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

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 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

LIST OF FIGURES

LIST OF TABLES

Chapter 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 that research. However, during the past few years, despite of the availability of vast computational power and large amounts of data, the improvement of 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 proposal is 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 proposal 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), we can model each document as a probability distribution across topics (Teh & Jordan, 2010). 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 chapters.

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 a 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 cannot find the segments since the number of segments is not known. One potential solution is to use a model comparison technique (Meignier et al., 2001) 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 a 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.

In this proposal, 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 Maximum Likelihood Linear Regression (MLLR). Figure 1 compares the word

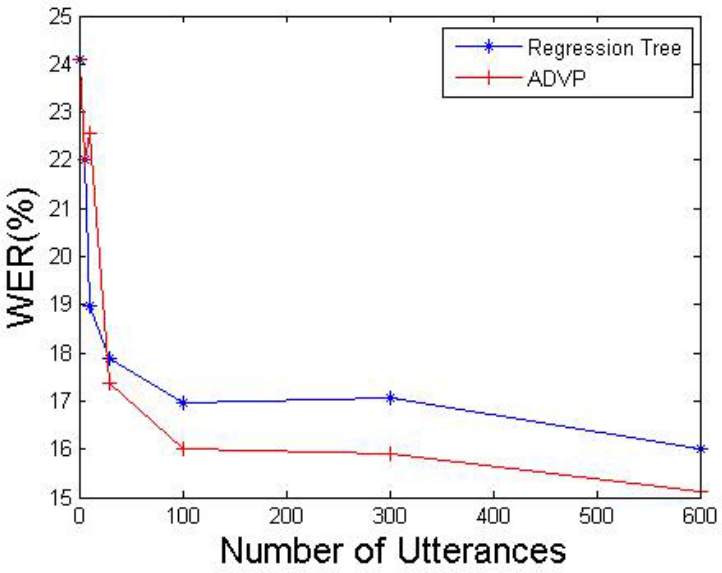

Figure 1 – A comparison of regression tree and DPM based clustering (Harati et al., 2012). Inference was implemented using an ADVP algorithm.

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 MLLR by 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 proposal, 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 proposal.

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

sub-word units such as context–dependent phones. 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 approached in a state of the art speech recognizer using left-to-right HMMs with a fixed number of states and a predetermined number of Gaussians per state (Rabiner, 1989). 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.

Chapter 2

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 proposal 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 proposal.

2.1 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

$\pi = (\pi_1, \pi_2, ..., \pi_K)$ where $0 \le \pi_k \le 1$ and $\sum_{k=1}^{K} \pi_k = 1$. This vector can characterize a multinomial distribution that is defined as:

$$p(x_1, x_2, ..., x_N \mid \pi) = \frac{N!}{\prod_k m_k!} \prod_k \pi_k^{m_k}, \quad m_k \triangleq \sum_n \delta(x_n, k) . \tag{1}$$

Equation (1) can be used to calculate the probability of selecting a category or class among K possible classes. In this definition m_k is the number of observations of category k. Given N observations, π can be estimated using a maximum likelihood (ML) approach (Sudderth, 2006). ML is a point estimate that 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):

$$\hat{\pi} = \arg\max_{\pi} \sum_{n=1}^{N} \log p(x_n \mid \pi) = \left(\frac{m_1}{N}, \frac{m_2}{N}, ..., \frac{m_K}{N} \right) . \tag{2}$$

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 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 contain 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 π using a Bayesian approach (Gelman, 2004). We should define a prior on π 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 $x_1,x_2,...,x_N$. Suppose the covariance matrix Σ is known. We can place a normal prior over the mean with mean μ_0 and covariance Σ_0. This prior is indicated with $Norm(\mu_0,\Sigma_0)$. After observing N data points the posterior over the mean is found using Bayes Rule and given by:

$$p\left(\mu \mid x_1,...,x_N,\Sigma,\mu_0,\Sigma_0\right) = Norm\left(\left(\Sigma_0^{-1}+\Sigma^{-1}\right)^{-1}+\left(\Sigma_0^{-1}\mu_0+\Sigma^{-1}\sum_{i=1}^{N}x_i\right),\left(\Sigma_0^{-1}+\Sigma^{-1}\right)^{-1}\right) . \quad (3)$$

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

$$P\left(\pi \mid \alpha\right) = Dir\left(\alpha_1,\alpha_2,...,\alpha_K\right) \triangleq \frac{\Gamma\left(\sum_k \alpha_k\right)}{\prod_k \Gamma\left(\alpha_k\right)} \prod_k \pi_k^{\alpha_k - 1}, \quad \alpha_k > 0 \quad . \quad (4)$$

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

$$\Gamma(z) = \int_0^{\infty} t^z e^{-t} \frac{dt}{t} . \quad (5)$$

A Gamma function is an extension of a factorial function to real and complex numbers (Milton et al., 1974). The concentration parameter, α, in is proportional to the inverse of the variance (Teh, 2010). Therefore, (5) 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:

$$E_{\alpha}\left[\pi_k\right] = \frac{\alpha_k}{\sum_j \alpha_j} \ . \tag{6}$$

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

$$\alpha_k = \frac{\sum_j \alpha_j}{K} \tag{7}$$

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

$$Var_{\alpha}\left[\pi_k\right] = \frac{K-1}{K^2\left(\sum_j \alpha_j + 1\right)} \ . \tag{8}$$

Equation (8) 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. For example, if a Dirichlet distribution is 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):

$$p\left(\pi \mid x_1, x_2, ..., x_N, \alpha\right) \propto p\left(\pi \mid \alpha\right) p\left(x_1, x_2, ..., x_N \mid \pi\right) \ . \tag{9}$$

By substituting from (1) and (4) we can write:

$$p\left(\pi \mid x_1, x_2, ..., x_N, \alpha\right) \propto \prod_{k=1}^{K} \pi_k^{\alpha_k + m_k - 1} \propto Dir\left(\alpha_1 + m_1, ..., \alpha_K + m_K\right) . \tag{10}$$

Equation (10) unlike (2) gives a distribution over π. The parameters of this distribution are learned from both the observed data and the prior assumptions.

From (10) we can see α_k 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, α_0, is equal to the sum of α_k:

$$\alpha_0 = \sum_k \alpha_k \ . \tag{11}$$

By considering this fact and (8) 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 (1) and (10) :

$$p\left(x_{new} \mid x_1,...,x_N,\alpha\right) = \frac{m_k + \alpha_k}{N + \sum_j \alpha_j} \ . \tag{12}$$

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 is a multinomial language model (e.g. bag of words). If we define the language model for a document (D) as π_D then for a sequence of independent terms we can write:

$$p\left(T_1,...,T_N \mid \pi_D\right) = \prod_{i=1}^{N} p\left(T_i \mid \pi_D\right) \ . \tag{13}$$

In this equation each $p\left(T_i \mid \pi_D\right)$ 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 (13). To compute this probability we have to compute π_D for all terms in the query. If we use the maximum likelihood solution in (2), we might get a 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 π_D

using a Dirichlet distribution as shown in (10) will solve this problem since it always gives a nonzero probability even if some of the terms are not presented in a document.

2.2 Dirichlet Process

A Dirichlet process (DP) is a distribution over distributions, or more precisely over discrete distributions. Formally, a Dirichlet process, $DP(\alpha, G_0)$, is "defined to be the distribution of a random probability measure G over Θ such that for any finite measurable partition $(A_1, A_2, \ldots, A_r)$ of Θ the random distribution $(G(A_1), \ldots, G(A_r))$ is distributed as finite dimensional Dirichlet distribution" (Teh et al., 2006):

$$\left(G\left(A_1\right), \ldots, G\left(A_r\right)\right) \sim Dir\left(\alpha G_0\left(A_1\right), \ldots, \alpha G_0\left(A_r\right)\right) . \tag{14}$$

In this definition α is the concentration parameter and is proportional to the inverse of the variance; G_0 is the base distribution and is the mean of the DP (e.g. $E\left(G\left(A\right)\right) = G_0\left(A\right)$).

