Toc341711175)

[DATA & EXPERIMENTS 20](#_Toc341711176)

[3.1 Data 20](#_Toc341711177)

[3.1.1 Mandarin Chinese vs. English: A Language Comparison 20](#_Toc341711178)

[3.1.2 LDC's CALLHOME Corpora 23](#_Toc341711179)

[3.2 Experimental Setup 24](#_Toc341711180)

[3.2.1 Data Preparation 24](#_Toc341711181)

[CHAPTER 4 29](#_Toc341711182)

[PRELIMINARY EXPERIMENTS & RESULTS 29](#_Toc341711183)

[4.1 Baseline Experiments 29](#_Toc341711184)

[4.2 Preliminary Results 30](#_Toc341711185)

[4.2.1 TIMIT 30](#_Toc341711186)

[4.2.2 CALLHOME English 32](#_Toc341711187)

[CHAPTER 5 34](#_Toc341711188)

[EXPECTATIONS & FUTURE WORK 34](#_Toc341711189)

[5.1 Expected Outcomes 34](#_Toc341711190)

[5.2 Timeline for Future Work 34](#_Toc341711191)

[REFERENCES 37](#_Toc341711192)

# LIST OF TABLES

**Table 1: A table showing the amount of CALLHOME English data before and after filtering out segments of overlapping speakers. 25**

**Table 2: A chart of the converted phone labels for the TIMIT corpus 28**

**Table 3: A table of misclassification errors generated from the random forest algorithm for varying numbers of trees. 30**

**Table 4: A table of misclassification errors generated from the neural network algorithm for varying numbers of neurons in the hidden layer. 31**

**Table 5: A table of misclassification errors generated from the K-nearest neighbor algorithm for varying values of K. For K > 1 misclassification is determined by a weighted average using a Euclidean distance metric. 32**

**Table 6: A table of misclassification errors generated from the random forest algorithm for varying numbers of trees. 32**

**Table 7: A table of misclassification errors generated from the neural network algorithm for varying numbers of neurons in the hidden layer. 33**

**Table 8: A table of misclassification errors generated from the K-nearest neighbor algorithm for varying values of K. For K > 1 misclassification is determined by a weighted average using a Euclidean distance metric. 33**

# LIST OF FIGURES

**Figure 1: A block diagram of a typical speech recognition system 3**

**Figure 2: A diagram depicting the Polya's urn method. 12**

**Figure 3: An outline of the steps used for the stick-breaking approach to DPs 13**

**Figure 4: An outline of the process of sampling from a DP using the Polya Urn method 15**

**Figure 5: An outline of the process of sampling from a DP using the stick-breaking approach 15**

**Figure 6: A description of the different tones in Mandarin Chinese. Words in ( ) indicate examples of English words that share similar sounds to Mandarin tones. 21**

**Figure 7: An example showing several equivalent sentences using different word orders (Lee, 2006) 22**

**Figure 8: An example of a neural network architecture used in this work. 29**

# CHAPTER 1

# INTRODUCTION

For the past several years acoustic modeling in speech recognition has been dominated by parametric statistical models. More specifically, Hidden Markov Models (HMM) trained using Mel frequency cepstral coefficients (MFCCs) have proven to yield reasonable error rates while maintaining feasible computational costs. More recently, however, researchers have increasingly paid attention to nonparametric Bayesian clustering methods that can automatically find underlying structure directly in the data. Prominent among these are Dirichlet Process Mixture Models (DPMMs). As an extension of the Dirichlet distribution, which spans non-negative numbers that sum to one, DPMMs are particularly useful for modeling distributions of distributions.

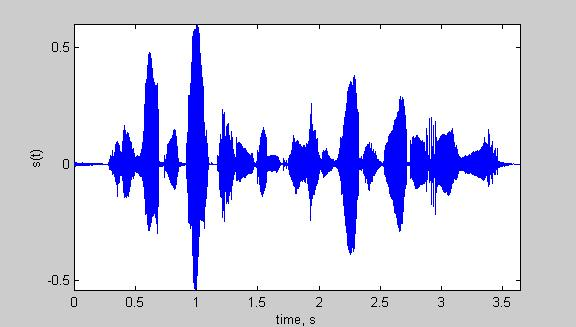
Additionally, where parametric methods such as HMMs have a predetermined number of mixtures, the number of mixtures in a DPMM can grow as new data becomes available. Thus, while parametric methods can capture general acoustic features by averaging across a set number of distributions, DPMMs' potentially infinite distributions can simultaneously capture more unique acoustic traits of individual speakers along with the more general acoustic features used for phone identification (Harati Nejad Torbati et al., 2012). This high degree of complexity makes sampling from these distributions intractable, however, so approximations are made using inference algorithms. The focus of this work is an evaluation of three of these Bayesian inference algorithms, which have only recently become computationally viable: Accelerated Variational Dirichlet Process Mixtures (AVDPM), Collapsed Variational Stick Breaking (CVSB), and Collapsed Dirichlet Priors (CDP).

Rather than conducting a complete speech recognition experiment where classification is affected by several factors (e.g. language and acoustic modeling), a simple phone classification task is chosen to more clearly assess the efficacy of these algorithms. Furthermore, this evaluation is conducted using the CALLHOME Mandarin and English corpora, two languages that, from a human perspective, are phonologically very different. This serves two purposes: (1) artifacts from either language that influence classification will be identified; (2) if no such artifacts exist, then these algorithms will have strongly supported their use for future multi-language recognition tasks. Finally, Mandarin misclassification error rates have consistently been much higher than those from comparable English experiments. Thus the last goal of this work is to show whether these three inference algorithms can help reduce this gap while maintaining reasonable computational complexity.

This chapter will focus first on a simple introduction to speech recognition followed by some of the statistical methods commonly used in speech recognition tasks. This includes generative models (e.g. HMMs), discriminative models, neural networks, and exemplar based approaches. Finally a brief overview of how nonparametric models have been adapted to speech research is also provided.

## Speech Recognition Overview

Although this work does not focus directly on improving the overall performance of a speech recognition system it is worth describing the general process since it will be alluded to later. Most systems follow a process similar to the one shown below in (Picone, 2012).



**Acoustic  
Front-end**

**Acoustic Models  
P(A/W)**

**Language Model  
P(W)**

**Search**

**Input  
Speech**

**Recognized Utterance**

Figure 1: A block diagram of a typical speech recognition system

The acoustic front end converts input audio data into feature vectors. Mel frequency cepstral coefficients (MFCCs), very commonly used features (Young, 1996) are used in this work. The new feature vectors are then passed to the next stage of the speech recognizer where acoustic models are trained. A wide variety of training methods can be used in this stage and are addressed in the following sections. To name a few, these can include parametric or non-parametric methods using generative, discriminative, or exemplar-based models. The general training process typically begins by first training monophone models which in turn are used to train more complex triphone models. During the triphone training phase, a decision tree is used for state-tying to help reduce the amount of necessary computation. A language model is also trained independently of the acoustic models. N-grams, a simple and very common type of language model, determine the probability of a word conditioned on the N previous words from the utterance, *P(Wk | Wk-1, Wk-2, …, Wk-N)*. The language model's score is combined with the acoustic model's to generate a maximum likelihood score. Either word lattices or one-best scores are used to determine the most likely output utterance(s). These newly created labels are compared to reference labels and a WER is determined for the system.

Although the focus of this work is not to evaluate a complete speech recognition system's performance, an acoustic model is still trained. Unlike the TIMIT corpus that includes a pre-generated phone alignment for the data, i.e. the locations of individual phones within the data,, the CALLHOME English and Mandarin corpora do not. Thus a monophone acoustic model is trained with HMMs and 16 Gaussian mixtures to generate a Viterbi-based time alignment. Features corresponding to individual phones can then be extracted and used for this phone classification task.

