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 and eliminate 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 (DPM) 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 non-emitting 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, something that is also important for an application 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.

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

- Corresponding Author: Amir Harati, Department of Electrical and Computer Engineering, Temple University, 1949 North 12th Street, Philadelphia, Pennsylvania, USA 19122 (Tel: 215-500-1255; Email: amir.harati@gmail.com.
- Joseph Picone is with the Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 19122. Email: joseph.picone@gmail.com.

1 INTRODUCTION

Hidden Markov models (HMMs) [1] are one of the most successful models for sequential data 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 mixture components. However, these 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 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 a different formulation, 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 [7].

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 [6]. 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). If the data has finite length, the beginning and end of a sequence is typically modeled as two additional discrete events – non-emitting initial and final states [7]. 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 with two parallel hierarchies that enables sharing of data among different states. We refer to this model as a Doubly Hierarchical Dirichlet Process Hidden Markov Model (DHDPHMM) [8]. In this paper, we introduce a general method to add non-emitting states to both HDPHMMs and DHDPHMMs. We also develop a framework to learn non-ergodic structures from the data and present comprehensive experimental results for a standard phoneme recognition task in speech processing.

The paper is organized as follows. In Section 2, we provide background on nonparametric Bayesian modeling and formally introduce the HDPHMM model. In Sections 3 and 4, we introduce the DHDPHMM model and its extensions for non-ergodic modeling and estimation of non-emitting states. In Section 5, we present results on three tasks: a pilot study on simulated data, phoneme classification and recognition on speech data. We compare DHDPHMM with both baseline and state of the art systems.

2 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. Since such conditions are extremely common in applications like speech recognition, an overarching goal of this work is to improve performance when channel conditions are mismatched.

2.1 Nonparametric Bayesian Models

A Dirichlet process (DP) [9] is a discrete distribution that consists of a countably infinite number of probability masses and defines a distribution over discrete distributions with infinite support. A DP is denoted by $DP(\alpha, G_0)$, and is defined as [10]:

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

$$\beta_k = v_k \prod_{l=1}^{k-1} (1 - v_l), \qquad v_k \mid \alpha, G_0 \sim Beta(1, \alpha). \tag{2}$$

where G_0 represents the mean of the distribution [9], δ_{θ_k} is the unit impulse function at θ_k, β_k are weights sampled according to (2) [10], and α is a concentration parameter that represents the degree of concentration around the mean (α is inversely proportional to 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 on the components of a mixture model. The resulting model is referred to as a Dirichlet Process Mixture (DPM) model and is defined as [4]:

$$\begin{aligned} &\pi \mid \alpha \sim GEM(\alpha) \\ &z_i \mid \pi \sim Mult(\pi) \\ &\theta_k \mid G_0 \sim G_0 \\ &x_i \mid z_i, \{\theta_k\} \sim F(\theta_{z_i}) . \end{aligned} \tag{3}$$

In this model, the 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, which is the nonparametric counterpart of a GMM [11].

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 [4]:

$$\begin{aligned} G_0 \mid \gamma, H &\sim DP(\gamma, H) \\ G_j \mid \alpha, G_0 &\sim DP(\alpha, G_0) \\ \theta_{ji} \mid G_j &\sim G_j \\ x_{ji} \mid \theta_{ji} &\sim F\left(\theta_{ji}\right) \qquad for \quad j \in J \ . \end{aligned} \tag{4}$$

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.

