**Non-Parametric Bayesian Approaches for Acoustic Modeling**

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

**A Dissertation Proposal**

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

**In Partial Fulfillment of the Requirements**

**for the Degree of Doctor of Philosophy**

**By**

**Amir Hossein Harati Nejad Torbati**

**February, 2013**

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

**Dr. Joseph Picone**

**Department of Electrical and Computer Engineering**

**College of Engineering**

**Thesis Advisor**

|  |  |
| --- | --- |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Dr. Iyad Obeid Department of Electrical and Computer Engineering College of Engineering**  **Committee Member** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Dr. Chang-Hee Won Department of Electrical and Computer Engineering**  **College of Engineering  Committee Member** |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Dr. Slobodan Vucetic**  **Department of Computer and Information Sciences**  **College of Science and Technology**  **Committee Member** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Dr. Marc Sobel Department of Statistics**  **Fox School of Business and Management Committee Member** |

|  |
| --- |
| © |
| By |
| Amir Harati |
| 2013 |
| All Rights Reserved |

|  |
| --- |
| © |
| By |
| John Smith |
| 2013 |
| All Rights Reserved |

ABSTRACT

The goal of Bayesian analysis is to reduce the uncertainty about unobserved variables by combining prior knowledge with observations. A fundamental limitation of any statistical model, including Bayesian approaches, is the inability of the model to learn new structures. These models are referred to as parametric models. The goal of the learning process is to estimate the correct values for these parameters. The accuracy of the parameters improves with more data but the model’s structure remains fixed and therefore new observations will not affect the overall complexity (e.g. number of parameters in the model). One way to address this problem is to define many different models and then to select the most likely one based on the observed data. However, the model selection process is computationally expensive, often requires large amounts of data and is critically dependent on a meaningful selection criterion.

Recently, nonparametric Bayesian methods have become a popular alternative to Bayesian approaches. In such approaches, we do not fix the complexity a priori (e.g. the number of mixture components in a mixture model) and instead place a prior over the complexity (or model structure). This prior usually biases the system towards sparse or low complexity solutions. This helps to control the number of parameters in the model yet allows the structure to be learned during a data-driven training process. Therefore models can adapt to new data encountered during the training process without distorting the modalities it has learned on the previously seen data.

In speech recognition technology, we deal with the complexity problem at many levels. Examples in acoustic modeling include the number of states and the number of mixture components in a hidden Markov model (HMM). Also, the number of models (and parameter-sharing between these models) is often determined as a compromise between complexity and computational issues. In language modeling, we must estimate the probabilities of unseen events in very large but sparse N‑gram models. Nonparametric Bayesian modeling has been previously used to smooth such N‑gram language models.

In this proposal, our goal is to investigate the application of nonparametric Bayesian modeling to acoustic modeling. Three important problems fundamental to the acoustic modeling component of a large vocabulary speaker independent continuous speech recognition system are addressed: (1) automatic discovery of sub-word acoustic units; (2) statistical modeling of sub-word acoustic units; and (3) supervised training algorithms for nonparametric acoustic models. We propose a nonparametric Bayesian algorithm based on an ergodic Hierarchical Dirichlet Process (HDP) HMM (HDP-HMM) that automatically segments and clusters the speech signal. We apply this algorithm to the problems of automatic discovery of acoustic sub‑word units and generation of a pronunciation lexicon.

A new type of HDP-HMM is presented that preserves the useful left-to-right properties of a conventional HMM, yet still supports automated learning of the structure and complexity from data. We will introduce a nonparametric Bayesian algorithm for training these models for continuous speech recognition that allows us to infer different HDP-HMM models and segment the training data simultaneously. This eliminates the need for manual sub-word segmentation of the data. Moreover, a nonparametric Bayesian approach is introduced that replaces the phonetic decision tree used in state of the art speech recognizers to tie triphone states.

Our nonparametric Bayesian approaches improve a model’s flexibility and its ability to adapt to previously unseen events. This is critical when training speech recognition systems on imperfect data where there might be channel mismatches or noisy transcriptions. We expect our proposed solutions for these well-known acoustical modeling problems to outperform conventional approaches without increasing complexity. This will enable a new generation of speech recognition systems capable of being trained on vast archives of found data (e.g., YouTube) and to enable the rapid development of speech recognition systems in new languages.

Table of Contents

[ABSTRACT iii](#_Toc348276837)

[list of figures vii](#_Toc348276838)

[list of tables viii](#_Toc348276839)

[Chapter 1 INTRODUCTION 1](#_Toc348276840)

[Chapter 2 NONPARAMETRIC BAYESIAN 6](#_Toc348276841)

[2.1 Dirichlet Process 6](#_Toc348276842)

[2.2 Hierarchical Dirichlet Process 14](#_Toc348276843)

[2.2.1 Stick-Breaking Construction 15](#_Toc348276844)

[2.2.2 Chinese Restaurant Franchise 16](#_Toc348276845)

[2.3 HDP-HMM 18](#_Toc348276846)

[2.3.1 CRF with Loyal Customers 20](#_Toc348276847)

[2.3.2 Inference Algorithm 21](#_Toc348276848)

[Chapter 3 ACOUSTIC MODELING 31](#_Toc348276849)

[3.1 Acoustic Modeling in sate of the Art Automatic Speech Recognizers 31](#_Toc348276850)

[Chapter 4 Speech Segmentation and Acoustical Unit Learning 35](#_Toc348276851)

[4.1 Problem statement 35](#_Toc348276852)

[4.2 Related Works 36](#_Toc348276853)

[4.3 Proposed Approach 36](#_Toc348276854)

[Chapter 5 LEFT-TO-RIGHT HDP-HMM MODELS 39](#_Toc348276855)

[5.1 Problem Statement 39](#_Toc348276856)

[5.2 Related works 39](#_Toc348276857)

[5.3 Proposed Approach 40](#_Toc348276858)

[Chapter 6 Nonparametric Bayesian training 44](#_Toc348276859)

[6.1 Problem statement 44](#_Toc348276860)

[6.2 Proposed approach 46](#_Toc348276861)

[6.2.1 Training left-right HDP-HMM 46](#_Toc348276862)

[6.2.2 Tying states 47](#_Toc348276863)

[Chapter 7 RESEARCH PLAN 49](#_Toc348276864)

[Chapter 8 CONCLUSION 51](#_Toc348276865)

[REFERENCES CITED 53](#_Toc348276866)

list of figures

**No table of figures entries found.**  
In your document, select the words to include in the table of contents, and then on the Home tab, under Styles, click a heading style. Repeat for each heading that you want to include, and then insert the table of contents in your document. To manually create a table of contents, on the Document Elements tab, under Table of Contents, point to a style and then click the down arrow button. Click one of the styles under Manual Table of Contents, and then type the entries manually.

list of tables

**No table of figures entries found.**  
In your document, select the words to include in the table of contents, and then on the Home tab, under Styles, click a heading style. Repeat for each heading that you want to include, and then insert the table of contents in your document. To manually create a table of contents, on the Document Elements tab, under Table of Contents, point to a style and then click the down arrow button. Click one of the styles under Manual Table of Contents, and then type the entries manually.

1. INTRODUCTION

For the past few decades the focus of speech recognition research, much like other pattern recognition applications, was on developing better models of speech and better algorithms to estimate the parameters of these models (Rabiner, 1989). Some of the most successful statistical modeling approaches (e.g. hidden Markov models) and some of the most efficient estimation algorithms (e.g. Baum-Welch) are among the results of this research. However, during the past few years, despite of the availability of vast computational power and large amounts of data, the performance of the state of the art systems has been at best marginal. One of the main reasons for this is the limited modeling capabilities of the underlying technology.

Generally, determining model complexity is among the most difficult problems in pattern recognition. An oversimplified model cannot describe the data and a very complex model generally is prone to over-fitting. Model selection techniques usually need a huge amount of data and are computationally expensive (Bishop, 2007). Any selection methodology needs a criterion for selecting a preferred model. There is not a widely accepted consensus on this criterion (Ghahramani, 2010). Hence, this process is application specific and involves searching through a discrete space (e.g., a combinational search over models). The final result is sensitive to the criterion used to guide the search and often application specific.

Nonparametric Bayesian methods provide a mathematically elegant framework that allows inference of model structure and complexity without diluting the purity of modes or clusters (Sudderth 2006). In a fully Bayesian framework, hyperparameters (i.e. parameters that control the complexity of the model) along with model parameters can be learned automatically from the data. In other words, the data can speak for itself. Unlike in a model selection problem, the optimization of the model parameters is a continuous optimization problem and hence is more tractable. Hierarchical modeling can be used to increase the power of nonparametric Bayesian models (Teh, et al., 2006). First, hierarchical modeling provides better control over the large number of degrees of freedom that exist in nonparametric models (Teh & Jordan, 2010). Second, it makes it possible to use simple building blocks (e.g., a Dirichlet process) to construct models that have rich probabilistic structures (Teh & Jordan, 2010).