## Previous Work

### Generative, Discriminative, Neural Network, and Exemplar Models

A wide variety of statistical modeling methods have been proposed to improve speech recognition performance. HMMs, the most commonly used type of generative models, were introduced to speech research in the 1980s and were widely acknowledged for their ability to model temporally changing data such as speech. These generative, graphical models utilize GMMs to represent the probability density function (pdf) of a feature vector *xt* at a given time *t*:

|  |  |  |
| --- | --- | --- |
|  |  | ((1) |

where are the HMM parameters, represents the weights of K mixtures such that , and and represent the means and covariances of the mixtures. With a collection of feature vectors , the likelihood of the data is then:

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

where S={*st*}is the set of state labels, π is the state probability such that , *a* represents the state transition probability such that . Equation (2) is trained using a maximum likelihood (ML) approach known as the expectation maximization (EM) algorithm. In each iteration a new set of HMM parameters are generated such that:

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

Finally, with an acoustic score from the HMMs and a language model as a prior, final classification is done:

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

Given labels, generative models like HMMs attempt to fit distributions to the data. Unlabeled data is then passed to the system and the output hypothesis is based on the maximum likelihood from the trained distributions. This is not necessarily optimal since theoretically an infinitely large set of data is required to correctly model a distribution and this can lead to several difficulties. Huge corpora can make systems computationally inefficient, while in other cases there may be insufficient amounts of data to train on. Furthermore, system performance is often evaluated by directly measuring a misclassification error rate. Discriminative models attempt to do this by maximizing the log of the posterior probability, *log P(W|X)* , i.e. the probability of a word (or phone) given the data.

One very common discriminative model is the neural network. These models are defined by a three level architecture, i.e. an input, hidden, and output layers, such that features are mapped to subsequent layers using a non-linear transfer function. A very choice is the sigmoid transfer function given by :

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

where *y* is the layer following *x*, *yj* is the new state in layer *y, bj* is the bias coefficient and *wij* are the weights for connections from layer *i* to layer *j*. It is important these weights are initially given small random values to break symmetry and prevent identical gradients among different states (i.e. preventing all weights within a layer from being equal). A cost function, such as minimizing mean squared error, is used to train the model using a back propagation algorithm.

This architecture makes neural networks ideally suited to modeling non-linear speech data that is generated by a relatively few number of variables, i.e. parts of the vocal tract. These models still require extremely large numbers of parameters, especially when modeling triphones which require an output layer of several thousands, and it was not until recent advances in hardware and training algorithms that manageable computation time was achieved. New research has shown that deep neural networks (DNNs), consisting of multiple hidden layers, can avoid the endemic problem of overfitting and are able to outperform GMM-HMM systems (Hinton et al., 2012); (Larochelle et al., 2007).

Many other discriminative models exist and Saon & Chien (2012) provide an excellent overview of some of these methods. These include techniques that incorporate objective functions that minimize classification error (MCE), discriminatively train acoustic models that minimize word or sentence error rates using maximized mutual information (MMI), or by minimizing phone error (MPE). All of these have shown to produce results that are significantly better than ML approaches (Juang, Chou, & Lee, 1997); (Bahl, Brown, de Souza, & Mercer, 1986); (Povey & Woodland, 2002). The difficulty that these discriminative approaches face lie with the optimization of objective functions. While HMMs and other generative models estimate continuous distributions, classification errors are discrete and therefore have parameters that cannot be estimated using gradient descent based algorithms. Although a complete description of these discriminative models and their training methods are beyond the scope of this paper it is worth pointing out that the methods mentioned above require various smoothing techniques.

All of the generative and discriminative methods mentioned in the above paragraphs build models by generalizing training data and are known as global data techniques. On the opposite end of the spectrum are exemplar based models which aim to classify data based on just a few examples or templates. While global data models can suffer if the amount of training data is limited, exemplar based techniques can still yield high performance (Belkin & Niyogi, 2003). These systems typically follow three stages (Sainath et al., 2012): exemplar modeling, instance modeling, and decoding. In the first stage, the search space is reduced and the best exemplars for a given label are determined. In the next phase, a given test vector is compared to the exemplar and weights are assigned to each of them. Finally, an acoustic score is generated that ultimately determines the most probable utterance. K nearest neighbors (KNN) is a good example of an exemplar model in which a distance metric is used to determine the *k* closest exemplars to a test vector. There are several methods to improve the computational efficiency of KNN (Samet, 2008) but these are beyond the scope of this paper.

It is worth mentioning that for many practical purposes, hybrid systems are developed that incorporate several of the models discussed above. These often include an initial training pass using traditional HMMs followed by a second pass with a discriminative, neural network, or SVM classifier to determine an exact emitting state (Heigold, Ney, Schluter, & Wiesler, 2012). A DNN-HMM system, for example, resulted in a 33% reduction in WER from the standard GMM-HMM performance on NIST's 2003 FSH test corpus (i.e. WER dropped from 27.4% to 18.5%) (Seide, Li, & Yu, 2011).

### 1.2.2 Applications of Nonparametric Models

The application of nonparametric Bayesian models to speech research has become increasingly popular although mostly in language modeling and speaker adaptation. In this section, some of these uses will be described.

N-grams are very simple and commonly used methods for language modeling where *P(W)* from (4) is determined by a conditional probability of a word given the previous words. This is a fast and efficient technique but is only able to capture local lexical information since *n* has to be manually set and is typically a relatively small number due to data sparseness constraints. To help reduce this, there have been several attempts at new smoothing techniques such as Kneser-Ney language models (KNLM), that help eliminate cases where various n-grams don't exist within the training data (S. F. Chen & Goodman, 1999). More recently, nonparametric Bayesian techniques such as the Pitman-Yor language model (PYLM) have expanded KNLMs and allow n-gram parameters to grow as new data is introduced to the system (Renals, 2010). This helps to greatly reduce the risk of over-fitting or under-fitting test data. Other techniques such as the latent Dirichlet allocation language model (LDALM) (Tam & Schultz, 2005), have been used to capture non-local lexical information. Rather than model a word's probability strictly from its context (i.e. *n-1* previous words), LDALMs map the word history into a topic class and therefore base the probability of a word on word history and semantic content.

Speaker adaptation is another area of research where Dirichlet processes are becoming increasingly popular. In ideal settings speech recognition systems are trained by infinitely large corpora containing words spoken by infinitely many speakers. Obviously, this is impossible to achieve in any real-world situation so speaker adaptation is used to complement acoustic training. This leads to higher performing systems that are trained only on data from a limited number of speakers or training environments. DPs' allow model parameters to grow as new data is introduced to a system which make them ideally suited to the task of offsetting these disparities between training and testing data. (Harati Nejad Torbati et al., 2012) have shown that DPMM based speaker adaptation has reduced WER by up to 10% from the more common maximum likelihood linear regression (MLLR) method.

## Thesis Overview

The remainder of this thesis will be broken down into the following sections. In Chapter 2 an in depth description of Dirichlet process mixture models is provided followed by explanations of the three inference algorithms used in this thesis: Accelerated Variational Dirichlet Process Mixtures, Collapsed Variational Stick Breaking, and Collapsed Dirichlet Priors. Experimental set-ups including data preparation are discussed in Chapter 3 along with a discussion of the major language differences between English and Mandarin. Chapter 4 provides a detailed analysis of the results found from these experiments and finally Chapter 5 presents possible directions that this work may lead to in future research.