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:

$$\begin{aligned} v_k \mid \alpha, G_0 &\sim Beta\left(1, \alpha\right), \qquad \theta_k \mid \alpha, G_0 \sim G_0 \\ \beta_k = v_k \prod_{l=1}^{k-1} \left(1 - v_l\right), &\qquad G = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k} . \end{aligned} \tag{15}$$

Starting with a stick of length one, we break it at v_1 and assign the length to β_1. Then we recursively break the remaining part of the stick and assign the corresponding lengths to β_k. In this representation β can be interpreted as a random probability measure over positive integers and is denoted by $\beta \sim GEM(\alpha)$.

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

$$\theta_i \mid \theta_1,...,\theta_{i-1},\alpha,G_0 \sim \sum_{k=1}^{i-1} \frac{1}{N-1+\alpha}\delta_{\theta_k} + \frac{\alpha}{N-1+\alpha}G_0 \ . \tag{16}$$

In the urn interpretation of (16), 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 $\theta_1^*,...,\theta_K^*$ be the distinct values and m_k be the number of θ_l associated with θ_k^*. We now have:

$$\theta_i \mid \theta_1,...,\theta_{i-1},\alpha,G_0 \sim \sum_{k=1}^{K} \frac{m_k}{i-1+\alpha}\delta_{\theta_k^*} + \frac{\alpha}{i-1+\alpha}G_0 \ . \tag{17}$$

Another useful interpretation of (17) is the Chinese restaurant process (CRP). In a CRP we have a Chinese restaurant with infinite number of tables. A new customer θ_i 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 (m_k) 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 z_i indicate the cluster associated with i^{th} observation. A CRP is the interpretation of the predictive distribution:

$$p\left(z_{N+1} = z \mid z_1,....,z_N,\alpha\right) = \frac{1}{\alpha+N}\left(\sum_{k=1}^{K} m_k\delta\left(z,k\right) + \alpha\delta\left(z,\bar{k}\right)\right) . \tag{18}$$

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. This metaphor illustrates a generative model for mixture modeling problem; where each component corresponds to a table and the food served at each table corresponds to the parameters of that component. CRF shows how data can be generated in a Dirichlet Process Mixture model.

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:

$$\begin{aligned} &\pi \mid \alpha \sim GEM(\alpha) \\ &z_i \mid \pi \sim Mult(\pi) \\ &\theta_k \mid G_0 \sim G_0 \\ &x_i \mid z_i, \{\theta_k\} \sim F\left(\theta_{z_i}\right) . \end{aligned} \tag{19}$$

In this model, observations x_i 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.

2.3 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 in tying groups together, i.e. to share clusters across all groups. Let's assume that we have an indexed collection of DPs with a common base distribution $\{G_j\} \sim DP(\alpha, G_0)$. Unfortunately this simple model cannot solve the problem since for continuous G_0 different G_j

have no atoms in common. The solution is to use a discrete G_0 with broad support. In other words, G_0 is itself a draw from a Dirichlet process.

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

$$\begin{aligned} &G_0 \mid \gamma, H \sim DP(\gamma, H) \\ &G_j \mid \alpha, G_0 \sim DP(\alpha, G_0) \\ &\theta_{ji} \mid G_j \sim G_j \\ &x_{ji} \mid \theta_{ji} \sim F\left(\theta_{ji}\right) \qquad for \qquad j \in J \ . \end{aligned} \tag{20}$$

In this definition H provides prior distribution for the factor θ_{ji}. The parameter γ governs the variability of G_0 around H and α controls the variability of G_j around G_0. H, γ and α are hyperparameters of the HDP. Equation (20) is just one representation of an HDP. Another representation can be obtained by introducing an indicator variable:

$$\begin{aligned} &\beta \mid \gamma \sim GEM(\gamma) \\ &\pi_j \mid \alpha, \beta \sim DP(\alpha, \beta) \\ &\theta_k \mid H, \lambda \sim H(\lambda) \\ &z_{ji} \mid \pi_j \sim \pi_j \\ &x_{ji} \mid \{\theta_k\}_{k=1}^{\infty}, z_{ji} \sim F\left(\theta_{ji}\right) \ . \end{aligned} \tag{21}$$

Figure 2 shows graphical models for both of these representations.

2.3.1 Stick-Breaking Construction

Because G_0 is a Dirichlet distribution it has a stick-breaking representation:

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k^{**}}, \tag{22}$$

where $\theta_k^{**} \sim H$ and $\beta = \left(\beta_k\right)_{k=1}^{\infty} \sim GEM\left(\gamma\right)$. Since support of G_j is contained within the support of G_0 we can write a similar equation to (22) for G_j:

$$G_j = \sum_{k=1}^{\infty} \pi_{jk} \delta_{\theta_k^{**}}. \tag{23}$$

Then we have:

$$\pi_j \sim DP(\alpha, \beta) \tag{24}$$

$$\begin{aligned} v_{jk} &\sim Beta\left(\alpha\beta_k, \alpha\left(1 - \sum_{l=1}^{k}\beta_l\right)\right) \\ \pi_{jk} &= v_{jk}\prod_{l=1}^{k-1}(1 - v_{jl}), \qquad for \quad k = 1,...,\infty \ . \end{aligned} \tag{25}$$

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. Customer i in restaurant j corresponds to θ_{ji} (customers are distributed according to G_j). Tables are i.i.d. variables θ^*_{jt} distributed according to G_0. Finally, foods are i.i.d. variables θ^{**}_k distributed according to H. If customer i at restaurant j sits at table t_{ji}

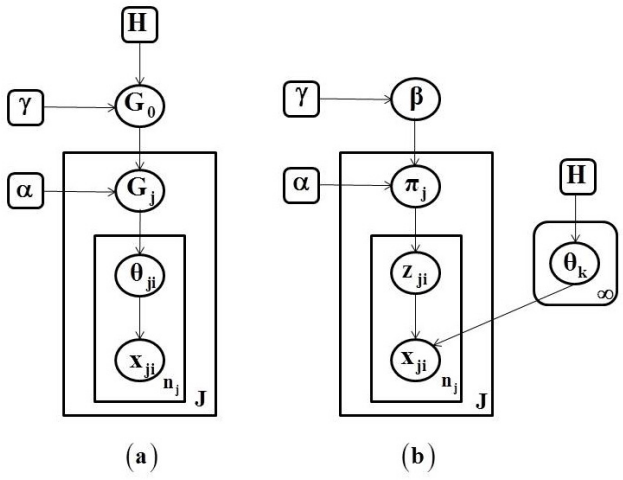

Figure 2 – In (a), an HDP representation of (5) is shown. In (b), an alternative indicator variable representation is shown (Teh et al., 2004).

and that table serves dish k_{ji}, we will have $\theta_{ji} = \theta^{*}_{jt_{ji}} = \theta^{**}_{k_{t_{ji}}}$. 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.

Next, we can introduce several variables that will be used throughout this paper: n_{jkt} is the number of customers in restaurant j, seated around table t, and who eat dish k; m_{jk} 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 dish labels $\boldsymbol{\theta}^{**} = \left\{\theta^{**}_{k}\right\}_{k=1,\ldots,K}$, tables $\left\{t_{ji}\right\}_{\substack{j=1,\ldots,J \\ i=1,\ldots,n_{\bullet\bullet}}}$ and dishes $\left\{k_{jt_{ji}}\right\}_{\substack{j=1,\ldots,J \\ i=1,\ldots,n_{\bullet\bullet}}}$. As a function of the state of the CRF, we also have the number of customers, $\mathbf{n} = \left\{n_{jtk}\right\}$, the number of tables, $\mathbf{m} = \left\{m_{jk}\right\}$, customer labels $\boldsymbol{\theta} = \left\{\theta_{ji}\right\}$ and table labels $\boldsymbol{\theta}^{*} = \left\{\theta^{*}_{jt}\right\}$ (Teh & Jordan, 2010). The posterior distribution of G_0 is given by:

$$G_0 \mid \gamma, H, \boldsymbol{\theta}^{*} \sim DP\left(\gamma + m_{\bullet\bullet}, \frac{\gamma H + \sum_{k=1}^{K} m_{\bullet k}\delta_{\theta^{**}_{k}}}{\gamma + m_{\bullet\bullet}}\right) \tag{26}$$

where $m_{\bullet\bullet}$ is the total number of tables in the franchise and $m_{\bullet k}$ is the total number of tables serving dish k. We can define the posterior for G_j:

$$G_j \mid \alpha, G_0, \boldsymbol{\theta}_{\mathbf{j}} \sim DP\left(\alpha + n_{j\bullet\bullet}, \frac{\alpha G_0 + \sum_{k=1}^{K} n_{j\bullet k}\delta_{\theta^{**}_{k}}}{\alpha + n_{j\bullet\bullet}}\right) \tag{27}$$

where $n_{j\bullet\bullet}$ is the total number of customers in restaurant j and $n_{j\bullet k}$ is the total number of customers in restaurant j eating dish k.