In speech recognition, like other pattern recognition applications, selection of an appropriate model complexity and the optimal hyperparameters are among the most difficult and time-consuming parts of the process, and has a direct effect on performance of the system. Model complexity is not just confined to the complexity of an individual hidden Markov model (HMM) or mixture model but it also includes the overall complexity of the system. A typical state of the art speech recognition system has a large number of degrees of freedom, often utilizing over 10M parameters that must be estimated during training. These parameters must be estimated using a complicated bootstrapping process. A major goal of this paper is to propose a formalization of this process in which a nonparametric extension is constructed within a hierarchical framework.

Among many possible hierarchical Bayesian nonparametric models, in this dissertation we only consider the hierarchical Dirichlet process (HDP) (Teh, et al., 2006). The motivation for defining an HDP can be understood better by considering the problem of modeling related grouped data. In this problem we are interested in modeling several groups of related data using mixture models. In a traditional nonparametric Bayesian solution we can use a Dirichlet process (DP) prior for each group. This solution can indeed solve the problem by modeling each group using a mixture model, but the resulting mixtures are not linked.

In many applications, for a variety of reasons to be explained later, we want to share components among groups. For example, in topic modeling application, each document can be regarded as a group (Teh et al., 2004). Moreover, under an exchangeability assumption (e.g. bag of words) (Teh & Jordan, 2010), we can model each document as a probability distribution across topics (Teh et al., 2004). In this case, each topic is a probability distribution across words. It should be noted that a document could have several topics with different strengths. Because the number of topics is unbounded the problem fits within the nonparametric framework. Specifically, it is an example of a Dirichlet process mixture (DPM) model. However, if we want different documents to share topics then we have to define another layer that links these individual DPMs together. In other words, there should be a common pool that contains all possible topics. Each document can be generated by first randomly selecting topics from this common pool and then generating words according to the topic specific distributions. The details of this model will be discussed in following sections.

HMMs are a time series generalization of a mixture model (Rabiner, 1989). As stated above, a DPM can also be considered as a nonparametric extension of a mixture model. Therefore, we expect to have a similar structure for nonparametric HMMs. An analogous structure exists, but it is based on an hierarchical Dirichlet process (Teh et al., 2006) and therefore is referred to as an HDP-HMM. To understand the motivation behind this definition we can imagine a segmentation problem where the number of segments is not known a priori and each segment can be represented by one state of an HMM. A parametric HMM (with specified number of states a priori) cannot find the segments since the number of segments is not known. One solution is to use model comparison and select the model with maximum likelihood. This solution is prone to overfitting (the likelihood will always increase as we increase the number of segments). Therefore an heuristic cost function is needed to determine when to stop the process so we avoid overfitting. Alternatively, a nonparametric HMM can learn the number of segments and therefore does not need any form of heuristic tuning or cost function.

In this dissertation, we propose application of the nonparametric Bayesian approach to the acoustic modeling problem in speech recognition. In an earlier preliminary study, we have studied the application of a Dirichlet Process Mixture (DPM) model to the speaker adaption problem (Harati et al., 2012). In that study we have shown that DPM can successfully replace the regression tree in a maximum likelihood linear regression (MLLR) algorithm. Figure 1 compares the word error rate (WER) for monophone models for both a DPM and a regression tree. From this figure, we can see that DPM improves performance over the MLLR algorithm by about 10%. This study was one of the motivations for the current proposal since it demonstrates that the nonparametric Bayesian framework is promising for speech recognition problems.

In the second part of this dissertation, nonparametric Bayesian methods used in the subsequent sections will briefly be introduced. In Chapter 3 we introduce the acoustic modeling problem. After these introductory sections, we will focus on three primary applications of nonparametric Bayesian methods that are the subject of this dissertation.

****

Figure – A comparison of regression tree and DPM based clustering (Harati et al., 2012). Inference was implemented using an ADVP algorithm.

In Chapter 4, we study the segmentation problem. Segmentation is among the most fundamental problems in speech and signal processing. In this section, an approach for automatically segmenting speech utterances will be proposed. Despite its importance, a segmentation algorithm by itself is not extremely useful. Hence, in this section we also propose to apply the nonparametric Bayesian approach to segment and cluster speech utterances in order to automatically discover acoustic sub-word units. This could replace more traditionally used units like phonemes. Finally, we propose a method to generate a lexicon to map words into these sub-word units.

In Chapter 5, we turn our attention to the problem of nonparametric Bayesian modeling of individual sub-word units. This problem is traditionally tackled using left-to-right HMMs with a fixed number of states and with a predetermined number of Gaussians per state in a state of the art speech recognizer. In this section we propose a new topologically constrained HDP-HMM, which we call left-to-right HDP-HMM with HDP emissions, and its corresponding inference algorithm. The proposed model will learn both the number of states and number of mixtures automatically from the data.

Finally in Chapter 6, we present an approach for training a complete speech recognizer within the nonparametric Bayesian framework. This approach will use the left-to-right HDP-HMMs to model each individual sub-word unit. Moreover, it can be used to train continuous speech recognizers using only utterance level transcriptions. We also introduce a data‑driven nonparametric Bayesian approach to replace phonetic trees for state tying. In Chapter 7, the research plan will be proposed and in Chapter 8 some conclusions and future directions will be discussed.

1. Nonparametric Bayesian Approaches

Parametric approaches have been used in machine learning and pattern recognition applications since mid-1900’s. The phrase “parametric” was coined by statistician Jacob Wolfowitz (1942):

“Most of these developments have this feature in common, that the distribution functions of the various stochastic variables which enter into their problems are assumed to be of known functional form, and the theories of estimation and of testing hypotheses are theories of estimation of and of testing hypotheses about, one or more parameters . . ., the knowledge of which would completely determine the various distribution functions involved. We shall refer to this situation . . . as the parametric case, and denote the opposite case, where the functional forms of the distributions are unknown, as the non-parametric case.”

These approaches provide reasonable performance with a fixed amount of complexity (Gelman, 2004). For some time, it was generally believed that such models could be arbitrarily improved through the use of larger data sets (Huang, 1992).

However, performance gains have leveled off for a variety of reasons, including the complex recording conditions embodied in these massive data sets. Using more data to train the models improves the estimation of individual parameters but it is not usually translated to overall better performance since the model itself is fixed. Nonparametric non-Bayesian approaches have been also used (e.g. decision trees) but it has been shown (Breiman et al., 1984; Bramer, 2007) that they are prone to the overfitting of the training data. It is also difficult to control the complexity of these models in a rigorous manner. A number of ad hoc algorithms (e.g. pruning in decision trees) have been used instead (Bramer, 2007).

Nonparametric Bayesian approaches make it possible to learn the model structure (and the degree of the complexity) from the data without the risk of over-fitting the model to the observations by biasing the model toward simpler structures. With the availability of big data resources (e.g. online videos at sites such as YouTube) these models becomes even more important since they can use the data more efficiently. Like all Bayesian approaches, nonparametric Bayesian approaches use Bayes Rule to combine the prior distributions with the observations (e.g. likelihoods) to estimate the posterior distribution for the models. This posterior implicitly contains the structure we have learned from the data. Depending on how we define the prior distribution we can define an unlimited number of nonparametric Bayesian models. In this dissertation we are interested in a very specific type of prior based on the Dirichlet Process, and therefore we restrict our discussion to this form of prior.

Mixture models are a very popular basic building block in many machine learning applications and also provide a framework for more complex models. For example, mixture models are used extensively in HMMs. A Dirichlet distribution is a parametric prior used frequently in Bayesian approaches involving mixture models. In this chapter we will review the Dirichlet distribution and its application in Bayesian modeling, including the use of mixture distributions. We then will introduce a nonparametric counterpart in which we replace the Dirichlet distribution with a Dirichlet Process (DP). Dirichlet processes, historically, are among the first priors used in nonparametric Bayesian modeling (Teh, 2010). Beside their applications in mixture modeling problems they also have been used as a building block for many other nonparametric models including the Hierarchical Dirichlet Process (HDP) (Teh & Jordan, 2010) and the infinite HMM (iHMMs) (Beal, 2002) which are also known as HDP-HMMs (Teh et al., 2006; Fox et al., 2011). These form the basis for the work presented in this dissertation.

* 1. The Dirichlet Distribution

Consider a random variable *x* over a finite *K*-dimensional space *X={1,2,…,K}*. The probability mass function in this space can be represented by a *K*-dimensional vector  where  and . This vector can characterize a multinomial distribution that is defined as:



Equation can be used to calculate the probability of selecting a category or class among K possible classes. In this definition *mk* is the number of observations of category. Given observations, π can be estimated using a maximum likelihood (ML) approach (Sudderth, 2006). ML is a point estimate which means it does not estimate the posterior distribution; instead it just estimates an important point (e.g. mean) of this distribution. In the case of a multinomial distribution, the result is empirical frequencies of discrete categories (e.g. for a specific observation the probability of each category can be calculated by dividing the number of samples in that category by the total number of samples):



However, if the number of data points is not large enough, ML estimation of π will have a high variance (e.g. the estimated value varies around the real value by a large amount) and some categories even may have a zero probability. Estimating zero probability for an event means that that we believe that event will never happen. In practice many events of interests are rare but with some positive probability of happening and therefore estimating a zero probability for their occurrences is a bad estimation.