# CHAPTER 2

# INFERENCE ALGORITHMS

Parameterized models have been widely popular for their efficiency, ease of use, and reasonable performance in various clustering and classification problems. They are of particular use when the underlying structure of the data is either previously known or can be easily approximated. However, this results in a significant limitation in application since clusters in real data often vary in shape and may follow their own individual distributions (Mallapragada, et. al., 2010). Thus parametric methods can sometimes fail to produce successful models. Nonparametric models on the other hand are able to independently infer the underlying structure of data by determining an optimal number of clusters or latent variables during fitting. Moreover, the complexity and accuracy of the model evolves as more data is added to the system. A good example of a nonparametric model is the Dirichlet process. In its basic essence DP's are distributions of distributions that thereby (theoretically) require an infinite number of parameters. As Sudderth (2006) pointed out, if θ is a probability measure spanning an infinite dimension space , then N infinitely exchangeable random variables can be described as:

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

If *X* is actually a *K*-dimensional discrete space then *θ* can be reduced to a *K-1* simplex. Moreover, if N is adequately large but is not actually infinitely exchangeable the distortion caused by this is not significant. Finally, *θ* is often assumed to have hyperparameters λ.

The purpose of these models is to determine the posterior probability of an event given a model's parameters, e.g. a speech recognition system attempts to find the probability that a word was spoken given a trained acoustic model. However, because these models are theoretically infinitely large, it is often impossible to manipulate such distributions directly so inference algorithms are used instead to generate predictions. In the next few sections Dirichlet distributions and processes will be introduced along with the three variational inference algorithms used for classification: AVDPMs, CVSBs, and CDPs.

## 2.1 Dirichlet Distributions & Processes

In these sections a basic overview of definitions and properties of Dirichlet distributions and processes are put forth. The following sections typically follow the explanations found in Frigyik et. Al.'s "Introduction to the Dirichlet Distribution and Related Processes" (Frigyik, Kapila, & Gupta, 2010).

### 2.1.1 Dirichlet Distributions

A Dirichlet distribution in its most basic definition is a distribution over pmfs. Mathematically this can be represented by the following. First, let *Q = | Q1, Q2, …, Qk |* be a random pmf such that *Qi ≥ 0* and . Furthermore, set the Dirichlet distribution's parameter *α = | α1, α2, …, αk |* such that *αi > 0* and *.* With these set, the Dirichlet distribution is given by:

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

Thus in the case where k=2 and if Q = (X, 1-X), (7) reduces to a Beta distribution:

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

A real world example of this is offered by Frigyik et. al. (2010). In this application a bag full of one hundred 6-sided dice is given. A roll of each die therefore has the same set of possible outcomes as any other die (1, 2, 3, 4, 5, or 6) but might have slightly different probabilities of achieving them. This can be modeled with a Dirichlet distribution, Dir(α), consisting of pmfs q(i), where I = 1, 2, …, L (and L=100 dice in this example), that have ni samples drawn from the ith pmf. This concept can be extended to speech recognition as well if the finite number of utterances are limited to a finite number of words where each word has slightly different probability of occurring (obtained from a language model). This is shown by :

The probability of a possible outcome x is then given by:

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

And where the probability that x is drawn from the ith pmf is:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  | (10) |

Finally, it is shown that is given by the multinomial distribution (Frigyik et al., 2010):

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

and follows a Dirichlet distribution and is given by:

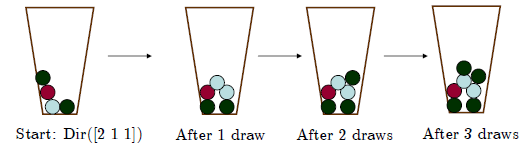
|  |  |  |
| --- | --- | --- |
|  |  | (12) |

When combined, (10) becomes:

|  |  |  |
| --- | --- | --- |
|  |  | (13) |

And finally this can be used in (9) to find the likelihood of the data. To find the optimal α that maximizes the likelihood, the log of (13) is maximized using an optimization technique such as the EM algorithm.

A few of the more common techniques to generate samples from a Dirichlet distribution include the Polya's urn and stick-breaking methods. In the Polya's urn method, *αi* balls of color *I* (where *I = 1, 2, …, k*) are placed in an urn. Each iteration consists of drawing a ball randomly from the urn and then replacing it along with an additional ball of the same color. After placing a set number of balls in the urn the proportions of balls of each color *I* is found. Frigyik et.al.(2010) provides a nice visualization and summary of this process:



**"Step 1:**  Set a counter n=1. Draw X1 ~ α/α0. (Note that α/α0 is a non-negative vector whose entries sum to 1 so it is a pmf.

**Step 2:**  Update the counter to n+1. Draw *Xn+1 | X1, X2, … , Xn ~ αn /αn0* where

and αn0 is the sum of the entries αn. Repeat this step an infinite number of times. "

Figure 2: A diagram depicting the Polya's urn method.

As mentioned in the previous chapter, this can easily applied to language modeling. If the different colored balls represent possible words, then the probability of a word occurring can easily be found using the steps mentioned in .

The stick-breaking method depicts a Dirichlet distribution as a stick of length one broken into *k* different pieces. For this example, assume *k = 3*. A temporary variable, *{ui}*, is used and initially generated from a Beta distribution, . This represents the first break in the stick and therefore the remaining length of the stick is given by 1-u1. Next, is generated from Beta (α2, α2) and set *q2 = (1-u1) u2*. For the case of *k=3* the final vector is then *qi = [u1, (1-u1)u2, 1 - u1 - (1-u1)u2]*. Frigyik et. al. (2010) summarizes these steps and generalizes them for any value of *k*:

**"Step 1:** Simulate and set . This is the first piece of the stick. The remaining piece has length 1 - u1.

**Step 2:** For 2 ≤ j ≤ k-1, if j-1 pieces with lengths, u1, u2, …, uj-1, have been broken off, the length of the remaining stick is . We simulate

and set . The length of the remaining part of the stick is =

**Step 3:**  The length of the remaining piece is qk."

Figure 3: An outline of the steps used for the stick-breaking approach to DPs

One possible interpretation of the stick-breaking approach is given by choosing a number of Gaussian mixtures during acoustic modeling. Initially, a single distribution that best models the data is used. Each successive mixture used to model the data is given increasingly less weight and eventually an ideal number of mixtures is achieved.

### 2.1.2 Dirichlet Processes

A Dirichlet process is a stochastic process parameterized using an arbitrary base measure, *H*, and a concentration, α (where α > 0). Unlike the Dirichlet distribution's αi, where i = 1, 2, … , k and whose values are discrete, Dirichlet processes are parameterized by a continuous function across the sample space, α(χ). Drawing from a Dirichlet process yields a discrete random distribution. One way to interpret a Dirichlet process is to compare it to a dartboard (Frigyik et al., 2010). If we assume the dartboard to be the infinite sample space and a realization from the Dirichlet process is a distribution characterized by an infinite set of darts of various lengths. The length of each dart represents the weight given to that distribution. These weights are constrained such that the sum of all weights must be equal to one. More formally:

|  |  |  |
| --- | --- | --- |
|  |  | (14) |

Where B represents a set of the infinite sample space, pk represents the darts' weights, δyk(B) indicates the location of the kth dart (δyk = 1 if yk ϵ B and (δyk = 0 otherwise), and .

Another nice example is given by Frigyik et. al. (2010) where a Dirichlet Process is compared to polling a group of people several times for their favorite color. Depending on someone's mood, each person may give a different answer depending on the day. We can therefore treat each person as a separate pmf and model the probability of a given color being their favorite. Since the colors they can choose are not specified, an infinite sample space, i.e. colors, is modeled over another infinite sample space, people. If the Dirichlet process is portioned in a finite manner, though, it can be modeled as a Dirichlet distribution. In the example of modeling people's favorite colors, the range of all (infinite) possible responses can be categorized into M distinct choices. This infers a Dirichlet distribution parameterized by α(*Β*i) for i = 1, 2, …, M where

|  |  |  |
| --- | --- | --- |
|  |  | (15) |

and is the indicator function of *B*i.