Conditional distributions can be obtained by integrating out G_j and G_0 respectively. By integrating out G_j from (27) we obtain:

$$\theta_{ji} \mid \theta_{j1},...,\theta_{j,i-1},\alpha,G_0 \sim \sum_{t=1}^{m_{j\cdot}} \frac{n_{jt\cdot}}{\alpha+n_{j\cdot\cdot}} \delta_{\theta^*_{jt}} + \frac{\alpha}{\alpha+n_{j\cdot\cdot}} G_0 \tag{28}$$

and by integrating out G_0 from (26) we obtain:

$$\theta^*_{jt} \mid \theta^*_{j1},...,\theta^*_{j,t-1},\gamma,H \sim \sum_{k=1}^{K} \frac{m_{\cdot k}}{\gamma+m_{\cdot\cdot}} \delta_{\theta^{**}_k} + \frac{\gamma}{\gamma+m_{\cdot\cdot}} H \ . \tag{29}$$

A draw from (26) can be obtained using:

$$\begin{aligned} &\beta_0,\beta_1,...,\beta_K \mid \gamma,G_0,\boldsymbol{\theta}^* \sim Dir\left(\gamma,m_{\cdot 1},...,m_{\cdot K}\right) \\ &G_0' \mid \gamma,H \sim DP\left(\gamma,H\right) \\ &G_0 = \beta_0 G_0' + \sum_{k=1}^{K} \beta_k \delta_{\theta^{**}_k} \end{aligned} \tag{30}$$

and a draw from (27) can be obtained using:

$$\begin{aligned} &\pi_{j0},\pi_{j1},...,\pi_{jK} \mid \alpha,\boldsymbol{\theta}_{\mathbf{j}} \sim Dir(\alpha\beta_0,\alpha\beta_1+n_{j\cdot 1},...,\alpha\beta_K+n_{j\cdot K}) \\ &G_j' \mid \alpha,G_0 \sim DP(\alpha\beta_0,G_0') \\ &G_j = \pi_{j0} G_j' + \sum_{k=1}^{K} \pi_{jk} \delta_{\theta^{**}_k}. \end{aligned} \tag{31}$$

From (30) and (31) we see that the posterior of G_0 is a mixture of atoms corresponding to dishes, and is an independent draw from *DP(γ,H)*. Similarly, G_j is a mixture of atoms at θ^{**}_k and an independent draw from $DP(\alpha\beta_0,G_0')$ (Teh & Jordan, 2010).

2.4 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 *t* with z_t and the state-specific transition distribution for state *j* by π_j. The Markovian structure is represented by $z_t \mid z_{t-1} \sim \pi_{z_{t-1}}$. Observations are conditionally independent given the state of the HMM and are denoted by $x_t \mid z_t \sim F\left(\theta_{z_t}\right)$.

An HDP-HMM is an extension of an HMM in which the number of states can be infinite. At each state z_t we should be able to transition 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 in inferring this association.

A major problem with original formulation of an HDP-HMM is state persistence. HDP-HMM has a tendency to make many redundant states and switch rapidly among them (Teh et al., 2006). This problem has been solved by introducing a sticky parameter to the definition of an HDP-HMM (Fox et al., 2011):

$$\begin{aligned}
&\beta \mid \gamma \sim GEM(\gamma) \\
&\pi_j \mid \alpha, \beta \sim DP(\alpha + \kappa, \frac{\alpha\beta + \kappa\delta_j}{\alpha + \kappa}) \\
&\theta_j^{**} \mid H, \lambda \sim H(\lambda) \\
&z_t \mid z_{t-1}, \{\pi_j\}_{j=1}^{\infty} \sim \pi_{z_{t-1}} \\
&x_t \mid \{\theta_j^{**}\}_{j=1}^{\infty}, z_t \sim F\left(\theta_{z_t}\right) .
\end{aligned} \tag{32}$$

Equation (32) shows the definition of a sticky HDP-HMM with unimodal emissions. The hyperparameter κ can be learned from data. The original formulation of an HDP-HMM is a special case with $\kappa = 0$. 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:

$$
\begin{aligned}
&\beta \mid \gamma \sim GEM(\gamma) \\
&\pi_j \mid \alpha, \beta \sim DP(\alpha+\kappa, \frac{\alpha\beta+\kappa\delta_j}{\alpha+\kappa}) \\
&\psi_j \mid \sigma \sim GEM(\sigma) \\
&\theta_{kj}^{**} \mid H, \lambda \sim H(\lambda) \\
&z_t \mid z_{t-1}, \left\{\pi_j\right\}_{j=1}^{\infty} \sim \pi_{z_{t-1}} \\
&s_t \mid \left\{\psi_j\right\}_{j=1}^{\infty}, z_t \sim \psi_{z_t} \\
&x_t \mid \left\{\theta_{kj}^{**}\right\}_{k,j=1}^{\infty}, z_t \sim F\left(\theta_{z_t s_t}\right) .
\end{aligned}
\tag{33}
$$

In this model, a DP is associated with each state and a model with augmented state (z_t,s_t) is obtained. Figure 3 shows a graphical representation.

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 x_t is going to restaurant j then it is more likely that he eats the specialty dish z_t=j. His children x_{t+1} also go to the same restaurant and eat the same dish. However, if x_t eats another dish ($z_t \neq j$) then his children go to the restaurant indexed by z_t and 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.

2.5 Inference Algorithms For HDP-HMM

2.5.1 Direct Sampler

In this section we present a sampler for an HDP-HMM with DP emission (Fox et al., 2011). The algorithm is divided into two steps: (1) sample the augmented state (z_t,s_t), and (2) sample β. In order to sample (z_t,s_t) we need to have a posterior. By inspecting Figure 3 and using the chain rule we can write the following relationship for this posterior:

$$
\begin{aligned}
&p\left(z_t = k, s_t = j \mid z_{\setminus t}, s_{\setminus t}, x_{1:T}, \beta, \alpha, \sigma, \kappa, \lambda\right) = \\
&p\left(s_t = j \mid z_t = k, z_{\setminus t}, s_{\setminus t}, x_{1:T}, \sigma, \lambda\right) p\left(z_t = k \mid z_{\setminus t}, s_{\setminus t}, x_{1:T}, \beta, \alpha, \kappa, \lambda\right) \propto \\
&p\left(s_t = j \mid \left\{s_\tau \mid z_\tau = k, \tau \neq t\right\}, \sigma\right) p\left(x_t \mid \left\{x_\tau \mid z_\tau = k, s_t = j, \tau \neq t\right\}\right) \\
&p\left(z_t = k \mid z_{\setminus t}, \beta, \alpha, \kappa\right) \sum_{s_t} \left(p\left(x_t \mid \left\{x_\tau \mid z_\tau = k, s_t, \tau \neq t\right\}\right) p\left(s_t \mid \left\{s_\tau \mid z_\tau = k, \tau \neq t\right\}, \sigma\right)\right).
\end{aligned} \tag{34}
$$