An example of this problem is the problem of N-gram modeling of phonemes. For instance, consider the problem of finding the probability of 3-grams of phonemes occurring in English. Given a finite amount of text, many of 3-grams will never be observed. If we model the problem using a multinomial distribution and use an ML approach to estimate the occurrence probabilities, the result will contains many zeroes or unrealistically small numbers. The estimated value for the probability of each 3-gram (parameters in question) will be a point estimate (in this case the mean) of the underlying distribution for these parameters.

An alternate approach is to infer  using a Bayesian approach (Gelman, 2004). We should define a prior on  in such a way that a posterior inferred by multiplying the prior and likelihoods remain in the same family of distributions. In Bayesian statistics, this particular property is named conjugacy (Gelman, 2004) and the prior is called a conjugate prior for the likelihood. For example, the conjugate prior for the Gaussian distribution with known covariance is itself a Gaussian distribution. Consider N Gaussian observations *x1,x2,…,xN*. Suppose the covariance matrix Σ is known. We can place a normal prior over the mean with mean *μ0* and covariance Σ0. This prior is indicated with Norm(*μ0,*Σ0). After observing *N* data points the posterior over mean is given by (using Bayes rule):



In the case of a multinomial distribution, the conjugate distribution is a Dirichlet distribution (Teh, 2010):



In this definition Γ is the gamma function and defined by:

A Gamma function is an extension of factorial function to real and complex numbers (Milton et al., 1974). The concentration parameter, *α*, in equation  is proportional to inverse of variance (Teh, 2010). Therefore, equation places a probability distribution over π which itself is a probability distribution.

A Dirichlet distribution, like all other discrete distributions, can be represented by two sets of parameters: locations of the impulse functions and their corresponding weights. The impulse functions are often referred to as atoms. For example, in a binomial distribution, there are exactly 2 atoms, *x*=0 and *x*=1, and two corresponding weights, P(*x*=0) and P(*x*=1).

The mean of Dirichlet distribution is given by:



If the parameter α is set symmetrically (e.g. set to equal values for all K dimensions):



then the variance of the distribution is given by (Gelman et al., 2004):



Equation clearly shows that the variance of the Dirichlet distribution is inversely proportional to the concentration parameter α. In other words, large concentration parameters correspond to distributions concentrated around the mean (e.g. if used as a prior then this implies the most likely value for the prior is around its mean, which is also equivalent to having a high confidence in the mean of the prior).

Given some data we can obtain a posterior distribution for π using Bayes rule (by multiplying the prior and likelihood):



By substituting from equation and equation we can write:



Equation unlike equation gives a distribution over π. The parameters of this distribution are learned from both the observed data and the prior assumptions.

From equation we can see *αk* acts as a pseudo observation. A pseudo observation is a term used to weight our belief in the prior knowledge. Mathematically it acts as an actual observation though it is not really observed. Hence, we refer to it as a pseudo observation for category *k*. The total number of pseudo observations, *α0*, is equal to the sum of *αk*:



By considering this fact and equation we can see the variance of the estimation decreases by increasing the number of pseudo observations. The predictive distribution for a new observation, which is the distribution of unseen data given observed data and priors, can be written using equations  and :



An explanatory example of the above discussion can be seen in language modeling. A language model assigns a probability to a document. One simple unigram language model (e.g. bag of words) is a multinomial language model. If we define the language model for a document (D) as πD then for a sequence of independent terms we can write:



In this equation each is a multinomial distribution.

As a simple example, consider a search engine application where we have some number of documents and a goal of finding the most relevant documents given a “query” of several terms. For each document D, we have to compute equation . To compute this probability we have to compute πD for all terms in the query. If we use the maximum likelihood solution in equation , we might get zero probability for a document if one of the terms does not exist in the document. Obviously, it is not an acceptable solution for a search engine application. On the other hand, estimating πD using a Dirichlet distribution as shown in equation will solve this problem since it always gives a nonzero probability even if some of the terms are not present in a document.

* 1. Dirichlet Process

A Dirichlet process (DP) is a distribution over distributions, or more precisely over discrete distributions. Formally, a Dirichlet process, DP(α,G0), is “defined to be the distribution of a random probability measure G over Θ such that for any finite measurable partition (A1,A2,…,Ar) of Θ the random distribution (G(A1),…,G(Ar)) is distributed as finite dimensional Dirichlet distribution” (Teh et al., 2006):



In this definition α is the concentration parameter and is proportional to the inverse of the variance; G0 is the base distribution and is the mean of the DP (e.g. ).

A constructive definition for a Dirichlet process is given by Sethuraman (1994) which is known as the Griffiths, Engen and McCloskey (GEM) construction, or the stick-breaking construction. This construction explicitly shows that draws (or in other words samples) from a DP are discrete with probability one:



Starting with a stick of length one, we break it at *ʋ1* and assign the length to *β1*. Then we recursively break the remaining part of the stick and assign the corresponding lengths to *βk*. In this representation *β* can be interpreted as a random probability measure over positive integers and is denoted by *β~GEM(α)*.

Another representation of the DP is the Polya urn process. In this approach, we have to consider i.i.d. draws from a DP and consider the predictive distribution over these draws (Teh et al., 2006):



In the urn interpretation of equation , we have an urn with several balls of different colors in it. We draw a ball and put it back in the urn and add another ball of the same color to the urn. With probability proportional to *α* we draw a ball with a new color. To make the clustering property more clear, we should introduce a new set of variables that represent distinct values of the atoms (e.g. observed balls). Let  be the distinct values and *mk* be the number of associated with . We now have:



Another useful interpretation of equation is the Chinese restaurant process (CRP). In a CRP we have a Chinese restaurant with infinite number of tables. A new customer  comes into the restaurant and can either sit around one of the occupied tables with probability proportional to the number of people already sitting there (*mk*) or initiate a new table with probability proportional to *α*. In this metaphor, each customer is a data point and each table is a cluster. Let *zi* indicate the cluster associated with *ith*observation. A CRP is the interpretation of the predictive distribution:



As this equation shows new data points (customers) tend to sit around crowded tables and eat the food served on that table (in other words, customers are social). However, sometimes, a customer initiates a new table (e.g. cluster) and orders new food.

As an illustrative example, consider the problem of automatic acoustic unit discovery. Given a set of segments (assume that data is pre-segmented) the goal is to cluster the segments into some units. However, the number of units is not known a priori. If we think of each “segment” as a customer then we see CRP acts as a prior distribution over the clusters.

A Dirichlet Process Mixture (DPM) is defined as:



In this model, observations are sampled from an indexed family of distributions denoted by *F*. If *F* is assumed to be Gaussian then the result is an infinite Gaussian mixture model. In the case of the acoustic unit discovery example, a Gaussian distribution is too simple to model a speech segment accurately and therefore better models are needed (e.g. Gaussian mixtures or dynamic models). It should be noted that a CRP induces priors that prefer simpler models (e.g. tables with many customers but fewer number of tables in a restaurant) which means number of discovered units would be much smaller than the number of observed segments.

* 1. Hierarchical Dirichlet Process

A Hierarchical Dirichlet Process (HDP) is the natural extension of a Dirichlet process for problems with multiple groups of data. Usually, data is split into *J* groups a priori. For example, consider a collection of documents. If words are considered as data points, each document would be a group. We want to model data inside a group using a mixture model. However, we are also interested to tie groups to each other, i.e. to share clusters across all groups. Let’s assume that we have an indexed collection of DPs with a common base distribution *{Gj}~DP(α,G0).* Unfortunately this simple model cannot solve the problem since for continuous G0 different Gj have no atoms in common. The solution is to use a discrete G0 with broad support. In other words, G0 is itself a draw from a Dirichlet process.

An HDP is defined by (Teh & Jordan, 2010) equation :



In this definition *H* provides prior distribution for the factor θji. The parameter *γ* governs the arbitrarity of G0 around *H* and *α* controls the variability of Gj around G0. *H*, *γ* and α are hyperparameters of the HDP. Equation is just one representation of an HDP. Another representation can be obtained by introducing an indicator variable:



Figure 2 shows graphical models for both of these representations.

