A Doubly Hierarchical Dirichlet Process Hidden Markov Model with a Non-Ergodic Structure

Amir H. Harati Nejad Torbati, Student Member, IEEE, and Joseph Picone, Senior Member, IEEE

Abstract- Nonparametric Bayesian models use a Bayesian framework to learn model complexity automatically from the data, eliminating the need for a complex model selection process. A Hierarchical Dirichlet Process Hidden Markov Model (HDPHMM) is the nonparametric Bayesian equivalent of a hidden Markov model (HMM), but is restricted to an ergodic topology that uses a Dirichlet Process Model to achieve a mixture distribution-like model. For applications involving ordered sequences (e.g., speech recognition), it is desirable to impose a left-to-right structure on the model. In this paper, we introduce a model based on HDPHMM that: (1) shares data points between states, (2) models non-ergodic structures, and (3) models nonemitting states. The first point is particularly important because Gaussian mixture models, which support such sharing, have been very effective at modeling modalities in a signal (e.g., speaker variability). Further, sharing data points allows models to be estimated more accurately, an important consideration for applications such as speech recognition in which some mixture components occur infrequently. We demonstrate that this new model produces a 20% relative reduction in error rate for phoneme classification and an 18% relative reduction on a speech recognition task on the TIMIT Corpus compared to a baseline system consisting of a parametric HMM.

Index Terms— nonparametric Bayesian models; hierarchical Dirichlet processes; hidden Markov models; speech recognition

I. INTRODUCTION

IDDEN Markov models (HMMs) [1] are one of the most successful models for application involves ordered sequences (e.g. speech) and have been applied to a wide range of applications including speech recognition. HMMs, often referred to as doubly stochastic models, are parameterized both in their structure (e.g. number of states) and emission distributions (e.g. Gaussian mixtures). Model selection methods such as the Bayesian Information Criterion (BIC) [2] are traditionally used to optimize the number of states and However, these mixture components. methods are computationally expensive and there is no consensus on an optimum criterion for selection [2].

Beal et al. [3] proposed a nonparametric Bayesian HMM

Manuscript received June 11, 2015; revised September 25, 2015; accepted November 11, 2015. This research was supported in part by the National Science Foundation through Major Research Instrumentation Grant No. CNS-09-58854.

A. Harati is with JIBO, Inc., 805 Veterans Blvd., Suite 303, Redwood, CA, USA, 94063 (email: amir.harati@gmail.com).

J. Picone is with the Department of Electrical and Computer Engineering, Temple University, 1949 N 12th Street, Philadelphia, PA, USA, 19122 (email: joseph.picone@gmail.com). with a countably infinite number of states. This model is known as an infinite HMM (iHMM) because it has an infinite number of hidden states. Teh et al. [4], [5] proposed an alternate formulation, a Hierarchical Dirichlet Process Hidden Markov Model (HDPHMM), based on a hierarchical Dirichlet process (HDP) prior. HDPHMM is an ergodic model – a transition from an emitting state to all other states is allowed. However, in many pattern recognition applications involving temporal structure, such as speech processing, a left-to-right topology is required [6].

For example, in continuous speech recognition applications we model speech units (e.g. phonemes), which evolve in a sequential manner, using HMMs. Since we are dealing with an ordered sequence (e.g. a word is an ordered sequence of phonemes), a left-to-right model is preferred [7]. The segmentation of speech data into these units is not known in advance and therefore the training process must be able to connect these smaller models together into a larger HMM that models the entire utterance. This task can easily be achieved using left-to-right HMMs (LR-HMM) and by utilizing special non-emitting states [8]. Further, these non-emitting states can be used to model finite-length sequences [6]. In the HDPHMM formulation, these problems are not addressed.

An HDPHMM, as well as a parametric HMM, models each emission distribution by data points mapped to that state. For example, it is common to use a Gaussian mixture model (GMM) to model the emission distributions. However, in an HDPHMM, the mixture components of these GMMs are not shared or reused. Sharing of such parameters is a critical part of most state of the art pattern recognition systems. We have introduced a model, which we refer to as Doubly Hierarchical Dirichlet Process Hidden Markov Model (DHDPHMM) [9], with two parallel hierarchies that enables sharing of data among states. In this paper, we introduce a general method to add non-emitting states to both HDPHMM and DHDPHMM. We also develop a framework to learn non-ergodic structures from the data and present comprehensive experimental results. We present a far more comprehensive derivation of DHDPHMM compared to [9]. An open source implementation of our DHDPHMM and HDPHMM algorithms is available at [10].

This paper is organized as follows. In Section II, we provide background on nonparametric Bayesian modeling and formally introduce HDPHMM. In Sections III and IV, we describe DHDPHMM and its extensions for non-ergodic modeling and estimation of non-emitting states. In Section V,

we present results on three tasks: a pilot study using simulated data, a phoneme classification task and a speech recognition task.

II. BACKGROUND

Nonparametric Bayesian models (NPBM) have become increasingly popular in recent years because of their ability to balance model accuracy with generalization. Machine learning algorithms often have trouble dealing with previously unseen data or data sets in which the training and evaluation conditions are mismatched. As a result, models with a large degree of freedom (e.g. complexity) often don't generalize well to new datasets. Adjusting complexity to match the needs of the data is an important requirement for good generalization.

A. Nonparametric Bayesian Models

A Dirichlet process (DP) [11] is a discrete distribution that consists of a countably infinite number of probability masses. DP defines a distribution over discrete distributions with infinite support. A DP, denoted DP(α ,G₀), is defined as [12]:

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

$$\boldsymbol{\beta}_{k} = \boldsymbol{v}_{k} \prod_{l=1}^{k-1} (1 - \boldsymbol{v}_{l}), \ \boldsymbol{v}_{k} \mid \boldsymbol{\alpha} \sim Beta(1, \boldsymbol{\alpha}).$$
⁽²⁾

where G_0 represents the base distribution [11], δ_{θ_k} is the unit impulse function at θ_k , β_k are weights sampled according to (2) [12], and α is a concentration parameter that represents the degree of concentration around the mean (α is inversely proportional to the variance). The impulse functions, δ_{θ_k} , are often referred to as atoms.

In this representation β can be interpreted as a random probability measure over positive integers. The β_k sampled by this process, denoted by $\beta \sim GEM(\alpha)$, are constructed using a stick-breaking process [4]. Starting with a stick of length one, we break each stick 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 .