To generate samples from a Dirichlet process, the same procedures for a Dirichlet distribution are used with a few slight changes. The first, described above, is the Polya urn method (also known as the Chinese restaurant process). The major difference from the above method is that there are now an infinite number of ball colors and the urn is initially empty. Or, in the case of the language model example mentioned above, the original set of words is unknown and there are infinitely many possibilities. Frigyik et. al. (2010) describe the steps for generating samples below by first setting *n=1* and then:

"**Step 1:** Pick a new color with probability distribution α/α(χ) from the set of infinite ball colors. Paint a new ball that color and add it to the urn.

**Step 2:** With probability pick a ball out of the urn, put it back with another ball of the same color, and repeat Step 2. With probability , go to Step 1"

Figure 4: An outline of the process of sampling from a DP using the Polya Urn method

Thus a random sequence of colors, or words, (X1, X2, …) is drawn from the set (y1, y2, …, yk, …, y∞). (Frigyik et al., 2010) continues to explain that if there are mk occurrences of K different colors, (y1, y2, …, yk) after the first n draws then:

|  |  |  |
| --- | --- | --- |
|  |  | (16) |

To generate samples from a Dirichlet process using the stick-breaking method the distributions {pk, yk} must be characterized. To simplify matters though, {θk, yk} will be used instead (measure theory can be used to prove the relationship between {pk, yk} and {θk, yk} but is beyond the scope of this work). The following steps are followed:

**Step 1:** Let p1 = θ1. Thus the stick (originally of length 1), now has a length of 1- θ1.

**Step 2:** Break off another piece of the remaining stick θ2.. Now, p2 = θ2(1-θ1) and the length of the remaining stick is (1-θ1) (1-θ2). If this is repeated k times, then the remaining stick's length is given by and .

**Step 3:** Finally the probability distribution can be found by using (14).

Figure 5: An outline of the process of sampling from a DP using the stick-breaking approach

## 2.2 Variational Inference

The process of making clustering or classification predictions using complex multivariate distributions is often intractable. Instead inference algorithms are used to analyze samples from distributions in order to generate an approximation which in turn can be used to make the necessary prediction. Markov chain Monte Carlo (MCMC) methods such as Gibbs sampling are extremely popular for their low mathematical complexity (Neal, 1991; Paisley, 2010; Rasmussen, 2000). However, these methods are sensitive to sample step size and can require significant computation time since large numbers of samples are required to sufficiently approximate a distribution. Moreover, the number of steps to ignore during the "burn in" phase, i.e. the number of steps necessary to find the correct region of the sample space, must be carefully chosen. Consequently, newer variational inference algorithms have been introduced which do not require any sampling but still yield comparable results with more reasonable computation times.

The problem variational inference solves is simply described by Eisner (2011): the calculations to generate predictions using a posterior distribution *p(y|x)*, where *y* represent outputs and *x* is an input, is intractable. The solution is to use a simpler distribution that makes more independence assumptions, *q(y)*, to approximate *p(y|x)*. This can be handled as an optimization problem, i.e. minimizing an objective function, where an optimal *q* is found from a set of distributions *Q={q1, q2, …, qm)*. This inherently requires more complex computation but is still much faster than MCMC methods.

In the next few subsections, the three variational inference algorithms used in this work are described. Each are fairly similar and have all shown to produce results comparable to Gibbs sampling but are significantly faster (Harati Nejad Torbati et al., 2012).

### 2.2.1 Accelerated Variational Dirichlet Process Mixtures (AVDPM)

Mean-field techniques (Blei & Jordan, 2006) are commonly used in variational inference. Mean-field approximation, in its basic essence, makes assumptions about a joint distribution's independencies. For example, a joint distribution *p(a,b,c, …)*, is approximated as (Eisner, 2011):

|  |  |  |
| --- | --- | --- |
|  |  | (17) |

In these methods, the Kullbach-Liebler divergence acts as the objective function and is minimized to create the best set of *T* variational distributions (i.e. approximations), *Q = {q1(y), q2(y), …, qT(y})*, of the posterior, *p(y|x)*.

Following Kurihara et. al.'s (2006) work, a DPM makes the following assumptions:

(1) There are an infinite number of components (i.e. distributions), , that are taken independently from a prior with hyperparameter , .

(2) represents an infinite number of stick lengths that are drawn from another prior with hyperparameters α, . These represent mixing weights for the mixtures for *i = 1, … , ∞*

(3) An observation model is used to generate a data point from distribution η.

(4) For dataset , each is generated from by assigning a component label where .

(5) The DPM has a set of latent variables, *W = {H, V, Z}* and hyperparameters *θ = {λ, α}*

(6) Class prediction relies on finding *p(W|X, θ)* which is intractable so a new set of variational distributions *q(W; ϕ)* is defined as

|  |  |  |
| --- | --- | --- |
|  |  | (18) |

In contrast to (Blei & Jordan, 2006), who truncate (18) with the stipulations that *L=T*, , and AVDPMs allow L to approach infinity but set any variational distributions equal to their priors for any *i > T*. Furthermore, binary trees known as kd-trees, are used. These trees consist of a root node that contains all data points and child nodes which contain subsets of their parent's data points. The number of times a parent node is split can be adjusted which allows the user to control the tradeoff between computational resources and model accuracy. Thus, with some management, AVDPMs can be used on larger corpora at the expense of some degradation in accuracy (Kurihara, Welling, & Vlassis, 2006).

### 2.2.2 Collapsed Variational Stick Breaking (CVSB) and Collapsed Dirichlet Priors (CDP)

As mentioned in previous sections, a common model for DPs is the Chinese restaurant process in which data points are partitioned, i.e. assigned to a specific table or group. This unfortunately leads to problems when using variational inference that makes use of approximate, factorized distributions . As Kurihara et. al. (2007) points out, the assignments *z1 = 1, z2 = 1, z3 = 2* represent the same partition if *z1 = 3, z2 = 3, z3 = 2*, i.e. (1,2)(3), which indicate that dependencies exist in the assignment variables. Kurihara further explains that this problem can be avoided if the approximations are formed in the space of cluster labels rather than partitions. The remainder of this subsection follows Kurihara et. al.'s (2007) work on two of these methods: CVSB and CDP.

A truncated stick-breaking model is used in CVSB where the joint density is given by:

|  |  |  |
| --- | --- | --- |
|  |  | (19) |

where is a beta distribution for *v*. It is important to note that if cluster labels are changed in this model, the probability of (19) will also be altered. CDP, on the other hand, does allow for interchangeable cluster labels by limiting the model to *K* clusters and by setting a symmetric prior *D* on the mixing weights, . The equivalent joint density for CDP is then:

|  |  |  |
| --- | --- | --- |
|  |  | (20) |

In either CVSB or CDP, the mixing weights can be marginalized out to produce the collapsed density (Kurihara, Welling, & Teh, 2007):

|  |  |  |
| --- | --- | --- |
|  |  | (21) |

But where the distributions over cluster labels *p(z)* are different for the CVSB and CDP models due to the interchangeability (or lack thereof) of cluster labels. Finally, the lower bound used in variational inference is given by:

|  |  |  |
| --- | --- | --- |
|  |  | (22) |

where *θ* represents {η,*v*}, {η, π}, or {η} depending on whether (19), (20), or (21) is used and the variational distributions are given by:

|  |  |  |
| --- | --- | --- |
|  |  | (23) |

|  |  |  |
| --- | --- | --- |
|  |  | (24) |

where and are marginalized out of (23) and (24) respectively if using the collapsed density shown in (21).