* + 1. Stick-Breaking Construction

Because G0  is a Dirichlet distribution it has a stick-breaking representation:



where  and. Since support of Gj is contained within the support of G0 we can write a similar equation to for Gj:



Then we have:





* + 1. Chinese Restaurant Franchise

The Chinese restaurant franchise (CRF) is the natural extension of Chinese restaurant process for HDPs. In a CRF, we have a franchise with several restaurants and a franchise wide menu. The first customer in restaurant *j* sits at one of the tables and orders an item from the menu. Other customers either sit at one of the occupied tables and eat the food served at that table or sit at a new table and order their own food from the menu. Moreover, the probability of sitting at a table is proportional to the number of customers already seated at that table. In this metaphor, restaurants correspond to groups and customer  in restaurant *j* corresponds to  (customers are distributed according to Gj). Tables are i.i.d. variables  distributed according to G0 and finally foods are i.i.d. variables  distributed according to *H*. If customer *i* at restaurant *j* sits at table *tji* and that table serves dish *kji*, we will have. Each restaurant represents a simple DP and therefore a cluster over data points. At the franchise level we have another DP but this time clustering is over tables.



Figure - (follow Fig. 1 format) HDP representation of (5) (b) Alternative indicator variable representation (The et al., 2004)

Next, we can introduce several variables that will be used throughout this paper: *njkt* is the number of customers in restaurant *j*, seated around table *t*, and who eat dish *k*; *mjk* is the number of tables in restaurant *j* serving dish *k* and *K* is the number of unique dishes served in the entire franchise. Marginal counts are denoted with dots.

A CRF can be characterized by its state which consists of the dish labels , the tables  and dishes . As a function of the state of the CRF, we also have the number of customers, , the number of tables, , customer labels  and table labels  (Teh & Jordan, 2010). The posterior distribution of G0 is given by:



where  is the total number of tables in the franchise andis the total number of tables serving dish . We can define the posterior for *Gj*:



where  is the total number of customers in restaurant j and  is the total number of customers in restaurant *j* eating dish *k*.

Conditional distributions can be obtained by integrating out Gj and G0 respectively. By integrating out Gj from we obtain:



and by integrating out G0 from we obtain:



A draw from can be obtained using and a draw from can be obtained using .





From and we see that the posterior of G0 is a mixture of atoms corresponding to dishes and an independent draw from DP(γ,H). Gj is a mixture of atoms at and an independent draw from  (Teh & Jordan, 2010).

* 1. HDP-HMM

Hidden Markov models (HMMs) are a class of doubly stochastic processes in which discrete state sequences are modeled as a Markov chain (Rabiner, 1989). In the following discussion we will denote the state of the Markov chain at time  with  and the state-specific transition distribution for state *j* by *πj*. The Markovian structure means. Observations are conditionally independent given the state of the HMM and are denoted by.

HDP-HMM is an extension of HMM in which the number of states can be infinite. The idea is relatively simple; at each state zt we should be able to go to an infinite number of states so the transition distribution should be a draw from a DP. On the other hand, we want reachable states from one state to be shared among all states so these DPs should be linked together. The result is an HDP. In an HDP-HMM each state corresponds to a group (restaurant) and therefore, unlike HDP in which an association of data to groups is assumed to be known a priori, we are interested to infer this association. A major problem with original HDP-HMM is the state persistence. HDP-HMM has a tendency to make many redundant states and switch rapidly among them (Teh et al., 2006). This problem is solved by introducing a sticky parameter to the definition of HDP-HMM (Fox et al., 2011). Equation shows the definition of a sticky HDP-HMM with unimodal emissions;is a sticky hyper-parameter and generally can be learned from data. Original HDP-HMM is a special case with. From this equation we can see for each state (group) we have a simple unimodal emission distribution. This limitation can be addressed using a more general model defined in . In this model, a DP is associated with each state and a model with augmented state (zt,st) is obtained. Figure 3 shows a graphical representation.





* + 1. CRF with Loyal Customers

The metaphor for the Chinese restaurant franchise for a sticky HDP-HMM is a franchise with loyal customers. In this case each restaurant has a special dish that is also served in other restaurants. If a customer xt is going to restaurant *j* then it is more likely that he eats the specialty dish zt=j. His children xt+1 also go to the same restaurant and eat the same dish. However, if xt eats another dish () then his children go to the restaurant indexed byand more likely eat their specialty dish. Thus customers are actually loyal to dishes and tend to go to restaurants where their favorite dish is the specialty.

* + 1. Inference Algorithms



Figure -Graphical model of HDP-HMM (Fox et al., 2011)

### Direct Sampler

In this section we present the sampler for HDP-HMM with DP emission (Fox et al, 2011). The algorithm is divided into two steps: the first step is to sample the augmented state (zt,st) and the second is to sample *β*. In order to sample (zt,st) we need to have the posterior. By inspecting Figure 3 and using the chain rule we can write the following relationship for this posterior:



The reason that we have summed over in the last line is because we are interested in calculating the likelihood for each state. This equation also tells us that we should first sample the state and then conditioned on the current state, sample the mixture component for that state. For Gaussian emissions we can write (Fox et al., 2011):







The algorithm is as follows:

1. Given a previous set of and
2. For all.
3. For each of the K currently instantiated states compute:
   1. The predictive conditional distributions for each of the  currently instantiated mixture components for this state, and also for a new component and for a new state.







* 1. The predictive conditional distribution of the HDP-HMM state without knowledge of the current mixture component.



1. Sample:



1. Sample conditioned on:



1. If k=K+1 increase the K and transform *β* as



1. Ifincrease.
2. Update the cache. If there is a state withor remove k and decrease. If remove the component *j* and decrease.
3. Sample auxiliary variables by simulating a CRF:
4. For eachset mjk=0 and n=0. For each customer in restaurant *j* eating dish(), sample:



1. Increase n and if x=1 increase mjk.
2. For each,sample the override variables in restaurant *j*:



1. Set the number of informative tables in restaurant *j*:



1. Sample *β*:



1. Optionally sample hyper-parameters ϭ, γ, α and κ.

### Block Sampler

The problem with the direct assignment sampler mentioned in the previous section is the slow convergence rate since we sample states sequentially. The sampler can also group two temporal sets of observations related to one underlying state into two separate states. However, in the last sampling scheme we have not used the Markovian structure to improve the performance. In this section a variant of forward-backward procedure is incorporated in the sampling algorithm that enables us to sample the state sequenceat once. To achieve this goal, a fixed truncation level L should be accepted which in a sense reduces the model to a parametric model (Fox et al, 2011). However, it should be noted that the result is different from a classical parametric Bayesian HMM since the truncated HDP priors induce a shared sparse subset of the L possible states (Fox el al, 2011). In short, we obtain an approximation to the nonparametric Bayesian HDP-HMM with maximum number of possible states set to L. For almost all applications this should not cause any problem if we set L reasonably high. The approximation used in this algorithm is the degree L weak limit approximation to the DP (Ishwaran & Zarepour, 2002), which is defined as:



Using *β* is approximated as (Fox et al, 2010):



We can write:



And posteriors are:



In njk is the number of transitions from state *j* to state k and is the same as . Finally an orderweak limit approximation is used for the DP prior on the emission parameters:



The forward-backward algorithm for the joint sample z1:T and s1:T given x1:T can be obtained by:



The right side of equation has two parts: forward and backward probabilities (Rabiner,1989). The forward probability includes  and the backward probability includes . The forward probabilities are approximated with. Therefore, for the backward probabilities we have:



As a result we have (Fox et al, 2010) :



where for Gaussian emission for components are given by 

The algorithm is as follows (Fox et al, 2010):

1. Given the previous and.
2. For, initialize messages to 
3. Forand compute



1. Sample the augmented state (zt,st) sequentially and start from t=1:
2. Set andforand
3. For all compute:



1. Sample augmented state (zt,st):



1. Increase andand add xt to the cached statistics.



1. Sample  similar to the previous algorithm
2. Update :



1. For :
   1. Sample and:



* 1. For  sample:



1. Set and
2. Optionally sample hyper-parameters ϭ, γ, α and κ.

### Learning Hyperparameters

Hyper-parameters including ϭ, γ, α and κ can also be inferred like other parameters of the model (Fox et al. , 2010).

#### Posterior for (α + κ )

Consider the probability of data xji to sit behind table t:



This equation can be written by considering equation and . From this equation we can say customer table assignment follows a DP with concentration parameter α + κ. Antoniak (1974) has shown that if  then the distribution of the number of unique values of zi resulting from N draws from *β* has the following form:



where s(N,K) is the Stirling number of the first kind. Using these two equations the distribution of the number of tables in the restaurantis as follows:



The posterior over α + κ is as follows:



The reason for the last line is that is not a function of α + κ and therefore can be ignored.

By substitution of  and also by considering that  we obtain:

Finally by considering the fact that we have placed a Gamma(a,b) prior on α + κ we can write:



where sj can be either one or zero. For marginal probabilities we obtain:







#### Posterior of γ

Similar to the discussion for if we want to find the distribution of the unique number of dishes served in the whole franchise, we would have. Therefore for the posterior distribution of γ we can write:



By considering the fact that that prior over γ is Gamma(a,b) we can finally write:



and finally for the marginal distributions we have:







#### Posterior of ϭ

The posterior for ϭ is obtained in a similar way to α+κ. We use two auxiliary variablesand; the final marginalized distributions are:







It should be noted that in cases where we use auxiliary variables we prefer to iterate several times before moving to the next iteration of the main algorithm.