One of the main applications of a DP is to define a nonparametric prior distribution for the components of a mixture model. The resulting model is referred to as a Dirichlet Process Mixture (DPM) model and is defined as [4]:

$$\pi \mid \alpha \sim GEM(\alpha)$$

$$z_{i} \mid \pi \sim Mult(\pi)$$

$$\theta_{k} \mid G_{0} \sim G_{0}$$

$$x_{i} \mid z_{i}, \{\theta_{k}\} \sim F(\theta_{z_{i}}).$$
(3)

In this model, the observations, x_i , are sampled from an indexed family of distributions, F. If F is assumed to be Gaussian then the result is an infinite Gaussian mixture model, which is the nonparametric counterpart of a GMM [13].

An HDP extends a DPM to problems involving mixture modeling of grouped data [4] in which we desire to share components of these mixture models across groups. An HDP is defined as:

$$G_{0} \mid \gamma, H \sim DP(\gamma, H)$$

$$G_{j} \mid \alpha, G_{0} \sim DP(\alpha, G_{0})$$

$$\theta_{ji} \mid G_{j} \sim G_{j}$$

$$x_{ii} \mid \theta_{ii} \sim F(\theta_{ii}) \quad for \quad j \in J.$$

$$(4)$$

where *H* provides a prior distribution for the factor θ_{ji} , 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. We use a DP to define a mixture model for each group and use a global DP, DP(γ ,*H*), as the common base distribution for all DPs.

B. Hierarchical Dirichlet Process Hidden Markov Model

Hidden Markov models are a class of doubly stochastic processes in which discrete state sequences are modeled as a Markov chain [1]. If we denote the state at time *t* by z_t , the Markovian structure can be represented by $z_t | z_{t-1} \sim \pi_{z_{t-1}}$, where $\pi_{z_{t-1}}$ is the multinomial distribution that represents a transition from state *t*-1 to state *t*. Observations are conditionally independent and are denoted by $x_t | z_t \sim F(\theta_{z_t})$. In a typical parametric HMM, the number of states is fixed so that a matrix of dimension *N* states by *N* transitions per state is used to represent the transition probabilities.

An HDPHMM is an extension of an HMM in which the number of states can be infinite. At each state z_t we can transition to an infinite number of states so the transition distribution should be drawn from a DP. However, in an HDPHMM, to obtain a chain process, we want reachable states from one state to be shared among all states so these DPs should be linked together. In an HDPHMM each state corresponds to a group 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 HDPHMM [4] is state persistence. HDPHMM has a tendency to make many redundant states and switch rapidly amongst them. Fox et al. [5] extended the definition of HDPHMM to HMMs with state persistence by introducing a sticky parameter κ :

$$\beta \mid \gamma \sim GEM(\gamma)$$

$$\pi_{j} \mid \alpha, \beta, \kappa \sim DP(\alpha + \kappa, \frac{\alpha\beta + \kappa\delta_{j}}{\alpha + \kappa})$$

$$\psi_{j} \mid \sigma \sim GEM(\sigma)$$

$$\theta_{kj}^{**} \mid H, \lambda \sim H(\lambda)$$

$$z_{i} \mid z_{i-1}, \{\pi_{j}\}_{j=1}^{\infty} \sim \pi_{z_{i-1}}$$

$$s_{i} \mid \{\psi_{j}\}_{j=1}^{\infty}, z_{i} \sim \psi_{z_{i}}$$

$$x_{i} \mid \{\theta_{kj}^{**}\}_{k,j=1}^{\infty}, z_{i}, s_{i} \sim F(\theta_{z_{i}}s_{i})$$
(5)

The state, mixture component and observations are represented by z_t , s_t and x_t respectively. The indices j and k are indices of the states and mixture components respectively. The base distribution, β , can be interpreted as the expected value of state transition distributions. The transition distribution for state j is a DP denoted by π_j with a concentration parameter α . Another DP, ψ_j , with a concentration parameter σ , is used to model an infinite mixture model for each state z_j . The distribution H is the prior for the parameters θ_{kj} .

A block sampler for HDPHMM with a multimodal emission distribution has been introduced [5] that jointly samples the state sequence $z_{I:T}$ given the observations, model parameters and transition distribution π_j . A variant of the forward-backward procedure is utilized that allows us to exploit the Markovian structure of an HMM to improve the convergence speed of the inference algorithm. However this algorithm requires an approximation of the theoretically infinite number of distributions with a "degree *L* weak limit" that truncates a DP to a Dirichlet distribution with *L* dimensions [14]:

$$GEM_L(\alpha) \triangleq Dir\left(\frac{\alpha}{L}, ..., \frac{\alpha}{L}\right).$$
 (6)

It should be noted that this result is different from a classical parametric Bayesian HMM since the truncated HDP priors induce a shared sparse subset of the L possible states [5].

III. DHDPHMM

We can extend the model in (5) to address the problem of sharable mixture components. Equation (5) defines a model with a multimodal distribution at each state. In an HDPHMM formulation these distributions are modeled using DPM:

$$\begin{split} \boldsymbol{\psi}_{j} \mid \boldsymbol{\sigma} \sim \boldsymbol{G}\boldsymbol{E}\boldsymbol{M}\left(\boldsymbol{\sigma}\right) \\ \boldsymbol{s}_{t} \mid \left\{\boldsymbol{\psi}_{j}\right\}_{j=1}^{\infty}, \boldsymbol{z}_{t} \sim \boldsymbol{\psi}_{\boldsymbol{z}_{t}} \\ \boldsymbol{\theta}_{kj}^{**} \mid \boldsymbol{H}, \boldsymbol{\lambda} \sim \boldsymbol{H}\left(\boldsymbol{\lambda}\right) \\ \boldsymbol{x}_{t} \mid \left\{\boldsymbol{\theta}_{kj}^{**}\right\}_{k,j=1}^{\infty}, \boldsymbol{z}_{t}, \boldsymbol{s}_{t} \sim \boldsymbol{F}\left(\boldsymbol{\theta}_{\boldsymbol{z},\boldsymbol{s}_{t}}\right) \,. \end{split}$$
(7)

Equation (7) demonstrates that when the state assignment, z_t , for data point x_t is known (or sampled previously), the mixture components can be sampled from a multinomial distribution with DP priors. Equation (5) also shows that each emission distribution is modeled independently of other distributions. It has been shown previously [16] that sharing data points, if done properly, can improve the accuracy of the model.