# CHAPTER 3

# DATA & EXPERIMENTS

## 3.1 Data

In this section the language differences between Mandarin Chinese and English are introduced followed by a description of the LDC's CALLHOME corpora.

### 3.1.1 Mandarin Chinese vs. English: A Language Comparison

With the advent of China's growing economic development over the past few decades, Mandarin has become a language that speech researchers are increasingly interested in. A study has shown that the world contains approximately 350 million native English speakers compared to Mandarin's one billion (K.-J. Chen et al., 1994). Moreover, there are as many English language learners in China as there are native speakers in the world (Chien et al., 1995). Although these statistics are somewhat dated and the populations have since grown dramatically, they highlight the increasingly pressing demand for high performance Mandarin speech recognition systems.

Unfortunately, there is a large disparity between speech recognition performance for English and Mandarin datasets. This is particularly apparent for conversational telephone speech (CTS) data sets. One study has shown that two comparable CTS corpora, one English and one Mandarin, yielded word error rates (WERs) of 17.5% and 42.7% respectively (Schwartz et al., 2004). In the following paragraphs major differences between the two languages will be discussed to highlight some of the key difficulties in Mandarin speech recognition.

While words in English are created using a phonetic alphabet, Chinese words consist of one or more syllables represented by Chinese characters. Approximately 8000 characters compose as many as 200k of the most common words in Mandarin. Furthermore, unlike English, whose words are segmented, i.e. separated by a space, Mandarin text is often not. Because of this, it is up to the user to determine which characters belong to a given word. Although this is not always the case, it can greatly affect a speech recognizers performance if transcripts are not properly parsed.

Each character is a monosyllabic morpheme and is assigned a specific tone. While tone generally represents a speaker's emotion in English, Mandarin's tones specify word meaning, i.e. two characters with the same phonetic syllable but with different tones represent two different words. Four distinct tones and one neutral tone exist in Mandarin Chinese:

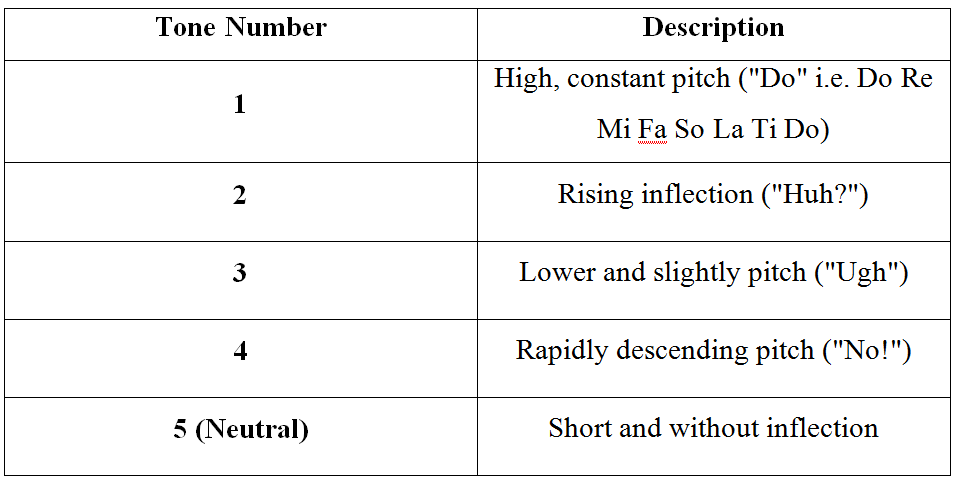


Figure 6: A description of the different tones in Mandarin Chinese. Words in ( ) indicate examples of English words that share similar sounds to Mandarin tones.

Although each character has a set tone associated with it, Mandarin is highly susceptible to effects from coarticulation and thus a character's tone can change depending on the surrounding context. One common example of this is if a word consists of two consecutive characters that have the third tone in which case the first character's tone is changed to the second tone. Another such example occurs when the word "不" ("bu4" - this form represents the phonetic translation of the character followed by its tone), the character used to negate meaning precedes another character with fourth tone. In such an example the tone of "不" is changed to the second tone.

Furthermore, Mandarin has just over 400 unique syllables ignoring tone (or about 1300 with tones) compared to English's 10,000 syllables (Gu, Hirose, & Fujisaki, 2006). Consequently, Mandarin has an extremely large number of homophones compared to English. This creates the need for a more developed language model during the decoding phase to be able to discern characters correctly.

Creating a strong language model is difficult though since Mandarin has an extremely flexible grammatical structure. Long phrases are often interchangeable with shortened versions consisting of only one or two characters. For example, the phrase for Beijing University, "北京大学" (Bei3 jing1 da4 xue2) is often abbreviated to "北大" (bei3 da4). Another, more significant example is shown in Figure 3.1.1a (Lee, 2006). This depicts an example of how word order can easily be altered without affecting the overall sentence meaning.

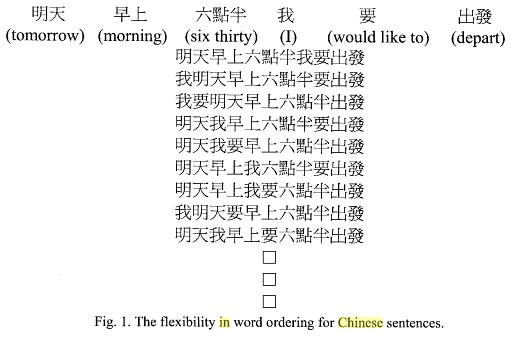


Figure 7: An example showing several equivalent sentences using different word orders (Lee, 2006)

This can reduce the efficacy of N-gram type language models and therefore require incorporating more advanced techniques such as neural networks or random forests (Oparin et. al., 2010).

Thus, the possible lack of a segmented lexicon, the need to model tones as well as phonetic sounds, the high number of homophones, and the incredibly flexible grammatical structure of Mandarin make it extremely difficult for Mandarin speech recognizers to perform as well as their English counterparts. Many of these challenges directly affect a typical speech recognizer's decoding phase, specifically the language model. The goal of this work is to investigate whether the three nonparametric algorithms are able to narrow the gap in performance by classifying the phones directly from the data.

### 3.1.2 LDC's CALLHOME Corpora

The goal of this project is to compare and contrast the performance of AVDPMs, CVSB, and CDPs on phone classification for both non-tonal and tonal languages. For such a comparison, it is necessary to use data for both languages that was recorded in comparable environments and contain similar content. For this reason, the CALLHOME English and Mandarin corpora are selected for this task. Although conversational telephone speech is not normally used for phone classification due to noisy background and the speech's weak grammatical structure, this choice of data will provide unique insight into the robustness of DPMM classification methods to data from more difficult environments.

The CALLHOME English corpus consists of 120 unscripted conversations that last up to 30 minutes. All calls were made from North America but only 30 were domestic while the remaining 90 calls were made to people overseas. Overall, 200 calls were recorded, 80 of which were designated to the training set while 20 calls were assigned to a development set and 20 for an evaluation set. The remaining 80 calls are held by the LDC for future speech recognition benchmark tests. The audio was sampled at a rate of 8kHz using a 2 channel µlaw format and encoded using Cambridge's SHORTEN format. Ten minute segments from calls from each of the training and developments sets were chosen to be transcribed while five minute segments were selected from calls from the evaluation set. Because of the extremely long audio and transcript files, the data was parsed into several smaller files corresponding to single utterances by using time stamp information from the original transcripts.

The CMU7 lexicon was used as a basis for this task and augmented with additional vocabulary. Despite this augmentation, a few words in the transcripts did not exist in the lexicon. The vast majority of these instances were proper nouns, mostly from foreign languages, or exceedingly rare or idiosyncratic words. These were all added to the lexicon using a garbage phone to represent their pronunciations.