#### Posterior of ρ

By definition  and by considering the fact that the prior on ρ is Beta(c,d) and we can write:



1. ACOUSTIC MODELING

The ultimate goal of speech recognition is to map the acoustic data into word sequences. This problem can be formulated as (ref, XXXX):



In this formulation, *P(W|A)* is the probability of a particular word sequence given acoustical observations. The goal is to find a sequence *W* that maximizes this probability. *P(W)* is the language model and indicates what is the prior probability of words. *P(A)* is the probability of the observed acoustic data and usually can be ignored. *P(A|W)* is the acoustic model. Therefore generally we can divide the problem into two separate sub-problems, namely language modeling and acoustic modeling, and solve each one independently. Our focus in this research will be on the acoustical modeling problem.

* 1. Acoustic Modeling in State of the Art Automatic Speech Recognizers

In this section, we review the approach that is used to tackle the acoustic modeling problem in most state of the art Automatic Speech Recognition (ASR) systems. The basic idea for acoustic modeling is to find a mapping between word sequences and acoustic observations. In early systems (Furui, 1986), each word was modeled separately. This approach is relatively simple and works satisfactory for small vocabulary and isolated speech recognition tasks. However, it is not scalable to continuous large vocabulary tasks. The problem is related to the selected acoustic units (i.e. words). Since the number of words in a typical language is very large and increases over the time, modeling all words independently is not feasible. An alternative approach is to break down words into some finite set of units common to all possible words and then just model these units. People used different units such as phonemes (Lee, 1990), syllable (Ganapathiraju et al., 2001) and acoustically inspired units (Paliwal, 1990). Phonemes are the most popular and easiest to use units and most successful commercial systems are based on them.

After selecting the type of the units (e.g. phonemes), and corresponding dictionary or lexicon that maps words into these units, we have to select a statistical (or generally a machine learning) model for these units. Given a set of trained models and some new observations we test all models against the observations and select the model with the highest score (e.g. likelihood). The most successful models used in state of the art ASRs are left-to-right or Bakis hidden Markov models (HMMs) with Gaussian or mixture of Gaussians emissions (Rabiner, 1989). An HMM is a generalization of mixture model where latent variables are not independent of each other and are related with a Markov chain. This makes them particularly attractive to model sequential observations. Most systems use a simple HMM with some predetermined number of states (e.g. 3) for all units and also with some predetermined number of mixture components per state.

State of the art speech recognizers usually use some form of context dependent unit instead of simple context independent units. For example, phoneme-based systems usually have 42 context-independent phonemes. In order to improve the quality of models we can incorporate the left and right context and define context-dependent units (e.g. triphones). However, the number of units grows exponentially with increasing the depth of the context. For example, number of possible triphones are 42x42x42 = 74,088. This means training context dependent models faces a serious data sparsity problem. In any practical situation, many models will never have any observation and many more will have just a few examples and therefore estimated parameters will have large variances.

In fact, the resulting system will perform worse than a context independent system for moderate or even relatively large amount of training data. This problem has been tackled by tying models and states together so similar models share data that is a tradeoff between model accuracy and amount of data. The most successful approach to tie states is based on a phonetic decision tree that is a binary tree with phonetic questions attached to its nodes (Young et al., 2006). The tying is happening between corresponding states of all triphones with the same central phoneme. For each state of a phoneme a tree grown from a single node that contains all the corresponding states of all triphones for that phoneme. The tree is grown by asking phonetic questions and stopped when the number of data points in a node reaches to a minimum amount or dividing a node does not increase the likelihood significantly. After this step, we will have enough data for all states of all triphones.

Therefore a general algorithm to train acoustic models in a contemporary ASR is as follow:

* The first step is to prepare the data. We need to obtain some transcribed speech utterances and convert them into appropriate features representation (e.g. Mel-frequency cepstral coefficients –MFCC). We also need a dictionary that contains all possible words and their corresponding sub-word (e.g. phonemes) decompositions.
* The next step is to train all context independent phonetic models using the transcribed data and using Expected Maximization (EM) algorithm or equivalently Baum-Welch. This step is usually performed using the self-organizing property of HMMs; e.g. we let HMMs to segment data into different models and states.
* After training good monophone models, the next step is to clone monophones into triphones by simply copying the emission distributions and transition matrix for all triphones with same central phoneme and then train them using the available data.
* The fourth step is to tie states (as mentioned above) and train the resulted models for several more iteration using EM algorithm.

In this research, our goal is to investigate the applications of nonparametric Bayesian methods that discussed in previous section in acoustic modeling. In a typical speech recognizer as described above, there are several tasks (specially clustering, segmentation and model topology) that can be viewed as potential candidates for nonparametric Bayesian modeling.

1. Speech Segmentation and Acoustic Unit Learning
   1. Problem statement

Speech segmentation is perhaps the most fundamental process in speech recognition. The popularity of HMMs is a result of their segmentation properties. By segmentation, we mean to find boundaries between different events in speech. If the event is a word then speech segmentation is becomes a problem that is equivalent to speech recognition. If the event is a phoneme then the segmentation is equal to phoneme recognition. If the event is a speaker then the segmentation is equal to speaker recognition and diarization (Fox et al., 2011). Segmentation is implicitly used in all speech recognition tasks, however explicit applications of speech segmentation is usually limited to unsupervised problems; e.g., finding similar segments in a given speech utterance which can be hypothesized to be a particular word of interest (word spotting and spoken term detection) (Lee & Glass, 2012), or to separate speech from non-speech.

Another interesting application of speech segmentation is for acoustic unit discovery. Acoustic unit selection is a critical issue in many speech recognition applications where there are limited linguistic resources or training data available for the target language. For example, recently IARPA’s Babel program (Harper, 2011) sponsored a competition to create a speech to text system in a mystery language in one week of time using very limited resources. Though traditional context-dependent phone models perform well when there is ample data, automatic discovery of acoustic units offers the potential to provide good performance for resource deficient languages with complex linguistic structures (e.g., African click languages).

Most approaches to automatic discovery of acoustic units (Bacchiani & Ostendorf, 1999) do this in two steps: segmentation and clustering. Segmentation is accomplished using a heuristic method that detects changes in energy and/or spectrum. Similar segments are then clustered using an agglomerative method such as a decision tree. Advantages of this approach include the potential for higher performance than that obtained using traditional linguistic units, and the ability to automatically discover pronunciation lexicons.

Both of the clustering and segmentation sub-problems are good candidates for nonparametric Bayesian modeling. In the following we discussed the related works and our proposed approach.

* 1. Related Works

As mentioned in the last section, classical methods for acoustic unit discovery involve segmentation and clustering. The segmentation is implemented using dynamic programming method that incorporates a heuristic stopping criterion (Bacchiani & Ostendorf, 1999) and clustering implemented using a heuristic agglomerative method.

Recently, Lee & Glass (2012) proposed a nonparametric Bayesian approach for unsupervised segmentation of speech. A Dirichlet Process Mixture (DPM) model was used. In order to obtain phoneme-like segments, they modeled each segment using a 3-state HMM. A Gibbs sampler was employed to estimate the segment’s boundaries along with their parameters.

Another related problem is speaker diarization. In this problem, the goal is to segment the speech utterance into speaker homogeneous area. Fox et al. (2011) have used HDP-HMM model to solve this problem by modeling each speaker as a single state. It has been shown that the results are comparable to the state of the art speaker diarization systems (Fox et al., 2011).

* 1. Proposed Approach

Our approach for speech segmentation is based on HDP-HMM model. We propose to segment the speech using an ergodic HMM. In this model, each state models an acoustic unit. Figure4 demonstrates an example on some primary experiments based on this model (Harati et al., 2013). From this figure we can see the discovered boundaries approximately coincide with phoneme boundaries. Table1 compares the performance of the proposed algorithm with some other state of the art algorithms. The number of co-occurrences of segments boundaries and phoneme boundaries is called recall. The percent of declared boundaries that coincide with phoneme boundaries is called precision. A single numeric score that represents the combination of these two is referred to as the F-score. It is defined as:



From this table we can see the proposed algorithm performs particularly well on recall, which implies that it is finding boundaries that better match the reference phoneme boundaries. The improvement in recall is over 11%.

Although theoretically HDP-HMM should assign segments to their corresponding clusters automatically, our initial results show this labeling is not reliable and so we need to perform another clustering stage. We propose to investigate different clustering methods including a nonparametric Bayesian approach (e.g. DPM) for this step.

Automatically discovered units are not very useful unless we can define a dictionary that maps words into the units. Therefore the next step is to align the transcription with the discovered segments (and hence units) and generate a lexicon. We are planning to use forced alignment or using the manually transcribed data to map words into acoustic units.