As we have discussed in Section II, HDP is the extension of a DPM to mixture modeling of grouped data. If the state assignment, z_i , is assumed to be known (or estimated) then an HDPHMM divides the data points into multiple groups. Therefore, we should be able to use the same principle and model the emission distributions with another HDP. The resulting model will have two parallel hierarchies and hence is referred to as a Doubly Hierarchical Dirichlet Process Hidden Markov Model (DHDPHMM). Applying (4) we can write:

$$\begin{aligned} \xi \mid \tau \sim GEM(\tau) \\ \psi_{j} \mid \sigma, \xi \sim DP(\sigma, \xi) \\ \theta_{kj}^{**} \mid H, \lambda \sim H(\lambda) \\ s_{t} \mid \left\{\psi_{j}\right\}_{j=1}^{\infty}, z_{t} \sim \psi_{z_{t}} \\ x_{t} \mid \left\{\theta_{kj}^{**}\right\}_{k, i=1}^{\infty}, z_{t} \sim F\left(\theta_{z_{t}s_{t}}\right), \end{aligned}$$

$$(8)$$

where ξ is the DP used as the base distribution for HDP and τ and σ are hyperparameters. By substituting (8) in (5) we can obtain a generative model for DHDPHMM:

$$\beta | \gamma \sim GEM(\gamma)$$

$$\pi_{j} | \alpha, \beta, \kappa \sim DP(\alpha + \kappa, \frac{\alpha\beta + \kappa\delta_{j}}{\alpha + \kappa})$$

$$\xi | \tau \sim GEM(\tau)$$

$$\psi_{j} | \sigma, \xi \sim DP(\sigma, \xi)$$

$$\theta_{kj}^{**} | H, \lambda \sim H(\lambda)$$

$$z_{t} | z_{t-1}, \{\pi_{j}\}_{j=1}^{\infty} \sim \pi_{z_{t-1}}$$

$$s_{t} | \{\psi_{j}\}_{j=1}^{\infty}, z_{t} \sim \psi_{z_{t}}$$

$$x_{t} | \{\theta_{kj}^{**}\}_{k,j=1}^{\infty}, z_{t}, s_{t} \sim F(\theta_{z_{t}s_{t}}).$$
(9)

DHDPHMM pools the data points while HDPHMM divides data points between different states. If we don't have enough data points for a particular state or mixture component, then the distribution parameters (e.g. the mean and covariance) will be estimated poorly. For example, in speech recognition systems we usually use features with a dimensionality of 39 which translates to 39+(39x40)/2+1=820 free parameters per Gaussian mixture component (assuming a full covariance). In an HDPHMM, with no sharing of parameters, we can easily end up with an intractable number of parameters.

It should be noted that an earlier version of the generative model described in (9) was first introduced in [9] as part of an experimental study. In this paper, we provide theoretical justifications for the model and present more detailed experimental results supporting the efficacy of the model.

A. Inference Algorithm for DHDPHMM

An inference algorithm is required to learn the model parameters from the data. One solution to this problem is the block sampler [5] discussed in the previous section. Using the "degree L weak limit" approximation to DP in (6) for HDP emissions of (8) we can write the following equations (replacing L' with L):

$$\boldsymbol{\xi} \mid \boldsymbol{\sigma} \sim Dir\left(\frac{\boldsymbol{\sigma}}{L'}, \dots, \frac{\boldsymbol{\sigma}}{L'}\right) \tag{10}$$

$$\boldsymbol{\psi}_{j} \mid \boldsymbol{\xi}, \boldsymbol{\tau} \sim Dir(\boldsymbol{\tau}\boldsymbol{\xi}_{1}, ..., \boldsymbol{\tau}\boldsymbol{\xi}_{L'}). \tag{11}$$

Following a similar approach in [5] we write the posterior distributions for these equations as:

$$\boldsymbol{\xi} \mid \boldsymbol{M}', \boldsymbol{\tau} \sim Dir\left(\frac{\boldsymbol{\tau}}{L'} + \boldsymbol{M}'_{\boldsymbol{\cdot}1}, \dots, \frac{\boldsymbol{\tau}}{L'} + \boldsymbol{M}'_{\boldsymbol{\cdot}L'}\right)$$
(12)

$$\psi_{j} \mid \sigma, \xi, Z_{LT}, S_{LT} \sim Dir \left(\sigma_{\xi_{1}}^{\xi} + n'_{j1}, \dots, \sigma_{\xi_{L'}}^{\xi} + n'_{jL'} \right)$$
(13)

where M'_{jk} is the number of clusters in state *j* with mixture component *k*; $M'_{.k}$ is total number of clusters that contain mixture component *k*. The number of observations in state *j* that are assigned to component *k* is denoted by n'_{jk} . The posterior distribution for τ , the hyperparameter in (12), is:

$$P(\tau \mid n_{1}, ..., n_{j_{\star}}, M'_{1}, ..., M'_{j_{\star}}) \propto Gamma \left(a + M'_{\star} - \sum_{j=1}^{L'} s_{j} b - \sum_{j=1}^{L'} \log r_{j} \right)$$
(14)

$$P(r_{j} \mid \tau, \mathbf{r}_{j}, \boldsymbol{s}, n_{1}, \dots, n_{J}, M'_{1}, \dots, M'_{J}) \propto Beta(\tau + 1, n_{j})$$
(15)

$$P\left(s_{j} \mid \tau, \mathbf{s}_{i}, \boldsymbol{r}, n_{i}, \dots, n_{j}, M'_{i}, \dots, M'_{j}\right) \propto Ber\left(\frac{n_{j}}{n_{j}} + \tau\right)$$
(16)

where *r* and *s* are auxiliary variables used to facilitate the inference for τ (following the same approach as in [5]) and *a* and *b* are hyperparameters over a Gamma distribution.

The first derivation of the modified block sampler has been presented in [9]. However, that algorithm did not include sampling of hyperparameters and its implementation did not utilize DHDPHMM properties to reduce the computational cost. Further in this new implementation, in addition to the block sampler, we used the Expectation-Maximization (EM) algorithm after performing the inference using a block sampler. This allows us to reduce the number of iterations required for the block sampler.

The main motivation behind DHDPHMM is the ability to share mixture components and therefore data points between different states. When using the modified block sampler algorithm, we only deal with L' Gaussian distributions. HDPHMM requires estimation of LxL' Gaussians. Since as much as 95% of the inference time is spent calculating the likelihood of the data for Gaussian distributions, a reduction

from LxL' to L' reduces the computational time considerably. Unlike EM forward-backward computations do not dominate the computation time.

We have utilized parallel programming facilities (e.g. openMP) for the implementation of both algorithms, which makes this process feasible for moderate size datasets. Figure 1 provides a comparison of both algorithms for different values of L and L' for phoneme /zh/. DHDPHMM's computational complexity is relatively flat as the maximum bound on the number of states increases while the inference cost for HDPHMM grows much faster. For example, for L'=200, computation time for HDPHMM grows 16.6 times when L increases from 5 to 200 while the computation time for DHDPHMM only increases 2.4 times.