2.2 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 with z_t, the Markovian structure can be represented by $z_t \mid z_{t-1} \sim \pi_{z_{t-1}}$, where $\pi_{z_{t-1}}$ is the multinomial distribution that represents the transition from state t-1 to state t. Observations are conditionally independent given the state of the HMM and are denoted by $x_t \mid z_t \sim F\left(\theta_{z_t}\right)$. 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:

$$
\begin{aligned}
&\beta \mid \gamma \sim GEM(\gamma) \\
&\pi_j \mid \alpha, \beta \sim DP(\alpha + \kappa, \frac{\alpha\beta + \kappa\delta_j}{\alpha + \kappa}) \\
&\psi_j \mid \sigma \sim GEM(\sigma) \\
&\theta_{kj}^{**} \mid H, \lambda \sim H(\lambda) \\
&z_t \mid z_{t-1}, \{\pi_j\}_{j=1}^{\infty} \sim \pi_{z_{t-1}} \\
&s_t \mid \{\psi_j\}_{j=1}^{\infty}, z_t \sim \psi_{z_t} \\
&x_t \mid \{\theta_{kj}^{**}\}_{k,j=1}^{\infty}, z_t \sim F\left(\theta_{z_t s_t}\right) .
\end{aligned}
\tag{5}
$$

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 state 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 $z_{1:T}$ 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 the HMM to improve the convergence speed of the

inference algorithm. However this algorithm requires approximation of the theoretically infinite distributions with a "degree L weak limit" approximation that truncates a DP into a Dirichlet distribution with L dimensions [12]:

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

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. Interested readers can refer to [5] for more details about this algorithm.

3 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 a DPM model:

$$\begin{aligned}
&\psi_j \mid \sigma \sim GEM(\sigma) \\
&s_t \mid \{\psi_j\}_{j=1}^{\infty}, z_t \sim \psi_{z_t} \\
&\theta_{kj}^{**} \mid H, \lambda \sim H(\lambda) \\
&x_t \mid \{\theta_{kj}^{**}\}_{k,j=1}^{\infty}, z_t \sim F\left(\theta_{z_t s_t}\right).
\end{aligned} \tag{7}$$

Equation (7) demonstrates 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 independent of other distributions. It has been shown previously [13] that sharing data points, if done properly, can improve the accuracy of the model.

As we have discussed in Section 2.1, HDP is the extension of a DPM to mixture modeling of grouped data. If the state assignment, z_t, 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:

$$
\begin{aligned}
&\xi \mid \tau \sim GEM(\tau) \\
&\psi_j \mid \sigma, \xi \sim DP(\sigma, \xi) \\
&\theta_{kj}^{**} \mid H, \lambda \sim H(\lambda) \\
&s_t \mid \left\{\psi_j\right\}_{j=1}^{\infty}, z_t \sim \psi_{z_t} \\
&x_t \mid \left\{\theta_{kj}^{**}\right\}_{k,j=1}^{\infty}, z_t \sim F\left(\theta_{z_t s_t}\right),
\end{aligned}
\tag{8}
$$

here ζ 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:

$$
\begin{aligned}
&\beta \mid \gamma \sim GEM(\gamma) \\
&\pi_j \mid \alpha, \beta \sim DP(\alpha + \kappa, \frac{\alpha\beta + \kappa\delta_j}{\alpha + \kappa}) \\
&\xi \mid \tau \sim GEM(\tau) \\
&\psi_j \mid \sigma, \xi \sim DP(\sigma, \xi) \\
&\theta_{kj}^{**} \mid H, \lambda \sim H(\lambda) \\
&z_t \mid z_{t-1}, \left\{\pi_j\right\}_{j=1}^{\infty} \sim \pi_{z_{t-1}} \\
&s_t \mid \left\{\psi_j\right\}_{j=1}^{\infty}, z_t \sim \psi_{z_t} \\
&x_t \mid \left\{\theta_{kj}^{**}\right\}_{k,j=1}^{\infty}, z_t \sim F\left(\theta_{z_t s_t}\right) .
\end{aligned}
\tag{9}
$$

DHDPHMM pools the data points while HDPHMM divides data points between different states. If we don't have enough data points in a particular state or a mixture component then the distribution parameters will be estimated poorly (e.g. the mean and covariance). 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.

3.1 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. Here we present modifications of this block sampler for inference for our DHDPHMM.