The CALLHOME Mandarin corpus was recorded under identical circumstances as its English counterpart and consisted of unscripted conversations that last up to 30 minutes. All calls originated in North America and were made to people overseas. Again, 200 calls were recorded and transcribed but 80 are reserved for future benchmarks. The remaining calls are divided into a training, development, and evaluation set consisting of 80, 20, and 20 calls respectively. An LDC Data Scholarship was awarded to this project and provided the Mandarin transcripts and lexicon which Temple University did not previously have the rights to.

## 3.2 Experimental Setup

This chapter describes the steps taken to prepare the data for our classification task followed by detailed explanations of each individual experiments for both baseline algorithms and inference algorithms.

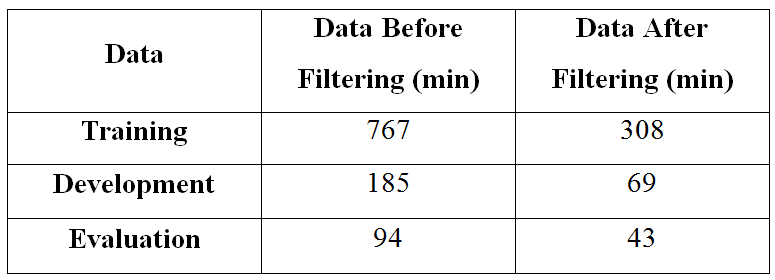
### 3.2.1 Data Preparation

#### 3.2.1.1 CALLHOME English

Before running any classification algorithms, several measures were taken to prepare the data. We begin this section first by describing the formatting used on the CALLHOME English transcripts and audio files followed by the steps taken for the CALLHOME Mandarin corpus.

The LDC distributes the CALLHOME English corpus consists of 120 calls, 80 of which are reserved for the training set, 20 for development, and the remaining 20 for evaluation. Each call is recorded as a sphere audio file and compressed using Tony Robinson's Shorten software. These files are uncompressed into a two channel ulaw format using w\_decode. Because each phone call encapsulates so much data (up to thirty minutes of conversation), the original audio files were divided into smaller audio clips corresponding to single utterances. This was accomplished using the time stamps from the CALLHOME English transcripts. Start and stop times were extracted in order to parse each call and to determine the number of speakers within a given utterance's time frame. Every clip is given a new utterance ID (in the form of originalFileName\_clip#). In several instances multiple speakers talk simultaneously which make phone classification extremely difficult without filtering to separate the audio channels. Since there is plenty of data, we choose to avoid this additional step and rather select only audio from single speakers during the training process. All other utterances from the newly created audio clips that have speaker overlap are discarded. The amount of data before and after this step are shown in Table 1 below.

Table 1: A table showing the amount of CALLHOME English data before and after filtering out segments of overlapping speakers.



The original transcripts from the LDC are also reformatted. This includes the following:

(1) Time stamps and speaker ID's are removed from the original transcripts.

(2) Markers for proper names, non-lexemes (i.e. "uh", "um", "er", etc.), and utterance comments from transcribers are removed.

(3) In some cases, certain audio files are distorted such that transcribers were forced to guess at the actual words. Rather than attempting to correctly identify these instances, they were simply replaced by a generic word, *"{garbled}"*, whose pronunciation is given by a garbage phone and treated as noise.

(4) When a speaker makes a noise such as a laugh, cough, sneeze, sigh, etc., it is replaced by a generic word, *"{sound}"*, whose pronunciation is given by a garbage phone and treated as noise.

(5) If a speaker voices a partial, unfinished word it is replaced by the generic word *"{partial}"*, whose pronunciation is given by a garbage phone and treated as noise.

(6) Markers are removed from a word spoken from a language other than English. Many of these words already exist in the dictionary (i.e. "hola", "c'est la vie", etc.) and do not require special attention.

(7) Any mispronounced words marked in the original files are transcribed as the intended word instead.

(8) Any utterances that contain only distorted and or partial words, or sounds are removed.

By changing these conventions we avoid trying to identify uninformative noise and focus on speech instead. With these formatting steps complete, we save the utterance and their new ID's in a .dot format (i.e. *'utterance (utteranceID)'* ).

The CMU7 dictionary is used as the primary dictionary for this task although it is manually augmented with additional word-forms and their pronunciations to supplement existing lexemes, as well as a few other common words (typically proper nouns). A list is compiled of every unique word found in the transcripts and compared to those in the augmented dictionary. Any words that are not found in the dictionary are added and given a pronunciation using a garbage phone (i..e. equivalent to noise). Approximately 500 words, consisting primarily of uncommon proper nouns and idiosyncratic words, are assigned the garbage phone out of a total of the total 8545 words in the compiled dictionary. Including the garbage phone, sil, and sp, these experiments have a total of 42 unique phones.

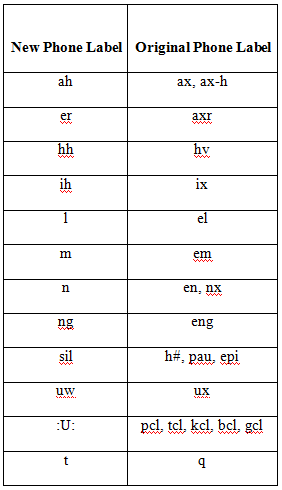
Once the dictionary was compiled and transcripts reformatted, all of the newly created audio clips are converted into 13 MFCCs and their first and second derivatives using the Hidden Markov Model Toolkit (HTK) (generating the typical 39 feature format). Only audio clips that contain no speaker overlap are selected for training. Sixteen Gaussian mixture models are then trained for monophones and passed to a Viterbi based time alignment in order to extract the start and stop times (measured in seconds, not frames) of individual phones from the audio data. Because it is difficult to classify phones that have multiple durations every instance of every phone is broken into a 39x3 matrix of features using a 30-40-30 averaging technique, i.e. each column represents the average features for a given portion of the phone's duration. This involves first converting a phone's start and stop time to frames and then averaging the features from the first 30% of the frame duration, the middle 40%, and the final 30%. The phone's total frame duration is added as an additional feature such that our final feature vector is a 40x3 matrix. In this way, we format the data such that every instance of each phone has a consistent number of features which will be used for our classification algorithms.

In the following subsections the algorithms used as baseline results are discussed. All programming is completed in Matlab in order to take advantage of its wide variety of built in functions. Since this is a novel experiment for the CALLHOME corpora, neural networks, random forests, and K-nearest neighbor algorithms were chosen to generate baseline results.

#### 3.2.1.2 TIMIT

The TIMIT corpus was selected for the expansive amount of phone recognition work completed on it. The corpus has also conveniently pre-generated a phone alignment which can be used to parse the MFCC feature vectors in the same manner outlined for the CALLHOME English corpus. Thus, there is no need to reformat the dictionary or transcripts for this corpus. All of the remaining preparation is identical to that used for the CALLHOME English corpus. For the sake of consistency, the phones are reduced to a set of 42 labels identical to those used in CALLHOME English. To do so, the following phones are changed:

Table 2: A chart of the converted phone labels for the TIMIT corpus



# CHAPTER 4

# PRELIMINARY EXPERIMENTS & RESULTS

## 4.1 Baseline Experiments

In order to verify the effectiveness of AVDPM, CVSB, and CDP models a few baseline experiments were conducted using simple and conventional algorithms. These include neural networks, random forests, and K-nearest neighbors. Moreover, experiments that utilize the CALLHOME corpora focus on a wide variety of tasks but do not involve direct phone classification as presented in this work. Consequently, to ensure the validity of these baseline experiments, the algorithms are run on TIMIT, another corpus known for its extensive amount of published work on phone classification. In this section, the basic setups of the neural network, random forest, and K-nearest neighbor algorithms are described. All of the results in the following section are found by using only the data from the training set of each corpus (i.e. only preliminary work has been done for tuning purposes and the final misclassification errors on the test data have not yet been completed).