IV. DHDPHMM WITH A NON-ERGODIC STRUCTURE

A non-ergodic structure for DHDPHMM can be achieved by modifying the transition distributions. These modifications can also be applied to HDPHMM using a similar approach. It should be noted that in [9] we introduced only the left-to-right structure with a loop transition. In the following section several other essential structures are introduced.

A. Left-to-Right DHDPHMM

The transition probability from state j has infinite support and can be written as:

$$\pi_{j} \mid \alpha, \beta \sim DP(\alpha + \kappa, \frac{\alpha\beta + \kappa\delta_{j}}{\alpha + \kappa}).$$
(17)

We observe that a transition distribution has no topological restrictions and therefore (5) and (9) define ergodic HMMs. In order to obtain a left-to-right (LR) topology we need to force the base distribution of the Dirichlet distribution in (17) to only contain atoms to the right of the current state. This means β should be modified so that the probability of transiting to states left of the current state (i.e. states previously visited) becomes zero. For state *j* we define $V_i = \{V_{ij}\}$:

$$V_{ji} = \begin{cases} 0, & i < j \\ 1, & i \ge j \end{cases}$$
(18)



Fig. 1. DHDPHMM improves scalability relative to HDPHMM. *L* represents the upper bound for the number of states and *L'* represents the upper bound for the maximum number of mixture components per state.

where *i* is the index for all following states. We can then modify β by multiplying it with V_j :

$$\beta'_{j} = \frac{\beta \cdot V_{j}}{\sum_{i} \beta_{i} V_{ji}}.$$
(19)

In the block sampler algorithm, we have:

$$\pi_{j} \sim Dir\left(\alpha\beta_{j1}' + n_{j1}, ..., \alpha\beta_{jj}' + \kappa + n_{jj}, ..., \alpha\beta_{jL}' + n_{jL}\right), \qquad (20)$$
$$j = 1, ..., L,$$

where n_{jk} are the number of transitions from state *j* to *k*. From (20) we can see that multiplying β with V_j biases π_j toward a left-to-right structure but there is still a positive probability to transit to the states left of *j* (asymptotically this probability tends to zero but for relatively small number of iterations, depending on implementation we might get non-zero values). If we leave π_j as in (20) the resulting model would be an LR model with possible loops. Models with an LR structure and possible loops will be denoted as LR-L.

In order to obtain an LR model with no loops, we have to multiply n_{jk} with V_j :

$$\pi_{j} \sim Dir(\alpha\beta_{j1}' + V_{j1}n_{j1}, ..., \alpha\beta_{jj}' + \kappa + V_{jj}n_{jj}, ..., \alpha\beta_{jL}' + V_{jL}n_{jL}), \quad (21)$$

$$j = 1, ..., L,$$

 V_j and β' are calculated from (18) and (19) respectively. This model always finds transitions to the right of state *j* and is referred to as an LR model.

Sometimes it is useful to have LR models that allow restricted loops to the first state. For example, when dealing with long sequences, a sequence might have a local left to right structure but needs a reset at some point in time. To modify β to obtain an LR model with a loop to the first state (LR-LF) we can write:

$$V_{ji} = \begin{cases} 0, & 0 < i < j \\ 1, & i \ge j, i = 0 \end{cases}$$
 (22)

where β' is calculated from (19) and π_i is sampled from (21).

The LR models described above allow skip transitions to learn parallel paths corresponding to different modalities in the training data. Sometimes more restrictions on the structure might be required. One such example is a strictly left to right structure (LR-S):

$$V_{ji} = \begin{cases} 0, & i \neq j+1 \\ 1, & i = j+1 \end{cases}.$$
 (23)

B. Initial and Final Non-Emitting States

In many applications, such as speech recognition, an LR-HMM begins from and ends with a non-emitting state. These states are required to model the beginning and end of finite duration sequences. More importantly, in many applications like speech recognition, we need to generate composite HMMs [8] that present a larger segment of the observations (e.g. for example in speech recognition an HMM models a phoneme while a composite HMM models a whole utterance). Without non-emitting initial and final states, building a composite HMM would be much more difficult. Therefore, virtually all HMM implementations for speech recognition applications, including HTK [8] and Kaldi [15], support modeling of initial and final non-emitting states. Adding a non-emitting initial state is straightforward: the probability of transition into the initial state is 1 and the probability distribution of a transition from this state is equal to π_{init} , which is the initial probability distribution for an HMM without non-emitting states. However, adding a final nonemitting state is more complicated. In the following sections we will discuss two approaches that solve this problem.

1) Maximum Likelihood Estimation

Consider state z_i depicted in Fig. 2. The outgoing probabilities for any state can be classified into three categories: (1) a self-transition (P_1), (2) a transition to all other states (P_2), and (3) a transition to a final non-emitting state (P_3). These probabilities must sum to 1: $P_1+P_2+P_3=1$. Suppose that we obtain P_2 from the inference algorithm. We will need to reestimate P_1 and P_3 from the data. This problem is, in fact, equivalent to the problem of tossing a coin until we obtain the first tails. Each head is equal to a self-transition and the first tail triggers a transition to the final state. This can be modeled using a geometric distribution [17]:

$$P(x=k) = (1-\rho)^{k-1} \rho.$$
(24)

Equation (24) shows the probability of *K*-1 heads before the first tail. In this equation $1-\rho$ is the probability of heads (success). We also have:

$$\frac{P_1}{1 - P_2} = 1 - \rho, \qquad \frac{P_3}{1 - P_2} = \rho.$$
(25)

Suppose we have a total of N examples but for a subset of these, M_i , the state z_i is the last state of the model (S_M) . It can be shown [17] that the maximum likelihood estimation is obtained by:

$$\hat{\rho}_i = \frac{M_i}{\sum_{j \in S_M} k_j} \tag{26}$$

where k_i are the number of self-transitions for state *i*. Notice that if z_i is never the last state, then $M_i = 0$ and $P_3 = 0$.