Figure 4-Segmentation of a speech utterance produced through a process of automatic unit discovery is shown by overlaying the duration and index of each unit on the waveform. The height of each rectangle overlay simply indicates the index of that unit

The performance of the system will be measured into two different ways:

1. To measure the unit classification error. This will show how units modeled using our approach perform without considering errors that can be introduced in the lexicon generation step.
2. To measure word error rate (WER) for a system trained completely using our proposed units. This method of measuring performance is more interesting from a practical point of view; however, the performance will be a function of lexicon building stage too.

Table 1- The segmentation performance of the HDP-HMM model is compared to several other nonparametric approaches. HDP-HMM excels in recall while maintaining an acceptable precision.

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | **Recall** | **Precision** | **F-score** |
| Dusan & Rabiner (2006) | 75.2 | 66.8 | 70.8 |
| Qiao et al. (2008) | 77.5 | 76.3 | 76.9 |
| Lee & Glass (2012) | 76.2 | 76.4 | 76.3 |
| Proposed Approach | **86.5** | 68.5 | 76.6 |

By looking at different steps of acoustic model training from the last section we see, in this section, a nonparametric Bayesian approach is suggested for the first step (generating sub-word units and lexicon) of the general training recipe.

1. LEFT-TO-RIGHT HDP-HMM MODELS
   1. **Problem Statement**

The most important element of acoustic modeling is the statistical approach used to model sub-word units. Most state of the art ASR systems use left-to-right HMMs with Gaussian mixtures emissions to model phonetic units (Rabiner, 1989). Usually, the number of states is fixed for all models (for example to 3) and mixtures trained progressively by starting from one mixture per state and increase the number of mixtures until further increment does not improve the likelihood of the training data. Number of mixtures per state is also a fixed parameter for all states and models.

Because of the simplicity and existence of efficient algorithms, these parametric HMM models have been used extensively in many different applications. However, it is evident that setting the parameters (number of states and number of mixtures per state) and even topology of the model a priori is heuristic and experimental. Moreover, all models usually have the same structure which is not the optimum choice.

During the past few decades, there were several attempts to address some of these problems. However, because of the lack of better models and also limited computation power these efforts were not very successful. In the next section, some of these works will be reviewed.

* 1. Related Work

As discussed above, HMMs are parameterized both in their topology (e.g. number of states) and emission distributions. Most attempts to relax these parameterizations were focused on the second aspect. Bourlard (1993) and others proposed to replace Gaussian Mixture Models (GMMs) with Multilayer Perceptron (MLP). It has proven (Bourlard & Morgan, 1993) when used for classification; MLPs generate reasonable estimates of posterior distribution of the output classes conditioned on the input patterns. This hybrid HMM-MLP system works slightly better than traditional HMM-GMMs, but the gain is not practically significant to justify moving from a well-established technology to a new one with unknown problems. Moreover, most of the gain fades by using more complicated systems (e.g. speaker adaption, post processing). Another example of this approach is reported in (Lefèvre, 2003) and (Shang, 2009) where nonparametric density estimators have been used to replace the GMMs. Again the improvements are marginal at best. All of these approaches can be classified as nonparametric non-Bayesian methods. Being non-Bayesian makes them especially prone to overfitting or over-smoothing.

Henter et al. (2012) introduced a new model named Gaussian Process dynamical model (GPDM) to completely replace HMMs in acoustic modeling. The new model is nonparametric Bayesian and is based on Gaussian process and supposedly solves some of the problems traditionally associated with hidden Markov models such as duration modeling and stepwise constant evolution (Henter et al., 2012). However, this model is used only in speech synthesis and there is no result reported for speech recognition problems using this model. One of the possible, issues with this model for speech recognition is the lack of any kind recognition algorithm that could use new model at this moment (since the recognition algorithms are developed for HMM structure).

* 1. Proposed Approach

We have introduced the nonparametric Bayesian counterpart of HMMs, HDP-HMMs, previously. Therefore one natural way to extend nonparametric methods in acoustic modeling is to replace HMMs with HDP-HMMs. However, HDP-HMM is a fully ergodic model (all states are connected to each other) while in speech application we usually need a more constraint topology to model a time sequence. Especially, left-to-right topology is proved to be useful in speech recognition and similar applications (Rabiner, 1989). We are proposing a new type of HDP-HMM that is restricted in this sense. Therefore, the model is still learning its structure (number of states and possible skip transitions) while it remains within the left-to-right family of HMMs. There are two approaches to do this. The first approach is to use a regular HDP-HMM and then convert it into a left-to-right structure and the second one is to directly define a left-to-right HDP-HMM. Here we propose to develop the second approach while comparing the result with the first approach. Therefore developing a left-to-right HDP-HMM and developing the inference algorithm (by updating the block sampler for the new model) is one of the proposed contributions of this research. shows the discovered structure for phonemes */aa/* and */sh/* using the proposed model and with different amount of training data. As it can be seen from this figure as data increases the system can learn a more complex model for the same phoneme. It is also important to note that learned structure for each phoneme is different and which reflects underlying differences among phonemes.

In addition to defining the left-to-right HDP-HMM we also define HDP-HMMs with HDP emissions. HDP-HMMs defined in (Fox et al., 2011) use a Dirichlet Process Mixture (DPM) to model the emission distribution for each state. While this model has a reasonable flexibility to learn different mixture of Gaussians for each state, the statistical strength is not shared among different states of the HDP-HMM. This can become a more serious problem for left-to-right HDP-HMMs since for a given sequence generally left-to-right models can generate more number of states and as a result the available data for estimating the emission distribution would be more limited. Using an HDP structure for modeling the emission distribution will address this problem and can potentially improves the overall performance of both ergodic and non-ergodic HDP-HMMs.

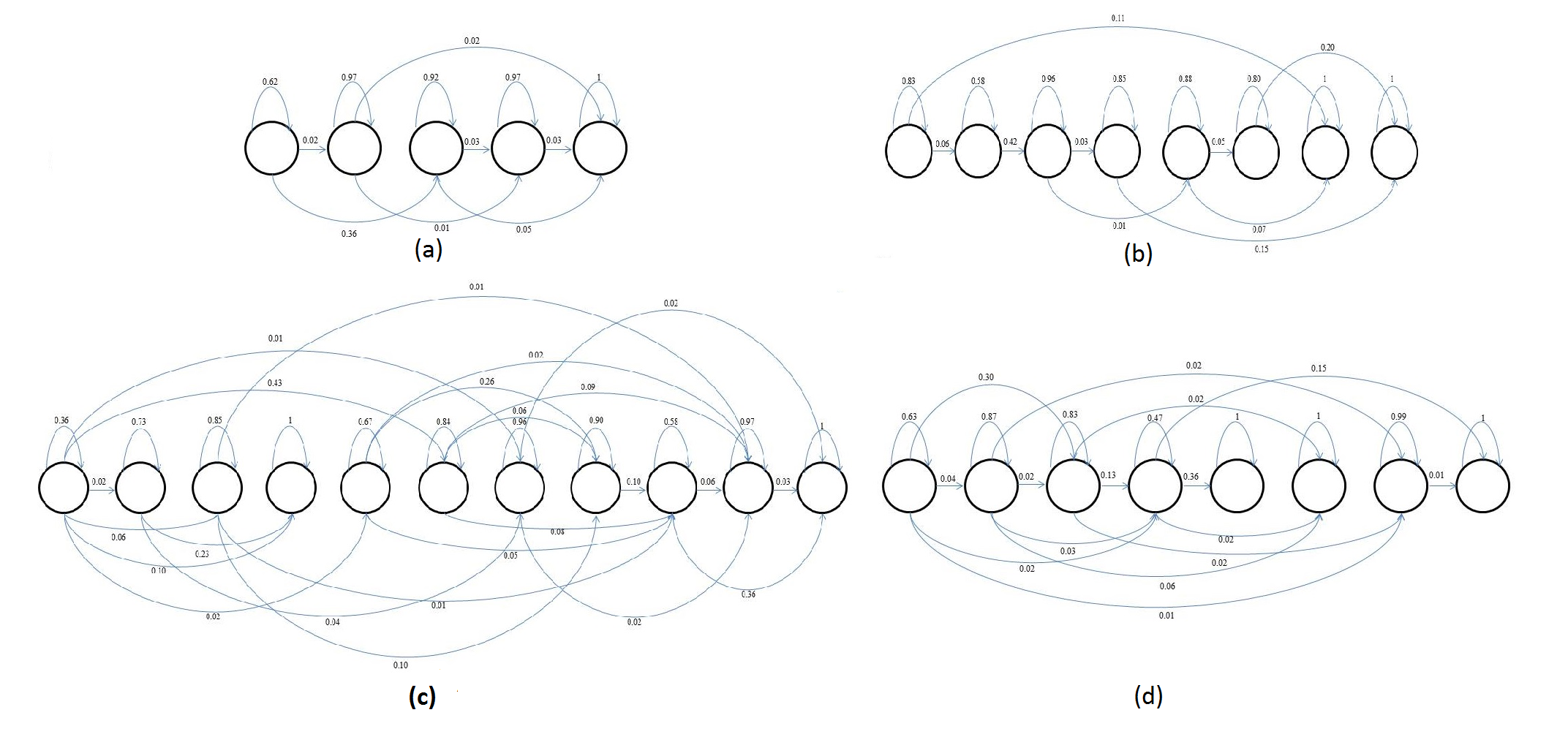


Figure - Learned model structure (without first and last dummy states) for (a) */aa/* with 175 examples (b) */sh/* with 100 examples (c) */aa/* with 2256 examples and (d) */sh/* with 1317 examples using left-to-right HDP-HMM model. The data used in this illustration extracted from training part of TIMIT data. As it can be seen, model can use more data to learn a more complex structure.