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):

$$
p\left(z_t = k \mid z_{\setminus t}, \beta, \alpha, \kappa\right) \propto \begin{cases} \left(\alpha\beta_k + n_{z_{t-1}k}^{-t} + \kappa\delta\left(z_{t-1}, k\right)\right)\left(\dfrac{\alpha\beta_{z_{t+1}} + n_{kz_{t+1}}^{-t} + \kappa\delta\left(k, z_{t+1}\right) + \delta\left(z_{t-1}, k\right)\delta\left(k, z_{t+1}\right)}{\alpha + n_{k\bullet}^{-t} + \kappa + \delta\left(z_{t-1}, k\right)}\right) & k \in \{1, \ldots, K\} \\ \dfrac{\alpha^2 \beta_{\bar{k}} \beta_{z_t+1}}{\alpha + \kappa}, & k = K+1 \end{cases} \tag{35}
$$

$$
p\left(s_t = j \mid \left\{s_\tau \mid z_\tau = k, \tau \neq t\right\}, \sigma\right) = \begin{cases} \dfrac{n_{kj}'^{-t}}{\sigma + n_{k\bullet}'^{-t}}, & j \in \left\{1, \ldots, K_k'\right\} \\ \dfrac{\sigma}{\sigma + n_{k\bullet}'^{-t}}, & j = K_k' + 1 \end{cases}. \tag{36}
$$

$$
\begin{aligned}
&p\left(x_t \mid \left\{x_\tau \mid z_\tau = k, s_t = j, \tau \neq t\right\}\right) = t_{\bar{\nu}_{kj} - d - 1}\left(x_t; \bar{\vartheta}_{kj}, \frac{\left(\bar{\zeta}_{kj} + 1\right)\bar{\nu}_{kj}}{\bar{\zeta}_{kj}\left(\bar{\nu}_{kj} - d - 1\right)}\bar{\Delta}_{kj}\right), k = 1, \ldots, K, j = 1, \ldots, K_k' \\
&p\left(x_t \mid \left\{x_\tau \mid z_\tau = k, s_t = j^{new}, \tau \neq t\right\}\right) = t_{\bar{\nu}_{kj^{new}} - d - 1}\left(x_t; \vartheta, \frac{\left(\zeta + 1\right)\nu}{\zeta\left(\nu - d - 1\right)}\Delta\right), k = 1, \ldots, K, j = j^{new} \\
&p\left(x_t \mid \left\{x_\tau \mid z_\tau = k^{new}, \tau \neq t\right\}\right) = t_{\nu_{k^{new}} - d - 1}\left(x_t; \vartheta, \frac{\left(\zeta + 1\right)\nu}{\zeta\left(\nu - d - 1\right)}\Delta\right), \qquad k = k^{new}.
\end{aligned} \tag{37}
$$

The algorithm is as follows:

1. Given a previous set of $\left(z_{1:T}^{(n-1)}, s_{1:T}^{(n-1)}\right)$ and $\beta^{(n-1)}$,
2. For all $t \in \{1, 2, \ldots, T\}$,
3. For each of the K currently instantiated states compute:

a. The predictive conditional distributions for each of the K_k' currently instantiated mixture components for this state, and also for a new component and for a new state.

$$f'_{k,j}(x_t) = \left(\frac{n'^{-t}_{kj}}{\sigma + n'^{-t}_{k\bullet}} \right) p\left(x_t \mid \{ x_\tau \mid z_\tau = k, s_t = j, \tau \neq t \} \right). \tag{38}$$

$$f_{k,K_k'+1}(x_t) = \frac{\sigma}{\sigma + n'^{-t}_{k\bullet}} p\left(x_t \mid \left\{ x_\tau \mid z_\tau = k, s_t = j^{new}, \tau \neq t \right\} \right). \tag{39}$$

$$f'_{k^{new},0}(x_t) = \left(\frac{\sigma}{\sigma + n'^{-t}_{\bullet\bullet}} \right) p\left(x_t \mid \left\{ x_\tau \mid z_\tau = k^{new}, \tau \neq t \right\} \right). \tag{40}$$

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

$$\begin{aligned} f_k(x_t) = & \left(\alpha\beta_k + n^{-t}_{z_{t-1}} + \kappa\delta(z_{t-1}, k) \right) \\ & \left(\frac{\alpha\beta_{z_{t+1}} + n^{-t}_{kz_{t+1}} + \kappa\delta(k, z_{t+1}) + \delta(z_{t-1}, k)\delta(k, z_{t+1})}{\alpha + n^{-t}_{k\bullet} + \kappa + \delta(z_{t-1}, k)} \right) \\ & \left(\sum_{j=1}^{K_k'} f'_{k,j}(x_t) + f_{k,K_k'+1}(x_t) \right), k = 1, \ldots, K. \end{aligned} \tag{41}$$

$$f_{K+1}(x_t) = \frac{\alpha^2 \beta_{\bar{k}} \beta_{z_t+1}}{\alpha + \kappa} f'_{k^{new},0}(x_t), k = K+1. \tag{42}$$

4. Sample z_t:

$$z_t \sim \sum_{k=1}^{K} f_k(x_t)\delta(z_t, k) + f_{K+1}(x_t)\delta(z_t, K+1). \tag{43}$$

5. Sample s_t conditioned on z_t:

$$s_t \sim \sum_{j=1}^{K_k'} f'_{k,j}(x_t)\delta(s_t, j) + f_{k,K_k'+1}(x_t)\delta(s_t, K_k'+1). \tag{44}$$

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

$$v_0 \mid \gamma \sim Beta(\gamma,1)$$
$$\left(\beta_0^{new},\beta_{K+1}^{new}\right)=\left(\beta_0 v_0,\beta_0(1-v_0)\right) . \tag{45}$$

7. If $s_t = K_k' + 1$ increase K_k'.
8. Update the cache. If there is a state with $n_{k\bullet} = 0$ or $n_{\bullet k} = 0$, remove k and decrease K. If $n_{kj}' = 0$ remove the component j and decrease K_k'.
9. Sample auxiliary variables by simulating a CRF:
10. For each $(j,k) \in \{1,...,K\}^2$ set m_{jk}=0 and n=0. For each customer in restaurant j eating dish k ($i = 1,...,n_{jk}$), sample:

$$x \sim Ber\left(\frac{\alpha\beta_k + \kappa\delta(j,k)}{n + \alpha\beta_k + \kappa\delta(j,k)}\right) . \tag{46}$$

11. Increase n and if $x=1$ increase m_{jk}.
12. For each $j \in \{1,...,K\}$, sample the override variables in restaurant j:

$$\omega_{j\bullet} \sim Binomial\left(m_{jj}, \frac{\rho}{\rho + \beta_j(1-\rho)}\right), \rho = \frac{\kappa}{\alpha+\kappa} . \tag{47}$$

13. Set the number of informative tables in restaurant j:

$$\bar{m}_{jk} = \begin{cases} m_{jk} & j \neq k \\ m_{jj} - \omega_{j\bullet} & j = k \end{cases} . \tag{48}$$

14. Sample β:

$$\beta^{(n)} \sim Dir\left(\gamma, \bar{m}_{\bullet 1},...,\bar{m}_{\bullet K}\right) . \tag{49}$$

15. Optionally sample hyperparameters σ, γ, α and κ.

2.5.2 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 sequence $z_{1:T}$ at 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. 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:

$$GEM_L(\alpha) \triangleq Dir(\alpha / L, ..., \alpha / L) \ . \tag{50}$$

Using (50) β is approximated as (Fox et al., 2010):

$$\beta \mid \gamma \sim Dir(\gamma / L, ..., \gamma / L) \ . \tag{51}$$

We can write:

$$\pi_j \mid \alpha, \kappa, \beta \sim Dir(\alpha\beta_1, ..., \alpha\beta_j + \kappa, ...\alpha\beta_L) \ . \tag{52}$$

The posteriors are given by:

$$\begin{aligned} &\beta \mid \bar{\mathbf{m}}, \gamma \sim Dir(\gamma / L + \bar{m}_{\cdot 1}, ..., \gamma / L + \bar{m}_{\cdot L}) \\ &\pi_j \mid z_{1:T}, \alpha, \beta \sim Dir(\alpha\beta_1 + n_{j1}, ..., \alpha\beta_j + \kappa + n_{jj}, ..., \alpha\beta_L + n_{jL}) \ . \end{aligned} \tag{53}$$