2) Bayesian Estimation

ŀ

Another approach to estimate transitions to a final nonemitting state, ρ_i , is to use a Bayesian framework. Since a beta distribution is the conjugate distribution for a geometric distribution, we can use a beta distribution [19] with hyperparameters (a,b) as the prior and obtain a posterior as:

$$P_{i} \sim Beta\left(a + M_{i}, b + \sum_{j \in S_{M}} (k_{j} - 1)\right)$$

$$P_{i}: \text{Probability of self-transition.}$$

$$P_{j}: \text{Probability of transition}$$

$$P_{j}: \text{Probability of transition}$$

$$P_{j}: \text{Probability of transition to a}$$

where M_i is the number of times which state z_i was the last state and S_M is the set of all such last states. Hyperparameters (a,b) can also be estimated using a Gibbs sampler if required [20]. If we use (27) to estimate ρ_i we need to modify (20) to impose the constraint that the sum of the transition probabilities add to 1. This is a relatively simple modification based on the stick-breaking interpretation of a DP in (2). This modification is equal to assigning ρ_i to the first break of the stick and then breaking the remaining $1-\rho_i$ portion as before.

C. An Integrated Model

The final definition for DHDPHMM with a non-ergodic structure is given by:

$$\beta \mid \gamma \sim GEM(\gamma), \beta'_{j} = \frac{V_{j} \cdot \beta}{\sum_{i} V_{ji} \beta_{i}}$$

$$\pi_{j} \mid \alpha, \beta'_{j} \sim DP(\alpha + \kappa, \frac{\alpha \beta'_{j} + \kappa \delta_{j}}{\alpha + \kappa})$$

$$\xi \mid \tau \sim GEM(\tau)$$

$$\psi_{j} \mid \sigma, \xi \sim DP(\sigma, \xi)$$

$$\theta_{kj}^{**} \mid H, \lambda \sim H(\lambda)$$

$$z_{t} \mid z_{t-1}, \{\pi_{j}\}_{j=1}^{\infty} \sim \pi_{z_{t-1}}$$

$$s_{t} \mid \{\psi_{j}\}_{j=1}^{\infty}, z_{t} \sim \psi_{z_{t}}$$

$$x_{t} \mid \{\theta_{kj}^{**}\}_{k,j=1}^{\infty}, z_{t} \sim F(\theta_{z,s_{t}})$$
(28)

 V_i should be replaced with the appropriate definition from the previous section based on the desired type of structure. For example, for an LR model, V_i should be sampled from (18). Note that by setting V_i to 1 we obtain the ergodic DHDPHMM in (9). A comparison of models is shown in Fig. 3.

The model described in (28) [9] did not incorporate modeling of non-emitting states as discussed above. If we choose to use a maximum likelihood approach for estimating the non-emitting states, then no change to this model is



Fig. 3. Comparison of an (a) ergodic HDPHMM[5] and (b) DHDPHMM.

required (e.g. we can estimate these non-emitting states after estimating other parameters). However, if we choose to use the Bayesian approach then we have to replace the sampling of π_i in (28) with:

$$w, \chi \sim MSB(\alpha, \beta, \kappa, j)$$

$$\pi_{j} \mid w, \chi \sim \sum_{k} w_{k} \delta_{\chi_{k}}, \qquad (29)$$

where MSB() is a modified stick-breaking process. Equations (1) and (2) show the basic stick-breaking algorithm – start with a stick of length 1 and then break the stick consecutively to obtain the weights in (2). The locations of atoms, represented by δ in (2), are sampled independently from another distribution – G_0 . In MSB(), we start from a stick of length $(1-\rho_j)$ and sample the atoms from a discrete distribution that represents the transition probabilities:

$$MSB(\alpha, \beta'_{j}, \kappa, j) = \begin{cases} for \ i = \{1, 2, ...\}: \\ v_{i} \mid \alpha, \kappa \sim Beta(1, \alpha + \kappa) \\ w_{i} \mid v_{i}, \rho_{j} = v_{i}(1 - \rho_{j}) \prod_{l=1}^{i-1} (1 - v_{l}) \\ \chi_{i} \mid \alpha, \beta'_{j}, \kappa \sim \sum_{k} \frac{\alpha \beta'_{jk} + \kappa \delta_{kj}}{\alpha + \kappa} \delta_{k}, \end{cases}$$
(30)

where v_i are sequences of independent variables drawn from a Beta distribution and w_i are stick weights. χ_i is the location of the atom that represents a transition to another state. χ_i determines which state we will transit to while w_i determines what is the probability to transit to this state.

By substituting (29) and (30) in (28) we can obtain a generative model that incorporates Bayesian modeling of the non-emitting states:

$$\beta \mid \gamma \sim GEM(\gamma)$$

$$\beta'_{j} \mid V_{j} = \frac{V_{j} \cdot \beta}{\sum_{i} V_{ji} \beta_{i}}$$

$$\rho_{j} \mid a, b, M_{j}, S_{M} \sim Beta \left(a + M_{i}, b + \sum_{j \in S_{M}} \left(k_{j} - 1 \right) \right)$$

$$\overline{w}, \overline{\chi} \sim MSB \left(\alpha, \beta'_{j}, \kappa, j \right)$$

$$\pi_{j} \mid \overline{w}, \overline{\chi} \sim \sum_{k} w_{k} \delta_{\chi_{k}}$$

$$\xi \mid \tau \sim GEM(\tau)$$

$$\psi_{j} \mid \sigma, \xi \sim DP(\sigma, \xi)$$

$$\theta_{kj}^{**} \mid H, \lambda \sim H(\lambda)$$

$$z_{t} \mid z_{t-1}, \left\{ \pi_{j} \right\}_{j=1}^{\infty} \sim \pi_{z_{t-1}}$$

$$s_{t} \mid \left\{ \psi_{j} \right\}_{j=1}^{\infty}, z_{t} \sim \psi_{z_{t}}$$

$$x_{t} \mid \left\{ \theta_{kj}^{**} \right\}_{k,j=1}^{\infty}, z_{t} \sim F\left(\theta_{z,s_{t}} \right),$$
(31)

where we have replaced DP with the modified stick-breaking process described above. Most of the results discussed above, including the inference algorithm, hold for this model as well.

V. EXPERIMENTS

In this section we provide some experimental results which compare DHDPHMM with HDPHMM, HMM and several other state of the art models.

A. HMM-Generated Data

To demonstrate the basic efficacy of the model, we generated data from a 4-state left to right HMM. The emission distribution for each state is a GMM with a maximum of three components, each consisting of a two-dimensional normal distribution. Three synthetic data sequences totaling 1,900 observations were generated for training. Three configurations have been studied: (1) an ergodic HDPHMM, (2) an LR HDPHMM and (3) an LR DHDPHMM. A Normal-inverse-Wishart distribution (NIW) prior is used for the mean and covariance. The truncation levels are set to 10 for both the number of states and the number of mixture components.