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):

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

$$\psi_j \mid \xi,\tau \sim Dir\left(\tau\xi_1,...,\tau\xi_{L'}\right). \tag{11}$$

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

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

$$\psi_j \mid \sigma,\xi,Z_{1:T},S_{1:T} \sim Dir\left(\sigma\xi_1+n'_{j1},...,\sigma\xi_{L'}+n'_{jL'}\right) \tag{13}$$

where M'_{jk} is the number of clusters in state j with mixture component k; $M'_{\cdot k}$ 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), can be written as:

$$P\left(\tau \mid n_{1\cdot},...,n_{J\cdot},M'_{1\cdot},...M'_{J\cdot}\right) \propto Gamma\left(a+M'_{\cdot\cdot}-\sum_{j=1}^{L'}s_j,b-\sum_{j=1}^{L'}\log r_j\right) \tag{14}$$

$$P\left(r_j \mid \tau,r_{\setminus j},s,n_{1\cdot},...,n_{J\cdot},M'_{1\cdot},...M'_{J\cdot}\right) \propto Beta\left(\tau+1,n_{j\cdot}\right) \tag{15}$$

$$P\left(s_j \mid \tau,s_{\setminus j},r,n_{1\cdot},...,n_{J\cdot},M'_{1\cdot},...M'_{J\cdot}\right) \propto Ber\left(\frac{n_{j\cdot}}{n_{j\cdot}+\tau}\right) \tag{16}$$

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.

3.2 Scalability

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. The HDPHMM model has LxL' Gaussians to estimate. Since up to 95% of the inference time is spent in calculating the likelihood of data for Gaussian distributions, a reduction from LxL' to L' reduces the computational time considerably. Also we have utilized parallel programming facilities (e.g. openMP) for the implementation of both algorithms, which makes this process feasible for large data sets. Fig. 1 provides a comparison of both algorithms for different values of L and L'. DHDPHMM's computational complexity is flat as the maximum bound on the number of states increases while the inference cost for HDPHMM grows linearly.

4 DHDPHMM WITH A NON-ERGODIC STRUCTURE

A non-ergodic structure for the DHDPHMM can be achieved by modifying the transition distributions. These modifications can also be applied to HDPHMM using a similar approach.

4.1 Left-to-Right DHDPHMM

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

$$\pi_j \mid \alpha, \beta \sim DP(\alpha + \kappa, \frac{\alpha\beta + \kappa\delta_j}{\alpha + \kappa}). \tag{17}$$

From (17) we can see 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 $V_j=\{V_{ji}\}$:

$$V_{ji} = \begin{cases} 0, & i < j \\ 1, & i \geq j \end{cases} \tag{18}$$

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

$$\beta' = \frac{\beta \bullet V_j}{\sum_i \beta_i V_{ji}}. \tag{19}$$

In the block sampler algorithm, we have:

$$\pi_j \sim Dir\left(\alpha\beta'_1 + n_{j1}, ..., \alpha\beta'_j + \kappa + n_{jj}, ..., \alpha\beta'_L + n_{jL}\right), j = 1, ..., L \tag{20}$$

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. If we leave π_j as in (20) the resulting model would be an LR model with possible loops. The model would be biased toward an LR structure but with the possibility of forming 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 :

$$\pi_j \sim Dir\left(\alpha\beta'_1 + V_{j1}n_{j1}, ..., \alpha\beta'_j + \kappa + V_{jj}n_{jj}, ..., \alpha\beta'_L + V_{jL}n_{jL}\right), \quad j = 1, ..., L. \tag{21}$$

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:

$$V_{ji} = \begin{cases} 0, & 0 < i < j \\ 1, & i \geq j, i = 0 \end{cases}. \tag{22}$$

β' can be calculated from (19) and π_j should be sampled from (21).

The LR models described above allow for skip transitions that mean the model learns parallel

paths that correspond 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):

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

4.2 Initial and Final Non-Emitting States

In many applications, such as speech recognition, an LR-HMM begins from and ends with non-emitting states. These states are required to model the beginning and end of finite duration sequences. 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 non-emitting state is more complicated. In the following sections we will discuss two approaches that solve this problem.