Finally, in some application like continuous speech recognition we need dummy states (states without emissions) at the beginning and end of the model; i.e. models always start from a dummy state and ends in a dummy state. This allows us to connect different HMMs in process of training continuous speech (or similar data). Adding the input dummy state is relatively straightforward; we can use the estimated initial probabilities as the output probabilities for the dummy state. However, adding the output dummy state is trickier and we have to estimate the probability of transition from each state to the output dummy state. This estimation can be calculated in a Maximum Likelihood (ML) or Bayesian framework. Both approaches will be investigated and the best one will integrated into the final left-to-right HDP-HMM with HDP emissions.

One of the intrinsic differences between ergodic HMMs and left-to-right HMMs is that the former model just one sequence of events. These events can happen in different orders but if we have two separate sequence we have to model them separately. Left-to-right HMMs, on the other hand, model ordered sequence of events with a start and an end. Therefore a single HMM can model several sequences. This also opens the door into two interesting issues: First, the state’s labels will not be arbitrarily and therefore there is no label switching problem. Secondly, as a consequence it looks like it is possible (though perhaps with some heuristics) to use a straightorward parallel inference (training) strategy. Investigating, this possibility will be another contribution of this research.

Overall, the proposed model will non-parametrically estimate the number of states and also number of mixtures per state. Since each state will have a different number of components for a Gaussian mixture that is determined directly from the data it is expected the estimated distribution be very close to the true distribution for that state. It should also be noted that unlike HMM-MLP most of the new complexity of our model is added to the training part and recognition part essentially remains the same.

For this section we will test our proposed model against regular HMM models for a phoneme recognition or isolated word recognition application. The reason to choose these applications is to focus our investigation on modeling capabilities of the proposed model. A general speech recognizer consists of many different parts (e.g. Viterbi decoder) that can alter the results. We will discuss about testing the performance of our model in continues speech recognizer in the next section.

1. Nonparametric Bayesian training
   1. Problem statement

In previous sections, we have discussed the general algorithm for training acoustic units in a state of the art speech recognizer. In Chapter 5 we introduced a left-to-right HDP-HMM model to replace ordinary HMMs in a speech recognizer. In this section we will introduce an algorithm to train these new models in a more general nonparametric Bayesian framework.

One of the interesting features of standard acoustic model training is the flat start (Young et al., 2006). Flat start means we can initialize HMM models using global calculations (e.g. means and covariance) over the training data and for each speech utterance connect its corresponding phoneme HMMs together and train them as a big HMM. This method makes it possible to avoid using phoneme level transcription (which are very difficult to produce) for training acoustic model training. Therefore we want the training procedure for nonparametric Bayesian model also has this convenient property.

Acoustic units usually trained in progressive steps, starting from very simple models and gradually training more and more complex ones. Broadly speaking the training procedure is as follow:

1. Bootstrap and flat-start: This step defines the basic models and initializes them.
2. Training monophones: This step trains monophone models.
3. Defining triphones and tie states: This step makes a much more complex model starting from simpler models (tying will be discussed in the following paragraph.)
4. Train tied state triphones.
5. Optionally use adaption techniques to adapt speaker independent models into speaker dependent models.

One of the important challenges in training more complex systems is the data sparsity problem. Context-dependent models like triphones can model acoustic events more accurately (relative to systems using context independent models.). However, each model has less data and so estimating the parameters correctly become a serious problem. Moreover, some of the triphones will never be observed in a given training dataset. To deal with these problems, people suggest tying either model or components of the models together. Tying similar models seems a good idea but it turned out that tying states is much more effective (Beulen at al., 1997). There are two mainstream approaches to tie states:

The first approach is the data driven approach:

1. First a list of all triphones existed in the dataset is produced.
2. Using monophone models trained in pervious steps, these triphones models initialized by cloning from monophone models.
3. After training these triphone models, corresponding states of all triphones with similar center phoneme grouped.
4. For each group, a clustering algorithm is applied. The clustering algorithm has two steps. First cluster similar states (based on Euclidian distance) and then merge clusters with few data points to the closest cluster.
5. Train tied models.
6. For triphones not existed in the data use back-off modeling (back off to diphones or monophones.)

Alternatively, we can use phonetic trees to cluster the data. In this case, we first group all corresponding states of all triphones with similar center phoneme. We also provide a pool of phonetic questions (e.g. is the left phoneme a stop? ). The clustering is as follow:

1. Put all states in the root node of the tree.
2. Find the best question that divide the node into two nodes and maximize the local likelihood scores.
3. Keep doing step two for all nodes until increments in the likelihood fall below a threshold. The resulted nodes are called terminal nodes and all states with in a terminal node will be tied together.
4. If data points in a node are less than a threshold combine it with its parent node.
5. Unseen models can be clustered by starting from the root and answering the question until we get to a terminal node.

Both of these approaches have been used successfully in state of the art speech recognition systems. Particularly phonetic tree based approach due to its simplicity and effectiveness has become a very successful and popular technology.

* 1. Proposed approach

In this section we describe the procedure to train left-to-right HDP-HMM models introduced in the previous section. Moreover, we describe a procedure within the nonparametric Bayesian paradigm to tie states. But it should be noted that training HDP-HMM models and tying are two separate problems. The training algorithm is independent of the sub-word unit used for speech recognition; therefore, in the following we will restrict our discussion to a phonetic based system. However, using other units (including acoustic units) is the same. The algorithm is not also confined to speech data and can be used in other similar problems.

* + 1. Training Left-to-right HDP-HMM

As discussed before, it is very important to have a training procedure that let us train our models without having phonetic level transcriptions. To this end, we introduce a variable *Zi* that contains the model id for each data point *Xi*. For a given speech utterance, the algorithm is as follows:

1. Initialize *Zi* either randomly or boot strap using a conventional system.
2. The result is several sub-sequences. Each sub-sequence will have a unique *Zi*. Therefore a sequence of *Xi* will converted into a sequence of sub-sequences *Wj*.
3. For a given sequence of data use the transcription to generate a list of models.
4. Regroup sub-sequences *Wj* based on their corresponding *Zj* and distribute each group to the corresponding HDP-HMM model (*MZi*).
5. Train each HDP-HMM using the inference algorithm. Training each left-right HDP-HMM involves several sequences of data . Fortunately, since each left-to-right HDP-HMM has a start dummy state (the first state that does not emit) using multiple sequence in inference algorithm does not change the algorithm too much.
6. After all models are trained, we should re-estimate the *Zi* for all *Xi*. This can be done using Viterbi algorithm or in a Bayesian framework.
7. After several iterations and after convergence we can fix the topology of each model.
   * 1. Tying States

After training context independent models, we can use phonetic trees to cluster states of the trained models and tie them together. Alternatively, we can use a nonparametric Bayesian approach which is closely related to data driven approach described previously. Here we describe the proposed algorithm:

1. Given the monophone models, train all existed triphones in the data set and also segment the data into different states.
2. Group all corresponding states of all triphones with the same central phoneme.
3. Each of these groups will contain all the data associated with states inside the group.
4. In each group use Dirichlet Process Mixture (DPM) to cluster the data. It is also possible to use a Hierarchical Dirichlet Process (HDP) across different groups.
5. Merge small clusters into closest cluster.
6. Use back-off modeling for unseen triphones.
7. RESEARCH PLAN

**Feb 1- March 30:**

1. Implementing left-to-right HDP-HMMs and the corresponding inference algorithm.
2. Use left-to-right HDP-HMMs to segment speech data from TIMIT.

**April 1-April 30:**

1. Experiments using left-to-right HDP-HMMs and compare to the baseline system.
2. Clustering and automatic unit discovery using segmentations produced from TIMIT dataset.

**May 1-May 31:**

1. Diagnosing possible problems related to left-to-right HDP-HMM implementation.
2. Generating the lexicon for automatic discovered units and using them in a state of the art speech recognizer and compare with baseline system.

**June 1-July 31:**

1. Wrap up the left-to-right HDP-HMM and its inference algorithm.
2. Diagnose and debugs problems related to the automatic unit discovery and lexicon building.

**August 1- September 30:**

1. Wrap up the speech segmentation and automatic unit discovery.
2. Implementing the nonparametric training framework for continues speech recognition (first section.)