The neural network algorithm is run using a single hidden layer as can be seen in an example architecture below:

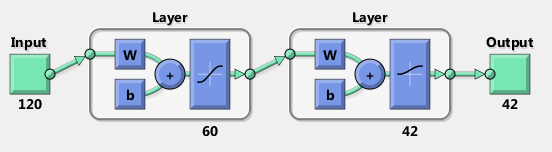


Figure 8: An example of a neural network architecture used in this work.

The network consists of an input layer of 120 features (corresponding to an unrolled version of the averaged MFCCs mentioned above), a hidden layer with a varying number of neurons, and an output layer whose size corresponds to the number of possible phones. The parameters are trained with tangent sigmoid transfer functions between each layer and by using resilient back propagation with stopping criteria set by 1000 maximum epochs or 20 consecutive epochs that fail to improve performance. For each network, the algorithm is run for five iterations. Initially, only the training set of the CALLHOME English corpus was used for these experiments. The training set is randomly divided during each iteration into 49% training, 21% validation, and 30% testing. The mean and variance of the training portion is used to normalize all three partitions such that they have zero mean and a standard deviation of one. The networks that yielded best performance were run for an additional 40 iterations and an average misclassification error and confusion matrix are generated.

The random forest algorithm, requiring the number of trees as its single parameter, is run for various values to determine a reasonable tradeoff between performance and computation time. The training set from the CALLHOME English is portioned into 70% training and 30% testing to determine a misclassification error rate.

In K-nearest neighbors, the data does not require partitioning since predictions are determined by finding any other samples that have a minimum Euclidean distance to a given test sample. The number of nearest neighbors, *K*, is set to five and a weighted average (using Euclidean distance as weights) of the best samples' labels is used to determine the final misclassification error.

## 4.2 Preliminary Results

### 4.2.1 TIMIT

Table 3: A table of misclassification errors generated from the random forest algorithm for varying numbers of trees.

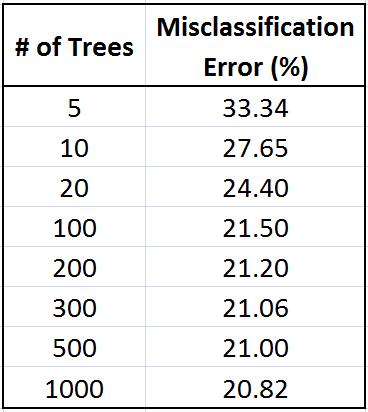


Table 4: A table of misclassification errors generated from the neural network algorithm for varying numbers of neurons in the hidden layer.

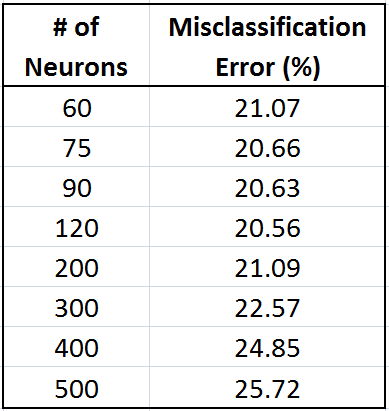
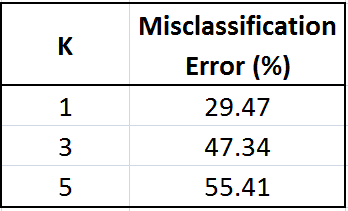


Table 5: A table of misclassification errors generated from the K-nearest neighbor algorithm for varying values of K. For K > 1 misclassification is determined by a weighted average using a Euclidean distance metric.



### 4.2.2 CALLHOME English

Table 6: A table of misclassification errors generated from the random forest algorithm for varying numbers of trees.

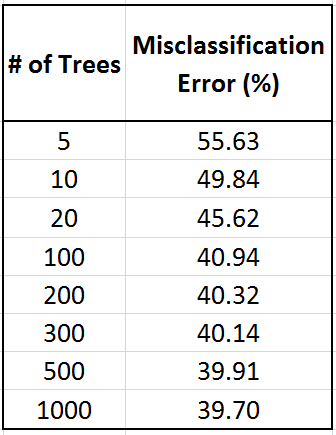


Table 7: A table of misclassification errors generated from the neural network algorithm for varying numbers of neurons in the hidden layer.

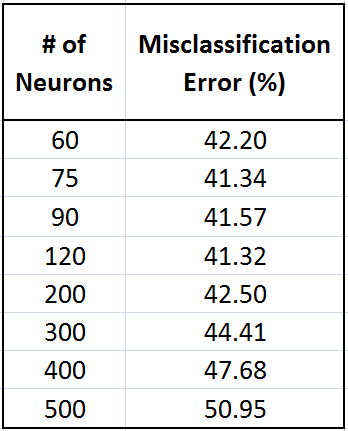
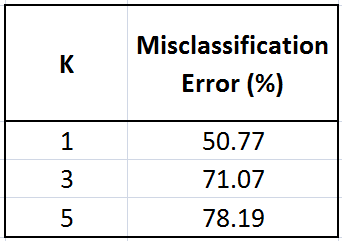


Table 8: A table of misclassification errors generated from the K-nearest neighbor algorithm for varying values of K. For K > 1 misclassification is determined by a weighted average using a Euclidean distance metric.



# CHAPTER 5

# EXPECTATIONS & FUTURE WORK

## 5.1 Expected Outcomes

Although this is novel research and final results have not yet been determined it is worth discussing some of the expected outcomes of these experiments. First, if the number of phones is kept consistent between corpora, the misclassification error rates should be relatively close for the two CALLHOME datasets (TIMIT may yield significantly different results due to a different recording environment). If this proves true, it will be confirmed that these algorithms are well suited to Mandarin speech recognition. This is an important discovery since Mandarin speech recognizers often suffer from inferior performance due to the extremely high number of homophones and flexible grammatical nature of the language. More specifically, this would alleviate some of the error rate caused by the sparseness of N-gram based language models.

These algorithms easily adapt and evolve as more data is provided to the system unlike some of the more traditional methods used in phone classification. Moreover, as mentioned in Chapter 2, all three methods are designed to run much faster than typical nonparametric methods. Thus, these algorithms will still be deemed extremely useful even if they only yield comparable results to the baseline methods mentioned above.

## 5.2 Timeline for Future Work

A significant amount of work is still required to complete this research. A timeframe for the remaining portions of this thesis along with expected completion dates are outlined below:

**December:**

(1) Generate Phone Alignments for CALLHOME Mandarin:

Transcripts and lexicons will need to be formatted in order to generate the time alignments necessary to parse MFCC features for phone recognition.

(2) Generate Baseline Results for CALLHOME Mandarin:

The same algorithms used for TIMIT and CALLHOME English will be used to generate baseline results for CALLHOME Mandarin. This should require few if any alterations to existing algorithms.

(3) Generate final misclassification error rates on test data for all corpora:

After tuning models using test data, a final misclassification error will be determined for all three corpora. This will include investigating the surprisingly low performance of K- nearest neighbors.

**January:**

(1) Find misclassification errors using variational inference algorithms:

Preexisting Matlab functions will be used to implement these algorithms but will require some minor revisions to be applied to this work. This requires tuning models and finding final misclassification errors on test sets.

(2) Complete an initial, complete draft of this thesis

This will include final error rates from all baseline experiments and variational inference algorithms along with a comprehensive analysis of these results.