4.2.1 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 tails triggers a transition to the final state. This can be modeled using a geometric distribution [14]:

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

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

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

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 [14] that the maximum likelihood estimation is obtained by:

$$\widehat{\rho_i} = \frac{M_i}{\sum_{j \in S_M} k_j} \tag{26}$$

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 $P_3 = 0$.

4.2.2 Bayesian Estimation

Another approach to estimate transitions to a final non-emitting 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 with hyperparameters (a,b) as the prior and obtain a posterior as [15], [16]:

$$\rho_i \sim Beta\left(a + M_i, b + \sum_{j \in S_M} \left(k_j - 1\right)\right) \tag{27}$$

where M_i and S_M are the number of times which state z_i was the last state and set of all such states respectively. Hyperparameters (a,b) can also be estimated using a Gibbs sampler if required [17]. 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 one. 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 $1\text{-}\rho_i$ portion as before.

4.3 An Integrated Model

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

$$
\begin{aligned}
&\beta \mid \gamma \sim GEM(\gamma), \beta' = \frac{V_j \cdot \beta}{\sum_i V_{ji} \beta_i} \\
&\pi_j \mid \alpha, \beta' \sim DP(\alpha + \kappa, \frac{\alpha\beta' + \kappa\delta_j}{\alpha + \kappa}) \\
&\xi \mid \tau \sim GEM(\tau) \\
&\psi_j \mid \sigma, \xi \sim DP(\sigma, \xi) \\
&\theta_{kj}^{**} \mid H, \lambda \sim H(\lambda) \\
&z_t \mid z_{t-1}, \{\pi_j\}_{j=1}^{\infty} \sim \pi_{z_{t-1}} \\
&s_t \mid \{\psi_j\}_{j=1}^{\infty}, z_t \sim \psi_{z_t} \\
&x_t \mid \{\theta_{kj}^{**}\}_{k,j=1}^{\infty}, z_t \sim F\left(\theta_{z_t s_t}\right) .
\end{aligned}
\tag{28}
$$

In this definition V_i should be replaced with the proper definition from previous section based on the type of structure we want. For example if we want an LR model then V_i should be sampled from (18). Also note that by setting V_i to one we obtain the ergodic DHDPHMM in (9). A graphical representation is shown in Fig. 3-b. The HDPHMM [5] is also displayed in Fig. 3-a for comparison.

We have not incorporated modeling of non-emitting states discussed above in (28). If we choose to use a maximum likelihood approach for estimating the non-emitting states then no change to this model is 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_j in (28) with:

$$
\begin{aligned}
&\overline{w}, \overline{\chi} \sim Modified \quad stick-breaking\left(\alpha, \kappa\right) \\
&\pi_j \mid \overline{w}, \overline{\chi} \sim \sum_k w_k \delta_{\chi_k}
\end{aligned}
\tag{29}
$$

$$
Modified \quad stick-breaking\left(\alpha, \kappa\right) = \begin{cases} for \quad i = \{1, 2, ...\}: \\ v_i \mid \alpha, \kappa \sim Beta(1, \alpha + \kappa) \\ w_i \mid v_i, \rho_j = v_i (1 - \rho_j) \prod_{l=1}^{k-1} (1 - v_l) \\ \chi_i \mid \alpha, \beta', \kappa \sim \sum_k \frac{\alpha\beta_k' + \kappa\delta_{kj}}{\alpha + \kappa} \delta_k \end{cases}
\tag{30}
$$

where we have replaced DP with the modified stick-breaking process described above.

5 EXPERIMENTS

In this section we provide some experimental results which compare DHDPHMM with HDPHMM, HMM and several other state of the art models. The experiments begin with artificial data and then proceed to standard phoneme classification and recognition tasks.

5.1 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 1900 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.