In (53) n_{jk} is the number of transitions from state j to state k and $\bar{m}_{jk}$ is the same as (48). Finally an order L' weak limit approximation is used for the DP prior on the emission parameters:

$$\psi_k \mid z_{1:T}, s_{1:T}, \sigma \sim Dir(\sigma / L' + n'_{k1}, ..., \sigma / L' + n'_{kL'}) \ . \tag{54}$$

The forward-backward algorithm for the joint sample $z_{1:T}$ and $s_{1:T}$ given $x_{1:T}$ can be obtained by:

$$\begin{aligned} & p\left(z_t, s_t \mid x_{1:T}, \mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\psi}, \boldsymbol{\theta}\right) \\ & \propto p\left(z_t \mid z_{t-1}, x_{1:T}, \boldsymbol{\pi}, \boldsymbol{\theta}\right) p\left(s_t \mid \psi_{z_t}\right) p\left(x_t \mid \theta_{z_t, s_t}\right) p\left(x_{1:t-1} \mid z_t, \boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\psi}\right) p\left(x_{t+1:T} \mid z_t, \boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\psi}\right) . \end{aligned} \tag{55}$$

The right side of (55) has two parts: forward and backward probabilities (Rabiner,1989). The forward probability includes $p\left(z_t \mid z_{t-1}, x_{1:T}, \boldsymbol{\pi}, \boldsymbol{\theta}\right) p\left(s_t \mid \psi_{z_t}\right) f\left(x_t \mid \theta_{z_t, s_t}\right) p\left(x_{1:t-1} \mid z_t, \boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\psi}\right)$ and the backward probability includes $p\left(x_{t+1:T} \mid z_t, \boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\psi}\right)$. The forward probabilities are approximated with $p\left(z_t \mid z_{t-1}, x_{1:T}, \boldsymbol{\pi}, \boldsymbol{\theta}\right) p\left(s_t \mid \psi_{z_t}\right) f\left(x_t \mid \theta_{z_t, s_t}\right)$. Therefore, for the backward probabilities we have:

$$\begin{aligned} & p\left(x_{t+1:T} \mid z_t, \boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\psi}\right) \propto m_{t, t-1}\left(z_{t-1}\right) \\ & \propto \begin{cases} \sum_{z_t} \sum_{s_t} p\left(z_t \mid \pi_{z_t-1}\right) p\left(s_t \mid \psi_{z_t}\right) f\left(x_t \mid \theta_{z_t, s_t}\right) m_{t+1, t}\left(z_t\right) & t \leq T \\ 1 & t=T+1 \end{cases} \\ & \Rightarrow m_{t, t-1}(k) \propto \begin{cases} \sum_{i=1}^{L} \sum_{l=1}^{L^{\prime}} \pi_{ki} \psi_{il} f\left(x_t \mid \theta_{z_t, s_t}\right) m_{t+1, t}\left(z_t\right) & t \leq T \\ & k=1, \ldots L \\ 1 & t=T+1 . \end{cases} \end{aligned} \tag{56}$$

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

$$p\left(z_{\ t}=k, s_t=j \mid x_{1:T}, \mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\psi}, \boldsymbol{\theta}\right) \propto \pi_{z_{t-1} k} \psi_{kj} f\left(x_t \mid \theta_{z_t, s_t}\right) m_{t+1, t}\left(z_t\right) \tag{57}$$

For Gaussian emission distributions, the components are given by $f\left(x_t \mid \theta_{z_t, s_t}\right)=\mathrm{N}\left(x_t ; \mu_{kj}, \Sigma_{kj}\right)$.

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

1. Given the previous $\boldsymbol{\pi}^{(n-1)}, \boldsymbol{\psi}^{(n-1)}, \boldsymbol{\beta}^{(n-1)}$ and $\boldsymbol{\theta}^{(n-1)}$.
2. For $k \in\{1, \ldots, L\}$, initialize $m_{T+1, T}(k)=1$,
3. For $t \in\{T-1, \ldots, 1\}$ and $k \in\{1, \ldots, L\}$ compute

$$m_{t,t-1}(k) = \sum_{i=1}^{L}\sum_{l=1}^{L} \pi_{ki}\psi_{il} N(x_{t+1};\mu_{il},\Sigma_{il}) m_{t+1,t}(i) \ . \tag{58}$$

4. Sample the augmented state (z_t,s_t) sequentially and start from $t=1$:

5. Set $n_{ik}=0, n'_{kj}=0$ and $\Upsilon_{kj}=\varnothing$ for $(i,k)\in\{1,...,L\}^2$ and $(k,j)\in\{1,...,L\}\times\{1,...,L'\}$

6. For all $(k,j)\in\{1,...,L\}\times\{1,...,L'\}$ compute:

$$f_{k,j}(x_t) = \pi_{z_{t-1},k}\psi_{k,j} N\left(x_t;\mu_{k,j},\Sigma_{k,j}\right) m_{t+1,t}(k) \ . \tag{59}$$

7. Sample the augmented state (z_t,s_t):

$$(z_t, s_t) \sim \sum_{k=1}^{L}\sum_{j=1}^{L'} f_{k,j}(x_t)\delta(z_t,k)\delta(s_t,j) \ . \tag{60}$$

8. Increase $n_{z_{t-1}z_t}$ and $n'_{z_t s_t}$ and add x_t to the cached statistics.

$$\Upsilon_{k,j} \leftarrow \Upsilon_{k,j} \oplus x_t \ . \tag{61}$$

9. Sample $\mathbf{m}$, $\boldsymbol{\omega}$, $\bar{\mathbf{m}}$ similar to the previous algorithm

10. Update β:

$$\beta \sim Dir(\gamma/L + \bar{m}_{\bullet 1},...,\gamma/L + \bar{m}_{\bullet L}) \ . \tag{62}$$

11. For $k\in\{1,...,L\}$:

    a. Sample π_k and ψ_k:

$$\begin{aligned} \pi_k &\sim Dir(\alpha\beta_1 + n_{k1},...,\alpha\beta_k + \kappa + n_{kk},...,\alpha\beta_L + n_{kL}) \\ \psi_k &\sim Dir(\sigma/L' + n'_{k1},...,\sigma/L' + n'_{kL'}) \ . \end{aligned} \tag{63}$$

    b. For $j\in\{1,...,L'\}$ sample:

$$\theta_{k,j} \sim p(\theta \mid \lambda, \Upsilon_{k,j}) \ . \tag{64}$$

12. Set $\boldsymbol{\pi}^{(n)}=\boldsymbol{\pi}, \boldsymbol{\psi}^{(n)}=\boldsymbol{\psi}, \beta^{(n)}=\beta$ and $\boldsymbol{\theta}^{(n)}=\boldsymbol{\theta}$.

13. Optionally sample hyperparameters σ, γ, α and κ.

2.5.3 Learning Hyperparameters

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

Posterior for ($\alpha + \kappa$)

Consider the probability of data x_{ji} to sit behind table t:

$$p\left(t_{ji}=t \mid \mathbf{t}^{-ji}, n_{jt\cdot}^{-ji}, \alpha, \kappa\right) \propto \begin{cases} n_{jt\cdot}^{-ji} & t \in \left\{t_1,...,m_{j\cdot}\right\} \\ \alpha+\kappa & t=t^{new} \end{cases}. \tag{65}$$

This equation can be written by considering (28) and (32). From this equation we can say customer table assignment follows a DP with concentration parameter $\alpha + \kappa$. Antoniak (1974) has shown that if $\beta \sim GEM(\gamma), z_i \sim \beta$ then the distribution of the number of unique values of z_i resulting from N draws from β has the following form:

$$p(K \mid N, \gamma) = \frac{\Gamma(\gamma)}{\Gamma(\gamma+N)} s(N,K) \gamma^K \tag{66}$$

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 restaurant j is as follows:

$$p\left(m_{j\cdot} \mid \alpha+\kappa, n_{j\cdot}\right) = s\left(n_{j\cdot\cdot}, m_{j\cdot}\right)(\alpha+\kappa)^{m_{j\cdot}} \frac{\Gamma(\alpha+\kappa)}{\Gamma\left(\alpha+\kappa+n_{j\cdot}\right)}. \tag{67}$$