**October 1- November 30:**

1. Diagnosing the training framework and run preliminary experiments.
2. Wrap up all other parts of the dissertation.

**December 1-December 30:**

1. Wrap up the first part of the training frame works and implement the second part (state tying).
2. Run experiments related to this section.

**January 1- January 31 :**

1. Wrap up the training frame work.
2. Finalize the draft of the dissertation.
3. CONCLUSION

In this paper, we investigated several applications of nonparametric Bayesian approach in acoustic modeling problem. The first application was speech segmentation and automatic sub-word discovery. For this application, we proposed to use nonparametric Bayesian methods for segmentation and clustering and also to generate a lexicon that maps words into discovered units. The second application is to use a nonparametric Bayesian model to model each sub-word unit. In this section we propose a new type of HDP-HMM named left-to-right HDP-HMM and its corresponding inference algorithm. Finally, we proposed a nonparametric Bayesian framework and training recipe to use left-to-right HDP-HMMs in a continues speech recognizer application.

Nonparametric Bayesian statistical models are one of the new promising approaches in machine learning and data modeling. It brings a good mix of flexibility and being biased toward simpler models (Occam's razor). By considering the exponential trends in data generation and computational power we can see approaches like nonparametric Bayesian are necessary tools to harness this enormous power. In this proposal, we proposed to investigate several applications in acoustic modeling problem. However, there are many directions that can be pursued in the future. One important and practical problem is to use the massive parallel processing powers (both clusters and GPUs) to accelerate the speed of inference algorithms. As of now, the main problem associated with nonparametric Bayesian approaches is their expensive computational cost. Because of this some groups already start to adapt parallel training techniques for the inference algorithm (Williamson et al., 2012) & (Suchard et al., 2010).

Another direction, especially in speech processing and similar applications, is to look into more complicated hierarchical models. Defining new models, under a Bayesian framework, is relatively straightforward. However designing efficient inference algorithm is a challenge. Also using models efficiently and intelligently in various problems might be a more difficult problem than just defining new models. For example, a new component to the proposed approach in this paper is to add another level of hierarchical clustering to cluster the data within a particular model based on acoustic similarities and differences. In such a way, we can train several instance for each model with better accuracy (for example it has been shown that having gender specific models significantly improves the recognition rate; this approach can be considered as a generalization of gender specific modeling) and since we use hierarchical framework we can tie the models and share the data in various ways. Particularly, by considering the vast amount of speech data (e.g. youtube) that became available during the past few years and by considering the huge acoustic diversity existed in this data (e.g. different speakers, environments) the importance of the this suggested direction becomes more clear.

REFERENCES CITED

Abramowitz, M., & Stegun, I. A. (1972). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York City, New York, USA: Dover Publications.

Antoniak, C. (1974). Mixtures of Dirichlet Process with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics*, 2(7), 1152–1174

Bacchiani, M., & Ostendorf, M. (1999). Joint lexicon, acoustic unit inventory and model design. *Speech Communication*, 29(2-4), 99–114.

Beal, M., Ghahramani, Z., & Rasmussen, C. E. (2002). The Inﬁnite Hidden Markov Model. *Proceedings of Neural Information Processing Systems* (pp. 577–584).

Beulen, K., Bransch, E., & Ney, H. (1997). state tying for context dependent phoneme models. *proceeding of Fifth European Conference on Speech Communication and Technolog* (pp. 1179–1182). Rhodes, Greece.

Bishop, C. (2011). *Pattern Recognition and Machine Learning* (2nd ed., p. 738). New York, New York, USA: Springer.

Bourlard, H., & Morgan, N. (1993). *Connectionist Speech Recognition A Hybrid Approach*. Springer.

Bramer, M. (2007). *Principles of Data Mining*. Springer.

Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. (1984). *Classification and regression trees* (1st ed., p. 368). Boca Raton, Florida, USA: Chapman and Hall/CRC.

Dusan, S., & Rabiner, L. (2006). On the relation between maximum spectral transition positions and phone boundaries. *Proceedings of INTERSPEECH* (pp. 1317–1320). Pittsburgh, Pennsylvania, USA

Fox, E., Sudderth, E., Jordan, M., & Willsky, A. (2010). Supplement to “ A Sticky HDP-HMM with Application to Speaker Diarization”. *The Annals of Applied Statistics*, *S*(2A), S1–S32.

Fox, E., Sudderth, E., Jordan, M., & Willsky, A. (2011). A Sticky HDP-HMM with Application to Speaker Diarization. *The Annalas of Applied Statistics*, 5(2A), 1020–1056.

Furui, S. (1986.). Speaker-independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(1), 52 – 59.

Gales, M. J. F. (1996). *model-based techniques for noise robust speech recognition*. Cambridge University.

Ganapathiraju, A., Hamaker, J., Ordowski, M., Doddington, G., & Picone, J. (2001). Syllable-based large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 9(4), 358–366.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2004). *Bayesian Data Analysis* (2nd ed.). Chapman & Hall.

Ghahramani, Z. (2010). Bayesian Hidden Markov Models and Extensions. *Proceedings of the Fourteenth Conference on Computational Natural Language Learning* (pp. 56–56). Uppsala, Sweden.

Harati, A., Picone, J., & Sobel, M. (2012). Applications of Dirichlet Process Mixtures to Speaker Adaptation. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 4321–4324). Kyoto, Japan.

Harati, A., Picone, J., & Sobel, M. (2013). Speech Segmentation Using Hierarchical Dirichlet Processes. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (p. TBD). Vancouver, Canada.

Harper, M. (2011). IARPA Solicitation IARPA-BAA-11-02. *IARPA BAA*.

Henter, G. E., Frean, M. R., & Kleijn, W. B. (2012). Gaussian process dynamical models for nonparametric speech representation and synthesis. *IEEE International Conference on Acoustics Speech and Signal Processing* (pp. 4505– 4508). Kyoto, Japan.

Huang, X., Alleva, F., Hon, H.-W., Mei-Yuh Hwang, & Rosenfeld, R. (1992). *The SPHINX-II speech recognition system: an overview*. School of Computer Science, Carnegie Mellon University

Ishii, J., Tonomura, M., & Matsunaga, S. (1996). Speaker Adaptation Using Tree Structured Shared-State HMMs. *Fourth International Conference on Spoken Language* (pp. 1149–1152). Philadelphia, Pennsylvania, USA

Ishwaran, H., & Zarepour, M. (2002). Exact and approximate sum representations for the Dirichlet process. *Canadian Journal of Statistics*, 30(2), 269–283

Lee, C., & Glass, J. (2012). A Nonparametric Bayesian Approach to Acoustic Model Discovery. *Proceedings of the Association for Computational Linguistics* (pp. 40–49). Jeju, Republic of Korea.

Lee, K.-F. (n.d.). Context-independent phonetic hidden Markov models for speaker-independent continuous speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(4), 599– 609

Lefèvre, F. (n.d.). Non-parametric probability estimation for HMM-based automatic speech recognition. *Computer Speech & Language*, 17(2-3), 113–136.

Paliwal, K. (1990). Lexicon-building methods for an acoustic sub-word based speech recognizer. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing* (pp. 729–732). Albuquerque, New Mexico, USA

Qiao, Y., Shimomura, N., & Minematsu, N. (2008). Unsupervised optimal phoneme segmentation: Objectives, algorithms and comparisons. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing* (pp. 3989–3992). Las Vegas, Nevada, USA

Rabiner, L. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2), 879–893

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, 639–650.

Shang, L. (n.d.). Nonparametric Discriminant HMM and Application to Facial Expression Recognition. *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2090– 2096). Miami, FL, USA

Suchard, M. A., Wang, Q., Chan, C., Frelinger, J., West, M., & Cron, A. (2010). Understanding GPU Programming for Statistical Computation: Studies in Massively Parallel Massive Mixtures. *Journal of Computational and Graphical Statistics*, 19(2), 419–438.

Sudderth, E. (2006). *Graphical Models for Visual Object Recognition and Tracking*. Massachusetts Institute of Technology

Teh, Y.-W. (2010). Dirichlet process. *Encyclopedia of machine learning* (pp. 280–287).

Teh, Y., & Jordan, M. (2010). Hierarchical Bayesian Nonparametric Models with Applications. In S. W. Hjort, C. Holmes, P. Mueller (Ed.), *Bayesian Nonparametrics: Principles and*

Teh, Y., Jordan, M., Beal, M., & Blei, D. (2004). *Hierarchical Dirichlet Processes*.

Teh, Y., Jordan, M., Beal, M., & Blei, D. (2006). Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(47), 1566–1581

Williamson, S. A., Dubey, A., & Xing, E. P. (2012). Exact and efficient parallel inference for nonparametric mixture models. *arXiv preprint arXiv:1211.7120*

Wolfowitz, J. (1942). Additive partition functions and a class of statistical hypotheses. *The Annals of Mathematical Statistics*, 13(3), 247–279.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., et al. (2006). *The HTK Book* (p. 384). Cambridge, UK.