Fig. 4-a shows the average likelihood for different models for held-out data by averaging five independent chains. Fig. 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 because 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 the structure close to the correct one, its likelihood is slightly lower than the ergodic HDPHMM. However, LR DHDPHMM produces a 15% (relative) improvement in likelihoods compared to the ergodic model. It is also interesting to note that the likelihoods of models discovered by all the nonparametric Bayesian algorithms are superior to the likelihood of the reference model itself.

5.2 Phoneme Classification on the TIMIT Corpus

The TIMIT Corpus [18] 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 [20].

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 [19] was also used.

5.2.1 A Comparison to HDPHMM

In Table 1 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 an LR model with two other models: a strictly LR topology and an ergodic model. As this table shows DHDPHMM is consistently better than their HDPHMM counterparts. Further, it can be seen that LR models perform better than ergodic models (as expected) while strictly LR models perform more poorly. This is due to the fact that a strictly LR model constrains the best path to one path while the other LR models learn many parallel paths. From the last column of this table we can see LR DHDPHMM finds 3888 Gaussians for all 48 phonemes while two different LR HDPHMM models find 4628 and 7281 Gaussians for all phonemes respectively. These numbers show DHDPHMM can learn a less complex model that can explain the data better than a more complex model learned by HDPHMM. This is an important property that validates the basic

philosophy of the NPBM and also follows Occam's Razor [20].

Fig. 5 shows the structures for phonemes /aa/ and /sh/ discovered by DHDPHMM. 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. This is shown in Fig. 5-b where S1-S2, S1-S3 and S1-S4 depict three parallel models.

Fig. 6 show the confusion matrix for the most confusable pairs of this classification task. The general confusion matrix follows the same trend but because it is too large it has not been shown in this paper. From this confusion matrix we can see that most errors occur, as expected, between acoustically similar phonemes. In fact, if we use 5 broad phonetic classes instead of using 39 phoneme classes, the classification error rate drops to 4.8%.

5.2.2 A Comparison to Other Representative Systems

Table 2 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 other two and has an error rate of 26.1% on the core subset. Our LR DHDPHMM model has 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 5760 (set a priori) while our LR DHDPHMM uses only 3888 Gaussians. Fig. 7 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 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.

Fig. 8 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 other type of priors.

The fourth row of Table 2 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 our ML 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 [24], unlike DHDPHMM, models the context implicitly, which seems to be one of the main reasons that it performs so well. We expect to obtain better results if we also use context dependent (CD) models instead of context independent (CI) models.

5.3 Supervised Phoneme Recognition

Speech recognition systems usually use a semi-supervised method to train acoustic models. By semi-supervised we mean the exact boundaries between phonemes are not given but instead the transcription only consists of a sequence of phones in an utterance. It is has been shown that this semi-supervised method actually works better than a completely supervised method [24]. However, in this section we use a completely supervised method to evaluate DHDPHMM models for a phoneme recognition task. As in the previous section, we have trained DHDPHMMs only using maximum likelihood and with no 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 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 *% Correct* and *% Error* as follows [19]:

$$\%\,Correct = \frac{N - S - D}{N} \tag{31}$$

$$\%\,Error = \frac{S + D + I}{N} \tag{32}$$

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 3 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 HMM based systems with and without contextual information. We can see the error rate drops from 35.4% to 26.2% when we use a system with contextual modeling. We can also see DHDPHMM works much better than a comparable CI HMM model (the error rate drops from 35.4% 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 CD model

of row four (CD HMM 3). We expect to obtain much better results if we use context dependent models. Our model also performs better than a discriminatively trained context-independent HMM. By comparing DHDPHMM with other systems presented in Table 3 we can see DHDPHMM is among the best models for context-independent systems but is not as good as state of the art context-dependent models.

6 CONCLUSIONS

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 we have only used ML training for DHDPHMM performance is comparable to discriminatively trained models. Further, DHDPHMM provides the best performance among context-independent 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 [31] or an Analysis of Density (AnDe) model [32]. It has been shown that the AnDe model is the appropriate model for problems involving sharing among multiple sets of density estimators [4], [20].