The posterior over $\alpha + \kappa$ is as follows:

$$\begin{aligned} p\left(\alpha+\kappa \mid m_{1\cdot},...,m_{J\cdot}, n_{1\cdot},...,n_{J\cdot}\right) &\propto p(\alpha+\kappa)\, p\left(m_{1\cdot},...,m_{J\cdot} \mid \alpha+\kappa, n_{1\cdot},...,n_{J\cdot}\right) \\ &\propto p(\alpha+\kappa) \prod_{j=1}^{J} p\left(m_{j\cdot} \mid \alpha+\kappa, n_{j\cdot}\right) \\ &\propto p(\alpha+\kappa) \prod_{j=1}^{J} s\left(n_{j\cdot}, m_{j\cdot}\right)(\alpha+\kappa)^{m_{j\cdot}} \frac{\Gamma(\alpha+\kappa)}{\Gamma\left(\alpha+\kappa+n_{j\cdot}\right)} \\ &\propto p(\alpha+\kappa)(\alpha+\kappa)^{m_{\cdot\cdot}} \prod_{j=1}^{J} \frac{\Gamma(\alpha+\kappa)}{\Gamma\left(\alpha+\kappa+n_{j\cdot}\right)}. \end{aligned} \tag{68}$$

The reason for the last line is that $\prod_{j=1}^{J} s\left(n_{j\cdot}, m_{j\cdot}\right)$ is not a function of $\alpha + \kappa$ and therefore can be ignored. By substitution of $\beta(x,y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)} = \int_0^1 t^{x-1}(1-t)^{y-1}\,dt$ and also by considering that $\Gamma(x+1) = x\Gamma(x)$ we obtain:

$$p\left(\alpha+\kappa \mid m_{1\cdot},...,m_{J\cdot},n_{1\cdot},...,n_{J\cdot}\right) \propto p\left(\alpha+\kappa\right)\left(\alpha+\kappa\right)^{m_{\cdot\cdot}} \prod_{j=1}^{J}\left(1+\frac{n_{j\cdot}}{\alpha+\kappa}\right)\int_0^1 r_j^{\alpha+\kappa}\left(1-r_j\right)^{n_{j\cdot}-1} dr_j \quad (69)$$

Finally by considering the fact that we have placed a Gamma(a,b) prior on $\alpha + \kappa$ we can write:

$$p\left(\alpha+\kappa, r, s \mid m_{1\cdot},...,m_{J\cdot},n_{1\cdot},...,n_{J\cdot}\right) \propto \left(\alpha+\kappa\right)^{a+m_{\cdot\cdot}-1} e^{-(\alpha+\kappa)b} \prod_{j=1}^{J}\left(\frac{n_{j\cdot}}{\alpha+\kappa}\right)^{s_j} r_j^{\alpha+\kappa}\left(1-r_j\right)^{n_{j\cdot}-1} \quad (70)$$

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

$$\begin{aligned} p\left(\alpha+\kappa \mid r, s, m_{1\cdot},...,m_{J\cdot},n_{1\cdot},...,n_{J\cdot}\right) &\propto \left(\alpha+\kappa\right)^{a+m_{\cdot\cdot}-1-\sum_{j=1}^{J} s_j} e^{-(\alpha+\kappa)\left(b-\sum_{j=1}^{J}\log r_j\right)} \\ &= Gamma\left(\alpha+m_{\cdot\cdot}-\sum_{j=1}^{J} s_j, b-\sum_{j=1}^{J}\log r_j\right). \end{aligned} \quad (71)$$

$$p\left(r_j \mid \alpha+\kappa, r_{\setminus j}, s, m_{1\cdot},...,m_{J\cdot},n_{1\cdot},...,n_{J\cdot}\right) \propto r_j^{\alpha+\kappa}\left(1-r_j\right)^{n_{j\cdot}-1} = Beta\left(\alpha+\kappa+1, n_{j\cdot}\right). \quad (72)$$

$$p\left(s_j \mid \alpha+\kappa, r, s_{\setminus j} m_{1\cdot},...,m_{J\cdot},n_{1\cdot},...,n_{J\cdot}\right)\alpha\left(\frac{n_{j\cdot}}{\alpha+\kappa}\right)^{s_j} = Ber\left(\frac{n_{j\cdot}}{n_{j\cdot}+\alpha+\kappa}\right). \quad (73)$$

Posterior of γ

Similar to (66) if we want to find the distribution of the unique number of dishes served in the whole franchise, we would have $p\left(K \mid \gamma, \bar{m}_{\cdot\cdot}\right) = s\left(\bar{m}_{\cdot\cdot}, K\right)\frac{\Gamma(\gamma)}{\Gamma\left(\gamma+\bar{m}_{\cdot\cdot}\right)}$. Therefore for the posterior distribution of γ we can write:

$$
\begin{aligned}
p(\gamma \mid K, \bar{m}_{\cdot\cdot}) &\propto p(\gamma)\, p(K \mid \gamma, \bar{m}_{\cdot\cdot}) \\
&\propto p(\gamma)\, \gamma^{K}\, \frac{\beta(\gamma+1, \bar{m}_{\cdot\cdot})}{\gamma\Gamma(\bar{m}_{\cdot\cdot})} \\
&\propto p(\gamma)\, \gamma^{K}\, (\gamma+\bar{m}_{\cdot\cdot}) \int_{0}^{1} \eta^{\gamma} (1-\eta)^{\bar{m}_{\cdot\cdot}-1}\, d\eta \,.
\end{aligned}
\tag{74}
$$

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

$$
p(\gamma, \eta, \varsigma \mid K, \bar{m}_{\cdot\cdot}) \propto \gamma^{\alpha+K-1} \left(\frac{\bar{m}_{\cdot\cdot}}{\gamma} \right)^{\varsigma} e^{-\gamma(b-\log\eta)} (1-\eta)^{\bar{m}_{\cdot\cdot}-1} \,. \tag{75}
$$

Finally for the marginal distributions we have:

$$
p(\gamma \mid \eta, \varsigma, K, \bar{m}_{\cdot\cdot}) \propto \gamma^{\alpha+K-1-\varsigma} e^{-\gamma(b-\log\eta)} = Gamma(\alpha+K-\varsigma, b-\log\eta) \tag{76}
$$

$$
p(\eta \mid \gamma, \varsigma, K, \bar{m}_{\cdot\cdot}) \propto \eta^{\gamma} (1-\eta)^{\bar{m}_{\cdot\cdot}-1} = Beta(\gamma+1, \bar{m}_{\cdot\cdot}) \tag{77}
$$

$$
p(\varsigma \mid \gamma, \eta, K, \bar{m}_{\cdot\cdot}) \propto \left(\frac{\bar{m}_{\cdot\cdot}}{\gamma} \right)^{\varsigma} = Ber\left(\frac{\bar{m}_{\cdot\cdot}}{\bar{m}_{\cdot\cdot}+\gamma} \right) . \tag{78}
$$

Posterior of σ

The posterior for σ is obtained in a similar way to α+κ. We use two auxiliary variables r' and s'. The final marginalized distributions are:

$$
p(\sigma \mid r', s', K'_{1\cdot}, \ldots, K'_{J\cdot}, n_{1\cdot}, \ldots, n_{J\cdot}) \propto (\sigma)^{\alpha+K'_{\cdot\cdot}-1-\sum_{j=1}^{J} s'_j}\, e^{-\sigma\left(b-\sum_{j=1}^{J}\log r'_j\right)} . \tag{79}
$$

$$
p\left(r'_j \mid \sigma, r'_{\setminus j}, s', K'_{1\cdot}, \ldots, K'_{J\cdot}, n_{1\cdot}, \ldots, n_{J\cdot}\right) \propto r_j'^{\sigma} \left(1-r_j'^{\sigma}\right)^{n_{j\cdot}-1} . \tag{80}
$$

$$
p\left(s'_j \mid \sigma, r', s'_{\setminus j}, K'_{1\cdot}, \ldots, K'_{J\cdot}, n_{1\cdot}, \ldots, n_{J\cdot}\right) \propto \left(\frac{n_{j\cdot}}{\sigma} \right)^{s'_j} . \tag{81}
$$