**February:**

(1) Work on publications:

Ideally, at least two publications will be generated from this work. Possible submissions include INTERSPEECH 2013 (March 18 deadline) and the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2014.

(2) Determine further directions of this research:

Additional questions inevitably arise during the course of research and this period will be devoted to addressing any new questions or problems that arise during this period.

**March-May:**

(1) Finish a complete and final draft of this thesis and finish work on any outstanding publications.

(2) Successfully defend this thesis.

# REFERENCES

Bahl, L., Brown, P., de Souza, P., & Mercer, R. (1986). Maximum mutual information estimation of hidden Markov model parameters for speech recognition. *ICASSP  ’86. IEEE International Conference on Acoustics, Speech, and Signal Processing* (Vol. 11, pp. 49–52). Institute of Electrical and Electronics Engineers. doi:10.1109/ICASSP.1986.1169179

Belkin, M., & Niyogi, P. (2003). Using manifold structure for partially labeled classification. *Proceedings of Neural Information Processing Systems* (p. 2003).

Blei, D. M., & Jordan, M. I. (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1), 121–143. doi:10.1214/06-BA104

Chen, K.-J., Tseng, C.-Y., Lyu, R., Chien, L.-F., Wang, H.-M., Shen, J.-L., Lin, S.-C., et al. (1994). Golden Mandarin(II)-an intelligent Mandarin dictation machine for Chinese character input with adaptation/learning functions. *Proceedings of ICSIPNN  ’94. International Conference on Speech, Image Processing and Neural Networks* (pp. 155–159). IEEE. doi:10.1109/SIPNN.1994.344943

Chen, S. F., & Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4), 359–393. doi:10.1006/csla.1999.0128

Chien, L.-F., Hwang, S.-H., Hsieh, Hu.-Y., Yang, R.-C., Bai, B.-R., Weng, J.-C., Yang, Y.-J., et al. (1995). Golden Mandarin (III)-a user-adaptive prosodic-segment-based Mandarin dictation machine for Chinese language with very large vocabulary. *1995 International Conference on Acoustics, Speech, and Signal Processing* (Vol. 1, pp. 57–60). IEEE. doi:10.1109/ICASSP.1995.479272

Eisner, J. (2011). High-Level Explanation of Variational Inference. Retrieved February 11, 2012, from http://www.cs.jhu.edu/~jason/tutorials/variational.html

Frigyik, B., Kapila, A., & Gupta, M. (2010). *Introduction to the Dirichlet Distribution and Related Processes*. Seattle, Washington, USA. Retrieved from https://www.ee.washington.edu/techsite/papers/refer/UWEETR-2010-0006.html

Gu, W., Hirose, K., & Fujisaki, H. (2006). Comparison of Perceived Prosodic Boundaries and Global Characteristics of Voice Fundamental Frequency Contours in Mandarin Speech. *Chinese Spoken Language Processing* (pp. 31–42). doi:10.1007/11939993\_8

Harati Nejad Torbati, A. H., Picone, J., & Sobel, M. (2012). Applications of Dirichlet Process Mixtures to speaker adaptation. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4321–4324). Philadelphia, Pennsylvania, USA: IEEE. doi:10.1109/ICASSP.2012.6288875

Heigold, G., Ney, H., Schluter, R., & Wiesler, S. (2012, November). Discriminative training for automatic speech recognition. *IEEE Signal Processing Magazine2*, 60–71.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohammed, A., Jaitly, N., Senior, A., et al. (2012, November). Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal Processing Magazine*, 83–97.

Juang, B.-H., Chou, W., & Lee, C.-H. (1997). Minimum classification error rate methods for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 5(3), 257–265. doi:10.1109/89.568732

Kurihara, K., Welling, M., & Teh, Y. W. (2007). Collapsed Variational Dirichlet Process Mixture Models. *Twentieth International Joint Conference on Artificial Intelligence*.

Kurihara, K., Welling, M., & Vlassis, N. (2006). Accelerated Variational Dirichlet Process Mixtures. *NIPS*.

Larochelle, H., Erhan, D., Courville, A., Bergstra, J., & Bengio, Y. (2007). An empirical evaluation of deep architectures on problems with many factors of variation. *Proceedings of the 24th international conference on Machine learning - ICML  ’07* (pp. 473–480). New York, New York, USA: ACM Press. doi:10.1145/1273496.1273556

Lee, L.-S. (2006). Large Vocabulary Continuous Speech Recognition for Mandarin Chinese: Principles, Application Tasks and Prototype Examples. *Advances In Chinese Spoken Language Processing* (pp. 125–152). World Scientific. doi:10.1142/9789812772961\_0006

Mallapragada, P. K., Jin, R., & Jain, A. (2010). Non-parametric Mixture Models for Clustering. *Structural, Syntactic, and Statistical Pattern Recognition* (pp. 334–343). Springer Berlin Heidelberg.

Neal, R. (1991). *Bayesian Mixture Modeling by Monte Carlo Simulation*. doi:10.1.1.51.387

Oparin, I., Lamel, L., & Gauvain, J.-L. (2010). Improving Mandarin Chinese STT system with Random Forests language models. *2010 7th International Symposium on Chinese Spoken Language Processing* (pp. 242–245). IEEE. doi:10.1109/ISCSLP.2010.5684903

Paisley, J. (2010). *Machine Learning with Dirichlet and Beta Process Priors: Theory and Applications*. Duke University.

Picone, J. (2012). HTK Tutorials. Retrieved from http://www.isip.piconepress.com/projects/htk\_tutorials/

Povey, D., & Woodland, P. C. (2002). Minimum Phone Error and I-smoothing for improved discriminative training. *IEEE International Conference on Acoustics Speech and Signal Processing* (pp. I–105–I–108). IEEE. doi:10.1109/ICASSP.2002.5743665

Rasmussen, C. E. (2000). The Infinite Gaussian Mixture Model. *In Advances in Neural Information Processing Systems* (pp. 554–560). MIT Press.

Renals, S. (2010). Hierarchical Bayesian Language Models for Conversational Speech Recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8), 1941–1954. doi:10.1109/TASL.2010.2040782

Sainath, T., Ramabhadran, B., Nahamoo, D., Kanevsky, D., Compernolle, D. Van, Demuynck, K., Gemmeke, J. F., et al. (2012, November). Exemplar-Based Processing for Speech Recognition. *IEEE Signal Processing Magazine*, 98–113.

Samet, H. (2008). 3. Similarity searching: Indexing, nearest neighbor finding, dimensionality reduction, and embedding methods for applications in multimedia databases. *2008 19th International Conference on Pattern Recognition* (pp. 1–1). IEEE. doi:10.1109/ICPR.2008.4760937

Saon, G., & Chien, J.-T. (2012, November). Large Vocabulary Continuous Speech Recognition Systems. *IEEE Signal Processing Magazine*, 18–34.

Schwartz, R., Colthurst, T., Duta, N., Gish, H., Iyer, R., Kao, C.-L., Liu, D., et al. (2004). Speech recognition in multiple languages and domains: the 2003 BBN/LIMSI EARS system. *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing* (Vol. 3, pp. iii–753–6). Cambridge, Massachusetts, USA: IEEE. doi:10.1109/ICASSP.2004.1326654

Seide, F., Li, G., & Yu, D. (2011). Conversational Speech Transcription Using Context-Dependent Deep Neural Networks. *INTERSPEECH*.

Tam, Y.-C., & Schultz, T. (2005). Dynamic Language Model Adaptation Using Variational Bayes Inference. *Proceedings of INTERSPEECH* (pp. 5–8).

Young, S. (1996). A review of large-vocabulary continuous-speech. *IEEE Signal Processing Magazine*, 13(5), 45. doi:10.1109/79.536824