Figure 4-a shows the average likelihood for different models for held-out data by averaging five independent chains. Figure 4-b compares the trained model to the reference structure. The LR DHDPHMM discovers the correct structure while the ergodic HDPHMM finds a more simplified HMM. The LR DHDPHMM constrains the search space to left to right topologies while HDPHMM has a less constrained search space.

Further, we can see that DHDPHMM has a higher overall likelihood. While LR HDPHMM can find a structure close to the correct one, its likelihood is slightly lower than the ergodic HDPHMM. This happens because LR HDPHMM needs to estimate more parameters and for a small amount of training data its accuracy is lower than HDPHMM. However, LR DHDPHMM produces a 15% (relative) improvement in likelihoods compared to the ergodic model.

This simple experiment suggests that sharing data points, especially for non-ergodic structures, could be important. LR HDPHMM finds more Gaussian components relative to an ergodic model but each Gaussian can potentially be estimated using fewer data points. By sharing mixture components, LR DHDPHMM implements a form of regularization that prevents the model from over-fitting. In the next section we examine the relative importance of each of these features in more detail. Also notice all non-parametric models perform better than a parametric HMM trained using EM.

B. Phoneme Classification on the TIMIT Corpus

The TIMIT Corpus [21] is one of the most cited evaluation data sets used to compare new speech recognition algorithms. The data is segmented manually into phonemes and therefore is a natural choice to evaluate phoneme classification algorithms. TIMIT contains 630 speakers from eight main dialects of American English. There are a total of 6,300 utterances where 3,990 are used in the training set and 192 utterances are used for the "core" evaluation subset (another 400 used as development set). We followed the standard practice of building models for 48 phonemes and then map them into *39* phonemes [22]. A standard 39-dimensional MFCC feature vector was used (12 Mel-frequency Cepstral Coefficients plus energy and their first and second derivatives) to convert speech data into feature streams. Cepstral mean subtraction [8] was also used.

For TIMIT classification and recognition experiments, we have used 10 chains. However, even using only one chain is often sufficient. The number of iterations for each chain is set to 400 iterations where we have thrown away the first 200 iterations and used the next 200 iterations to obtain the expected value for the parameters. For both HDPHMM and DHDPHMM the upper bounds for maximum number of states and unique Gaussians per model (not per state) are set to 200. After learning the structure of the model and estimating its posterior distribution, we have used a few iterations of EM (using HTK [8]). This allows a relatively small number of expensive block sampler iterations to be used.

To reduce the number of required iterations for the block sampler, we have used a small subset of the training data to determine a good range of values for hyperparameters and then for each chain we have initialized the chains with values in this range. We have used nonconjugate priors for the Gaussians and placed a normal prior on the mean and an



Fig. 4. Comparison of (a) log-likelihoods of the models, and (b) the corresponding model structures.

inverse-Wishart distribution prior on the covariance. Parameters of these priors (e.g. mean and covariance) are computed empirically using the training data. For the inverse-Wishart distribution, the degrees of freedom can change between 100 to 1000 (250 works best for most models). For the concentration parameters we have placed Gamma priors with values from (10,1) to (3,0.1). Finally, we have placed a Beta distribution with parameters (10,1) for the self-transition parameter. It should also be noted that for classification experiments we have used a MAP decoder (e.g. multiplying the likelihood by the prior probability of each class estimated on the training set).

1) A Comparison to HDPHMM

In Table I we compare the performance of DHDPHMM to HDPHMM. We provide error rates for both the development and core subsets. In this table we have compared HDPHMM and DHDPHMM models with ergodic and LR structures. It can be seen that the ergodic DHDPHMM is slightly better than an ergodic HDPHMM. LR HDPHMM is much better than an ergodic HDPHMM. However, when we also allow sharing of mixture components in LR DHDPHMM we obtain the best results (more than 4% absolute improvement). This happens because LR models tend to discover more Gaussians (4,628 for LR HDPHMM vs. 1,850 for ergodic HDPHMM) which means some of the Gaussians will only have a few observations associated with them.

One way to regulate this is to allow some of the Gaussians to be shared among states. Our LR DHDPHMM model explicitly supports this. LR DHDPHMM reduces the number of unique Gaussians to 3,888 and also shows significant improvement relative to LR HDPHMM. This is an important property that validates the basic philosophy of the NPBM and also follows Occam's Razor [22]. From the table we can see that an ergodic DHDPHMM finds a lower number of unique Gaussians relative to LR DHDPHMM. This is due to the fact that an ergodic model is usually more compact and it reuses states while the LR model creates new parallel paths. A strictly LR topology is significantly worse than the others because instead of discovering parallel paths it is constrained to learn one long path.

Figure 5 shows the structures for phonemes /aa/ and /sh/ discovered by our model. It is clear that the model structure evolves with amount of data points, validating another characteristic of the NPBM. It is also important to note that the structure learned for each phoneme is unique and reflects underlying differences between phonemes. Finally, note that the proposed model learns multiple parallel left-to-right paths.

 TABLE I

 COMPARISON OF LR DHDPHMM WITH HDPHMM

Madal	Dev Set	Core Set	No.	
wodel	(% Error)	(% Error)	Gauss.	
Ergodic HDPHMM	24.3%	25.6%	1,850	
LR HDPHMM	23.5%	24.4%	4,628	
Ergodic DHDPHMM	24.0%	25.4%	2,704	
LR-S DHDPHMM	39.0%	38.4%	2,550	
LR DHDPHMM	20.5%	21.4%	3,888	



Fig. 5. An automatically derived model structure for a left-to-right DHDPHMM model (without the first and last non-emitting states) for (a) /aa/ with 175 examples (b) /aa/ with 2,256 examples (c) /sh/ with 100 examples and (d) /sh/ with 1,317 examples.

This is shown in Fig. 5-b where *S1-S2*, *S1-S3* and *S1-S4* depict three parallel models.

2) A Comparison to Other Representative Systems

Table II shows a full comparison between DHDPHMM and both baseline and state of the art systems. The first three rows of this table show three-state LR HMMs trained using maximum likelihood (ML) estimation. HMM with 40 Gaussians per state performs better than the other two and has an error rate of 26.1% on the core subset.