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 $\rho = \frac{\kappa}{\alpha + \kappa}$. By considering the fact that the prior on ρ is *Beta(c,d)* and $\omega_{jt} \sim Ber(\rho)$ we can write:

$$\begin{aligned} p(\rho \mid \omega) &\propto p(\omega \mid \rho)\, p(\rho) \\ &\propto Binomial\left(\sum_j \omega_{j\cdot}; m_{\cdot\cdot}, \rho \right) Beta(c,d) \\ &\propto Beta\left(\sum_j \omega_{j\cdot} + c, m_{\cdot\cdot} - \sum_j \omega_{j\cdot} + d \right). \end{aligned} \tag{82}$$

In this chapter, we have reviewed nonparametric Bayesian methods used in the other parts of this dissertation. We started from Dirichlet Process as a fundamental building block and used it to construct the Hierarchical Dirichlet Process and eventually HDP-HMMs. Two inference algorithms for HDP-HMM have been reviewed.

Chapter 3

ACOUSTIC MODELING

The ultimate goal of speech recognition is to map the acoustic data into word sequences. This problem can be formulated as (Gelman, 2004):

$$P(W \mid A) = \frac{P(A \mid W)P(W)}{P(A)}. \tag{83}$$

In this formulation, *P(W|A)* is the probability of a particular word sequence given acoustic 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 acoustic modeling problem.

3.1 Acoustic Modeling in State of the Art Systems

In this section, we review the most popular approach to acoustic modeling in speech recognition 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 word 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. Different acoustic units such as phones (Lee, 1990), syllables (Ganapathiraju et al., 2001) and acoustically inspired units

(Paliwal, 1990) have been used over the years. Phones are the most popular and easiest to use units. Most successful commercial systems are based on them.

After selecting the type of the units, a lexicon is needed that maps words into these units. We also have to select a statistical model to be used as a model for each acoustic unit. 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 systems are left-to-right HMMs with mixtures of Gaussians used to model emission probabilities (Rabiner, 1989). An HMM is a generalization of a 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. A predetermined number of mixture components per state, often ranging from 16 to 128 depending on the application, are employed.

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 observations associated with them and many more will have just a few examples. Therefore estimated parameters will have large variances, and sophisticated parameter sharing techniques must be employed (Young et al., 2006).

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 addressed by tying models and states together so similar models share data. This is a tradeoff between model accuracy and the amount of available training 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 iterations using EM algorithm.

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

Chapter 4

SPEECH SEGMENTATION AND ACOUSTIC UNIT LEARNING

4.1 Problem statement

Speech segmentation, defined as the process of finding boundaries between various acoustic units such as words or phones, is perhaps the most fundamental process in speech recognition. Accurate segmentation often results in correct recognition. Conversely, recognition errors can usually be traced to segmentation errors. HMM-based speech recognition systems do a very good job of segmenting the speech signal. Segmentation is implicitly used in all speech recognition tasks. However, explicit, or standalone applications of speech segmentation algorithms are usually limited to problems such as word spotting (Gish and Ng, 1993) and speech/non-speech classification (Shin et al., 2000).

Another interesting application of speech segmentation is 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 discuss related work and our proposed approach.

4.2 Relevant Work

Classical methods for acoustic unit discovery involve segmentation and clustering. The segmentation is typically implemented using a dynamic programming method that incorporates a heuristic stopping criterion (Bacchiani & Ostendorf, 1999), while clustering is implemented using a heuristic agglomerative method (Bacchiani & Ostendorf, 1999).

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 partition an input audio stream into homogeneous segments according to the speaker identity. Fox et al. (2011) have used an 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.

4.3 Proposed Approach

Our approach for speech segmentation is also based on an HDP-HMM model. We propose to segment the speech using an ergodic HMM. In this model, each state models an acoustic unit. Figure 4 demonstrates an example on some preliminary experiments based on this model (Harati et al., 2013). From this figure we can see the discovered boundaries approximately

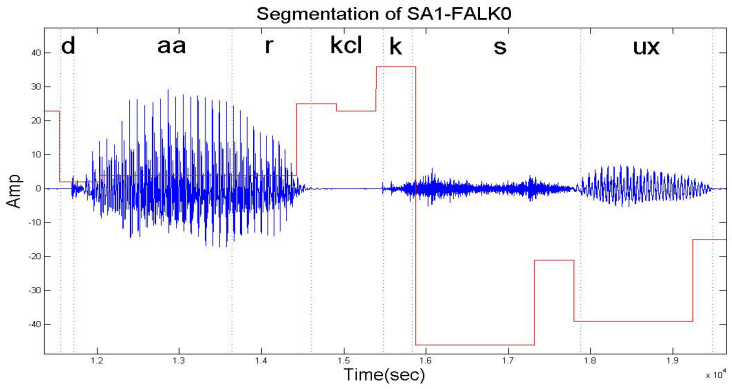

Figure 4 – Segmentation of a speech utterance produced through a process of automatic unit discovery is shown by overlaying the duration and index of each unit on the waveform. The height of each rectangle overlay simply indicates the index of that unit.

coincide with phoneme boundaries. Table 1 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:

$$\text{F-score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{84}$$

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

Table 1 – The segmentation performance of HDP-HMM is compared to several 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

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 generate a lexicon. We are planning to use forced alignment or manually transcribed data to map words into acoustic units. The performance of the system will be measured into two ways:

1. 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. Word Error Rate (WER): This will assess the impact on performance for a system trained completely using our proposed units.

The latter method of measuring performance is more interesting from a practical point of view. However, the performance will be a dependent on our ability to automatically generate a high-quality lexicon.

Chapter 5

LEFT-TO-RIGHT HDP-HMM MODELS

5.1 Problem Statement

The most important element of acoustic modeling is the statistical approach used to model sub-word units. Most state of the art systems use left-to-right HMMs with Gaussian mixtures to model phonetic units (Rabiner, 1989). Usually, the number of states is fixed for all models (e.g., *3*). Mixtures are trained progressively by starting from one mixture per state and increasing the number of mixtures until a further increment does not improve the likelihood of the training data. The 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 mixture components 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 an optimum choice.

5.2 Related Work

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 a neural network based on a multilayer perceptron (MLP). It was shown that MLPs generate reasonable estimates of a posterior distribution of an output class conditioned on the input patterns (Bourlard & Morgan, 1993). This hybrid HMM-MLP system works slightly better than traditional HMM-GMMs, but the gain was not significant enough to justify adopting this new technology. Most of the gain can be recovered by using more sophisticated processing steps such as speaker adaptation.

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 were 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 a Gaussian process dynamical model (GPDM) to completely replace HMMs in acoustic modeling. The new model is nonparametric Bayesian and is based on a 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 no results have been reported for speech recognition.

5.3 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 applications we usually need a more constrained topology. The left-to-right topology has proven to be useful in speech recognition and similar applications (Rabiner, 1989). We propose 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.

Figure 5 shows the discovered structure for phonemes */aa/* and */sh/* using the proposed model. As the amount of data increases the system can learn a more complex model for the same phone. It is also important to note that structure learned for each phone is different and reflects underlying differences between phones.

In addition to defining the left-to-right HDP-HMM we can also define HDP-HMMs with HDP emissions. HDP-HMMs defined in (Fox et al., 2011) use a DPM to model the emission distribution for each state. While this model is reasonably flexible, each data point is strictly associated with a single state and hence statistical estimation of each parameter would be less reliable. This is a more serious problem for HDP-HMMs with a left-to-right topology since these models will discover more states. As a result the available data for estimating the emission distribution for each state would be more limited. Using an HDP structure for modeling the emission distribution will address this problem and can potentially improve the overall performance of both ergodic and non-ergodic HDP-HMMs.

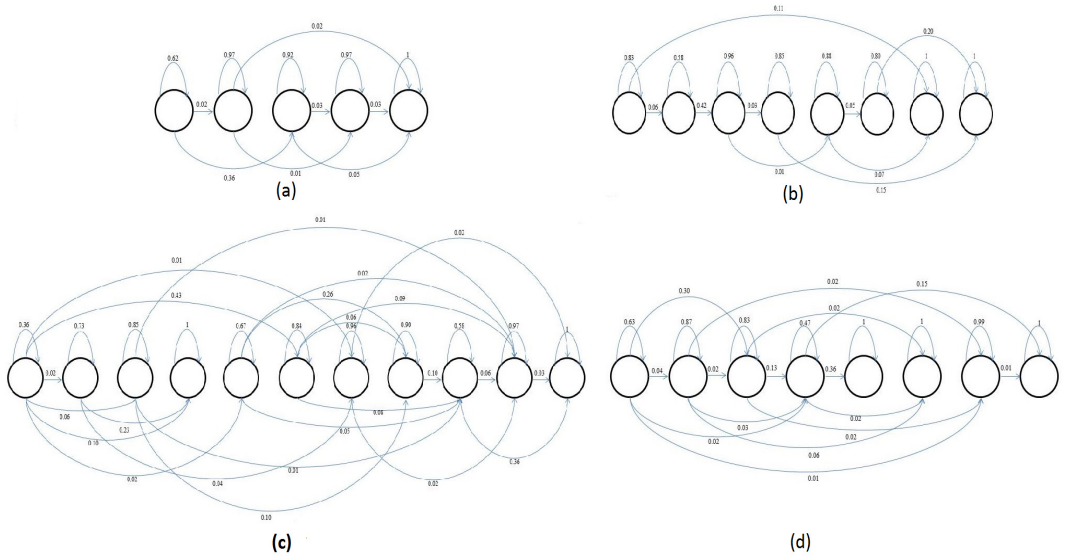

Figure 5 – An automatically derived model structure (without the first and last dummy states) for (a) */aa/* with 175 examples (b) */sh/* with 100 examples (c) */aa/* with 2256 examples and (d) */sh/* with 1317 examples using left-to-right HDP-HMM model. The data used in this illustration was extracted from the training portion of the TIMIT Corpus.

Finally, in some applications, such as continuous speech recognition, we need dummy states (states without emissions) at the beginning and end of the model. Models always start from a dummy state and end in a dummy state. This allows us to connect different HMMs as part of the process of training systems on continuous speech 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 more difficult. We have to estimate the probability of transition from each state to the output dummy state. This estimate 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 models just one sequence of events. These events can happen in different orders but if we have two separate sequences we have to model them separately. Left-to-right HMMs, on the other hand, model ordered sequences of events with a start and an end. Therefore a single HMM can model several sequences. This also opens the door to address two interesting issues. First, the state's labels will not be arbitrary and therefore there is no label switching problem. Secondly, as a consequence, it is possible (though perhaps with some heuristics) to use a straightforward 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 the 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 off-line training process and does not impact the recognition part of the system. We will test our proposed model on two types of tasks – phone recognition and isolated word recognition. The reason we chose

these applications was to focus our investigation on modeling capabilities of the proposed model and to avoid interactions with other portions of a speech recognition system.

Chapter 6

NONPARAMETRIC BAYESIAN TRAINING

6.1 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 process (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. We then train these concatenated models as one big HMM. This method makes it possible to avoid using phoneme-level transcriptions (which are very difficult to produce) for acoustic model training. Therefore we want the training procedure for nonparametric Bayesian models to also have 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 tying 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 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, models or components of the models are tied 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 a data-driven approach:

1. A list of all triphones is produced.
2. Using monophone models trained in previous steps, these triphones models are initialized by cloning monophone models.
3. After training these triphone models, corresponding states of all triphones with a similar center phoneme are 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 only a few data points to the closest cluster.
5. Train tied models.
6. For triphones not observed in the training data, use a back-off modeling procedure (Beulen et al., 1997).

Alternatively, we can use phonetic trees to cluster the data (Beulen et al., 1997). In this case, we first group all corresponding states of all triphones with similar center phones. We also provide a pool of phonetic questions (e.g. is the left phoneme a stop?). The clustering is as follows:

1. Put all states in the root node of the tree.
2. Find the best question that divides the node into two nodes and maximizes the local likelihood scores.
3. Iterate for all nodes until increments in the likelihood fall below a threshold. The resulting nodes are called terminal nodes and all states within a terminal node will be tied together.
4. If the number of data points in a node is less than a threshold, combine it with its parent node.
5. Unseen models can be clustered by starting from the root and answering questions until we reach 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.

6.2 Proposed approach

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 derived units) is the same. The algorithm is not also confined to speech data and can be used in other similar problems.

6.2.1 Training A Left-to-right HDP-HMM

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

1. Initialize Z_i either randomly or bootstrap using a conventional system.
2. The result is several sub-sequences. Each sub-sequence will have a unique Z_i. Therefore a sequence of X_i will be converted into a sequence of sub-sequences W_j.
3. For a given sequence of data use the transcription to generate a list of models.
4. Regroup sub-sequences W_j based on their corresponding Z_j and distribute each group to the corresponding HDP-HMM model (M_{Zi}).
5. Train each HDP-HMM using the inference algorithm. Training each left-to-right HDP-HMM involves several sequences of data $\{W_j \mid Z_j = Z_i\}$. Fortunately, since each left-to-right HDP-HMM has a start dummy state (the first state that does not emit) using multiple sequences in the inference algorithm does not change the algorithm.
6. After all models are trained, reestimate the Z_i for all X_i. 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.

6.2.2 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 that is closely related to the data-driven approach described previously. Here we describe the proposed algorithm:

1. Given the monophone models, train all existing 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 phone.
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 (i.e. use monophones instead of triphones) for unseen triphones.

Chapter 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 proposal.

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 proposal.

Chapter 8

CONCLUSION

In this proposal, 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 algorithm to use left-to-right HDP-HMMs in a continuous speech recognizer application.

Major contributions of this research are: (1) introducing left-to-right HDP-HMMs with HDP emission and corresponding inference algorithm; (2) introducing an algorithm to train left-to-right HDP-HMMs in a continuous speech recognition system; (3) study the performance of a left-to-right HDP-HMM; (4) use a nonparametric clustering approach for state tying; and (5) study the application of the nonparametric Bayesian models in automatic acoustic unit discovery. It is expected that the discovered units will perform at least as good as manually designed units and will not require extensive linguistic knowledge. It is also expected that our nonparametric model improves the word error rate (WER) of the system with respect to conventional ASR systems without increasing the computational complexity of the recognizer.

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 is 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. However, there are many directions that can be pursued in the future. One important and practical problem is to use massive parallel processing (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 have already started to adapt parallel training techniques for the inference algorithm (Williamson et al., 2012; Suchard et al., 2010).

Another direction that is especially relevant in speech is to look into more complicated hierarchical models. Defining new models, under a Bayesian framework, is relatively straightforward. However designing an 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 decreased recognition error rates. Our approach can be considered as a generalization of gender specific modeling. Considering the vast amount of speech data that has become available in recent years, and by considering the huge acoustic diversity that exists in this data (e.g. different speakers, environments), there are significant opportunities to apply nonparametric Bayesian approaches in speech processing.

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.

Gish, H., & Ng, K. (1993). A segmental speech model with applications to word spotting. *proceedings of IEEE international Conference on Acoustics, Speech and Signal Processing* (pp. 447–450). Minneapolis, MN, USA.

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.

Meignier, S., Bonastre, J.-F., & Igouney, S. (2001). E-HMM approach for learning and adapting sound models for speaker indexing. *Proceedings of Odyssey 2006: Speaker and Language Recognition Workshop* (pp. 175–180). Crete, Greece.

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

Shin, W., Lee, B.-S., Lee, Y.-K., & Lee, J.-S. (2000). Speech/non-speech classification using multiple features for robust endpoint detection. *proceedings of IEEE international Conference on Acoustics, Speech and Signal Processing* (pp. 1899–1402). Istanbul, Turkey.

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.