Our LR DHDPHMM model has an error rate of 21.4% on the same subset of data (a 20% relative improvement). It should be noted that the number of Gaussians used by this HMM system is 5,760 (set a priori) while our LR DHDPHMM uses only 3,888 Gaussians. Also note that an HMM with 27 components per state (3,888 total Gaussians) produces an error

TABLE II Comparison of Phoneme Classification Performance

Model	Disc.	Dev Set (% Err)	Core Set (% Err)
HMM (10 Gauss.)	No	28.4%	28.7%
HMM (27 Gauss.)	No	25.4%	26.4%
HMM (40 Gauss.)	No	25.0%	26.1%
HMM/MMI (20 Gauss.) [22]	Yes	23.2%	24.6%
HCRF/SGD [22]	Yes	20.3%	21.7%
Large Margin GMMs [24]	Yes	-	21.1%
GMMs/Full Cov. [24]	No	-	26.0%
SVM [25]	Yes	-	22.4%
Data-driven HMM [26]	No	-	21.4%
LR DHDPHMM	No	20.5%	21.4%



Fig. 6. Error rate vs. amount of training data for LR DHDPHMM and LR HMM.

rate of 26.4% on the core set. Figure 6 shows the error rate vs. the amount of training data for both HMM and DHDPHMM systems. As we can see DHDPHMM is always better than the HMM model. For example, when trained only using 40% of the data, DHDPHMM performs better than an HMM using the entire data set. Also it is evident that HMM performance does not improve significantly when we train it with more than 60% of the data (error rates for 60% and 100% are very close) while DHDPHMM improves with more data.

Figure 7 shows the number of Gaussians discovered by DHDPHMM versus the amount of data. The model evolves into a more complex model as it is exposed to more data. This growth in complexity is not linear (e.g. number of Gaussians grows 33% when the amount of data increases 5 times) which is consistent with the DP prior constraints. If we want to change this behavior, we would have to use a different prior.

The fourth row of Table II shows the error rate for an HMM trained using a discriminative objective function (e.g. MMI). We can see discriminative training reduces the error rate. However, the model still produces a larger error rate relative to the generatively trained DHDPHMM. This suggests that we can further improve DHDPHMM if we use discriminative training techniques. Several other state of the art systems are shown that have error rates comparable to our model. Data-driven HMMs [26], unlike DHDPHMM, model the context implicitly. We expect to obtain better results if we also use context dependent (CD) models instead of context independent (CI) models.

C. Supervised Phoneme Recognition

Speech recognition systems usually use a semi-supervised method to train acoustic models. By semi-supervised we refer to the fact that the exact boundaries between phonemes are not given. The transcription only consists of a sequence of phones in an utterance. It has been shown that this semi-supervised method actually works better than a completely supervised method [26]. However, in this section we use a completely supervised method to evaluate DHDPHMM models for a phoneme recognition task. As in the previous section DHDPHMMs are trained generatively and are trained without context information. In the phoneme recognition problem, unlike phoneme classification, the boundaries between subsequent phonemes are not known (during the recognition phase) and should be estimated along with the phoneme labels. During recognition we have to decide if a given frame belongs to the current group of phonemes under consideration or we have to initiate a new phoneme hypothesis. This decision is made by considering both the likelihood measurements and the language model probabilities. All systems compared in this section use bigram language models. However, the training procedure and optimization of each language model is different and has some effect on the reported error rates.

In the following we define % Error as follows [8]:

$$\% Error = \frac{S + D + I}{N}$$
(32)

where N is the total number of labels in the reference transcriptions, S is the number of substitution errors, D is the number of deletion errors and I is the number of insertion errors.

Table III presents results for several state of the art models. As we can see, systems can be divided into two groups based on their training method (discriminative or not) and context modeling. The first two rows of this table show two similar

TABLE III

COMPARISON OF I HONEME RECOGNITION I EXPORMANCE								
Model	Discr.	Context	% Err	Subset				
CI-HMM [27]	No	No	35.9%	TID7				
CD-HMM 1[27]	No	Yes	26.2%	TID7				
CD-HMM 2[28]	No	Yes	30.9%	Core				
CD-HMM 3[16]	No	Yes	27.7%	Core				
HMM MMI 1 [29]	Yes	No	32.5%	Rand.				
HMM MMI 2/Full Cov. [29]	Yes	No	30.3%	Rand.				
Heter. Class. [30]	Yes	Yes	24.4%	Core				
Data-driven HMM [26]	N/A	Yes	26.4%	Core				
Large Marg. GMM [24]	Yes	No	30.1%	Core				
CRF [31]	Yes	No	29.9%	All				
Tandem HMM [31]	Yes	Yes	30.6%	All				
CNN/CRF [32]	Yes	No	29.9%	Core				
LR DHDPHMM	No	No	29.7%	Core				
LR DHDPHMM	No	No	28.6%	Dev				
LR DHDPHMM	No	No	29.2%	All				

10

T-ASL-05266-2015

HMM based systems with and without contextual information. We can see the error rate drops from 35.9% to 26.2% when we use a system with context modeling. We can also see DHDPHMM works much better than a comparable CI HMM model (the error rate drops from 35.9% for HMM to 28.6% for DHDPHMM).

The third and fourth rows show two context-dependent HMM models. DHDPHMM performs slightly better than the CD model in row three (CD HMM 2) but slightly worse than the CD model of row four (CD HMM 3). We expect to obtain much better results if we use CD models. Our model also performs better than a discriminatively trained CI HMM. By comparing DHDPHMM with other systems presented in Table III we can see DHDPHMM is among the best models for CI systems but is not as good as state of the art CD models.

VI. CONCLUSION

In this paper we introduced a DHDPHMM that is an extension of HDPHMM which incorporates a parallel hierarchy to share data between states. We have also introduced methods to model non-ergodic structures. We demonstrated through experimentation that LR DHDPHMM outperforms both HDPHMM and its parametric HMM counterparts. We have also shown that despite the fact that DHDPHMM is trained generatively its performance is comparable to discriminatively trained models. Further, DHDPHMM provides the best performance among CI models.

Future research will focus on incorporating semi-supervised training and context modeling. We have also shown that complexity grows very slowly with the data size because of the DP properties (only 33% more Gaussians were used after increasing the size of the data five times). Therefore, it makes sense to explore other types of prior distributions to investigate how it can affect the estimated complexity and overall performance. Another possible direction is to replace HDP emissions with more general hierarchical structures such as a Dependent Dirichlet Process [33] or an Analysis of Density (AnDe) model [34]. It has been shown that the AnDe model is the appropriate model for problems involving sharing among multiple sets of density estimators [4], [22].

ACKNOWLEDGEMENTS

The authors wish to thank Professor Marc Sobel of the Department of Statistics at Temple University for many valuable discussions on these topics. The authors also thank the anonymous reviewers for their constructive comments and feedback.

REFERENCES

- L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286, 1989.
- [2] J. B. Kadane and N. A. Lazar, "Methods and Criteria for Model Selection," *Journal of the ASA*, vol. 99, no. 465, pp. 279-290, 2004.
- [3] M. Beal, Z. Ghahramani, and C. E. Rasmussen, "The Infinite Hidden Markov Model," *Proceedings of NIPS*, 2002, pp. 577-584.
- [4] Y. W. Teh, M. Jordan, M. Beal, and D. Blei, "Hierarchical Dirichlet Processes," *Journal of the ASA*, vol. 101, no. 47, pp. 1566-1581, 2006.
- [5] E. Fox, E. Sudderth, M. Jordan, and A. Willsky, "A Sticky HDP-HMM with Application to Speaker Diarization.," *The Annalas of Applied Statistics*, vol. 5, no. 2A, pp. 1020-1056, 2011.
- [6] G. A. Fink, "Configuration of Hidden Markov Models From Theory to Applications," *Markov Models for Pattern Recognition*, Springer Berlin Heidelberg, 2008, pp. 127-136.
- [7] B.-H. Juang and L. Rabiner, "Hidden Markov Models for Speech Recognition," *Technometrics*, vol. 33, no. 3, pp. 251-272, 1991.
- [8] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollagson, D. Povey, V. Valtchev, and P. Woodland, "The HTK Book," Cambridge, UK, 2006.
- [9] A. Harati Nejad Torbati, J. Picone, and M. Sobel, "A Left-to-Right HDP-HMM with HDPM Emissions," *Proceedings of the CISS*, 2014, pp. 1-6.
- [10] http://www.isip.piconepress.com/projects/npb_acoustic_modeling/downl oads/hdphmm/
- [11] Y.-W. Teh, "Dirichlet process," *Encyclopedia of Machine Learning*, Springer, 2010, pp. 280-287.
- [12] J. Sethuraman, "A constructive definition of Dirichlet priors," *Statistica Sinica*, vol. 4, no. 2, pp. 639-650, 1994.
- [13] C. E. Rasmussen, "The Infinite Gaussian Mixture Model," *Proceedings* of NIPS, 2000, pp. 554-560.
- [14] H. Ishwaran and M. Zarepour, "Exact and approximate sum representations for the Dirichlet process.," *Canadian Journal of Statistics*, vol. 30, no. 2, pp. 269-283, 2002.
- [15] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, 2011.
- [16] S. Young and P. C. Woodland, "State clustering in HMM-based continuous speech recognition," *Computer Speech & Language*, vol. 8, no. 4, pp. 369-383, 1994.



Fig. 7. Number of discovered Gaussians vs. amount of training data.

11

T-ASL-05266-2015

- [17] J. Pitman, Probability. New York, New York, USA: Springer-Verlag, 1993, pp. 480-498.
- [18] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, 2nd ed. Chapman & Hall, 2004.
- [19] P. Diaconis, K. Khare, and L. Saloff-Coste, "Gibbs Sampling, Conjugate Priors and Coupling," Sankhya A, vol. 72, no. 1, pp. 136-69, 2010.
- [20] F. A. Quintana and W. Tam, "Bayesian Estimation of Beta-binomial Models by Simulating Posterior Densities," *Journal of the Chilean Statistical Society*, vol. 13, no. 1-2, pp. 43-56, 1996.
- [21] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallet, N. Dahlgren, and V. Zue, "TIMIT Acoustic-Phonetic Continuous Speech Corpus," *The Linguistic Data Consortium Catalog*, 1993.
- [22] C. E. Rasmussen and Z. Ghahramani, "Occam's Razor," Proceedings of NIPS, 2001, pp. 294 -300.
- [23] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt, "Hidden Conditional Random Fields for Phone Classification," *Proceedings of INTERSPEECH*, 2005, pp. 1117-1120.
- [24] F. Sha and L. K. Saul, "Large Margin Gaussian Mixture Modeling for Phonetic Classification and Recognition," *Proceedings of ICASSP*, 2006, pp. 265-268.
- [25] P. Clarkson and P. J. Moreno, "On the use of support vector machines for phonetic classification," *Proceedings of ICASSP*, 1999, pp. 585-588.
- [26] S. Petrov, A. Pauls, and D. Klein, "Learning Structured Models for Phone Recognition," *Proceedings of EMNLP-CoNLL*, 2007, pp. 897-905.
- [27] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Transactions on ASSP*, vol. 37, no. 11, pp. 1641-1648, 1989.
- [28] L. Lamel and J.-L. Gauvain, "High Performance Speaker-Independent Phone Recognition Using CDHMM," *Proceedings of EUROSPEECH*, 1993, pp. 121-124.
- [29] S. Kapadia, V. Valtchev, and S. Young, "MMI training for continuous phoneme recognition on the TIMIT database," *Proceedings of ICASSP*, 1993, pp. 491-494.
- [30] A.K. Halberstadt and J. R. Glass, "Heterogeneous acoustic measurements and multiple classifiers for speech recognition," *Proceedings of ICSLP*, 1998, pp. 995-998.
- [31] J. Morris and E. Fosler-Lussier, "Conditional Random Fields for Integrating Local Discriminative Classifiers," *IEEE Transactions on* ASSP, vol. 16, no. 3, pp. 617-628, 2008.
- [32] D. Palaz, R. Collobert, and M. Magimai-Doss, "End-to-end Phoneme Sequence Recognition using Convolutional Neural Networks," *Proceedings of the NIPS Deep Learning Workshop*, 2013, pp. 1-8.
- [33] S. N. MacEachern, "Dependent Nonparametric Processes," in ASA Proceedings of the Section on Bayesian Statistical Science, 1999, pp. 50-55.
- [34] G. Tomlinson and M. Escobar, "Analysis of Densities," University of Toronto, Toronto, Canada, 1999.



Amir Harati received a PhD in electrical and computer engineering from Temple University in 2015. His primary research interests include speech recognition, statistical modeling and applications of machine learning in different domains. For the past few years Mr. Harati has been involved with various projects including nonparametric Bayesian modeling, automatic interpretation of EEG signals and speech

recognition. He currently works as a speech recognition engineer at Jibo Inc.



Joseph Picone received a PhD in electrical engineering from Illinois Institute of Technology in 1983. He is currently a professor in the Department of Electrical and Computer Engineering at Temple University where his primary research interests are machine learning approaches to acoustic modeling in speech recognition. His research group is known for producing many innovative open source materials for signal processing including a public domain speech recognition system (see www.isip.piconepress.com).

He has also spent significant portions of his career in academia, research and the government. Dr. Picone is a Senior Member of the IEEE and has been active in several professional societies related to human language technology. He has authored numerous papers on the subject and holds several patents in this field.