ACKNOWLEDGEMENTS

The authors wish to thank Professor Marc Sobel for many valuable discussions on these topics.

This research was supported in part by the National Science Foundation through Major Research Instrumentation Grant No. CNS-09-58854.

REFERENCES

1. 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. 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. B.-H. Juang and L. Rabiner, "Hidden Markov Models for Speech Recognition," *Technometrics*, vol. 33, no. 3, pp. 251-272, 1991.

7. G. A. Fink, "Configuration of Hidden Markov Models From Theory to Applications," *Markov Models for Pattern Recognition*, Springer Berlin Heidelberg, 2008, pp. 127-136.

8. 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.

9. Y.-W. Teh, "Dirichlet process," *Encyclopedia of Machine Learning*, Springer, 2010, pp. 280-287.

10. J. Sethuraman, "A constructive definition of Dirichlet priors," *Statistica Sinica*, vol. 4, no. 2, pp. 639-650, 1994.

11. C. E. Rasmussen, "The Infinite Gaussian Mixture Model," *Proceedings of NIPS*, 2000, pp. 554-560.

12. 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.

13. 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.

14. J. Pitman, *Probability*. New York, New York, USA: Springer-Verlag, 1993, pp. 480-498.

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

16. P. Diaconis, K. Khare, and L. Saloff-Coste, "Gibbs Sampling, Conjugate Priors and Coupling," *Sankhya A*, vol. 72, no. 1, pp. 136-69, 2010.

17. 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.

18. 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. [Online]. Available: http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1

19. 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.

20. C. E. Rasmussen and Z. Ghahramani, "Occam's Razor," *Proceedings of NIPS*, 2001, pp. 294 -300.

21. A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt, "Hidden Conditional Random Fields for Phone Classification," *Proceedings of INTERSPEECH*, 2005, pp. 1117-1120.

22. F. Sha and L. K. Saul, "Large Margin Gaussian Mixture Modeling for Phonetic Classification and Recognition," *Proceedings of ICASSP,* 2006, pp. 265-268.

23. P. Clarkson and P. J. Moreno, "On the use of support vector machines for phonetic classification," *Proceedings of ICASSP*, 1999, pp. 585-588.

24. S. Petrov, A. Pauls, and D. Klein, "Learning Structured Models for Phone Recognition," *Proceedings of EMNLP-CoNLL*, 2007, pp. 897-905.

25. 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.

26. L. Lamel and J.-L. Gauvain, "High Performance Speaker-Independent Phone Recognition Using CDHMM," *Proceedings of EUROSPEECH*, 1993, pp. 121-124.

27. S. Kapadia, V. Valtchev, and S. Young, "MMI training for continuous phoneme recognition on the TIMIT database," *Proceedings of ICASSP*, 1993, pp. 491-494.

28. A.K. Halberstadt and J. R. Glass, “Heterogeneous acoustic measurements and multiple classifiers for speech recognition,” *Proceedings of ICSLP*, 1998, pp. 995-998.

29. 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.

30. 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.

31. S. N. MacEachern, “Dependent Nonparametric Processes,” in *ASA Proceedings of the Section on Bayesian Statistical Science*, 1999, pp. 50-55.

32. G. Tomlinson and M. Escobar, “Analysis of Densities,” University of Toronto, Toronto, Canada, 1999.

List of Figures:

Fig. 1. DHDPHMM improves scalability relative to HDPHMM

Fig. 2. Outgoing probabilities for state z_i

Fig. 3. Comparison of models: (a) ergodic HDPHMM [5] (b) DHDPHMM

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

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

Fig. 6. Confusion matrix for phoneme classification for the most confusable pairs

Fig. 7. Error rate vs. amount of training data for LR DHDPHMM and LR HMM

Fig. 8. Number of discovered Gaussians vs. amount of training data

List of Tables:

TABLE 1 Comparison of LR DHDPHMM with HDPHMM

TABLE 2 Comparison of LR DHDPHMM to other algorithms

TABLE 3 Comparison of phoneme recognition performance

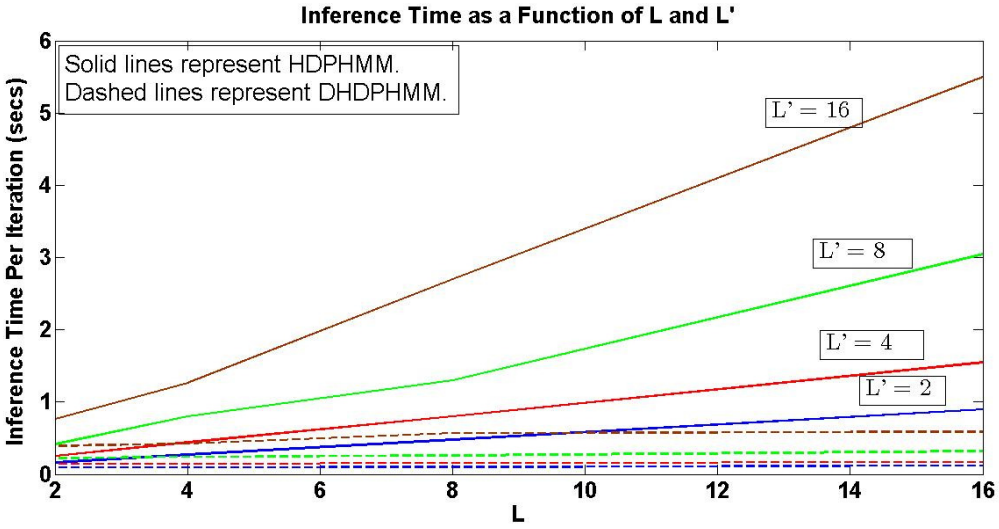

Fig. 1. DHDPHMM improves scalability relative to HDPHMM

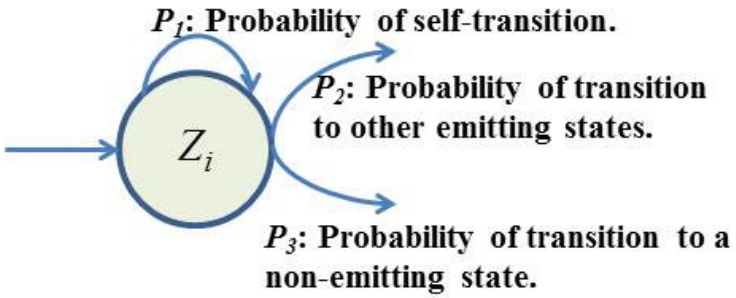

Fig. 2. Outgoing probabilities for state zi

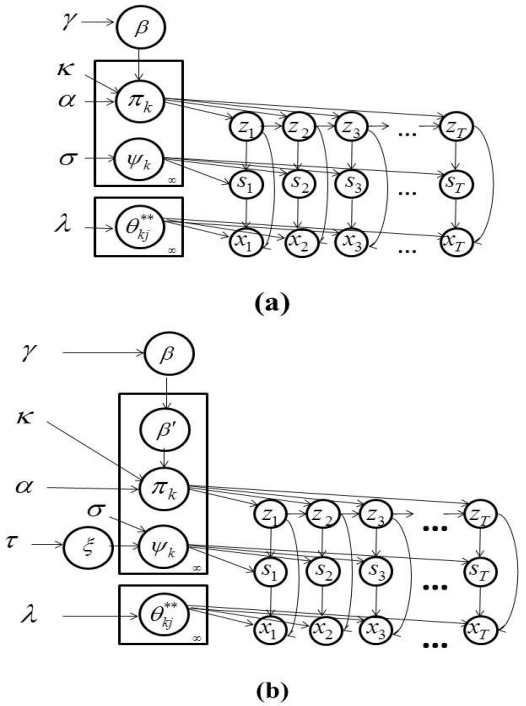

Fig. 3. Comparison of models: (a) ergodic HDPHMM [5] (b) DHDPHMM

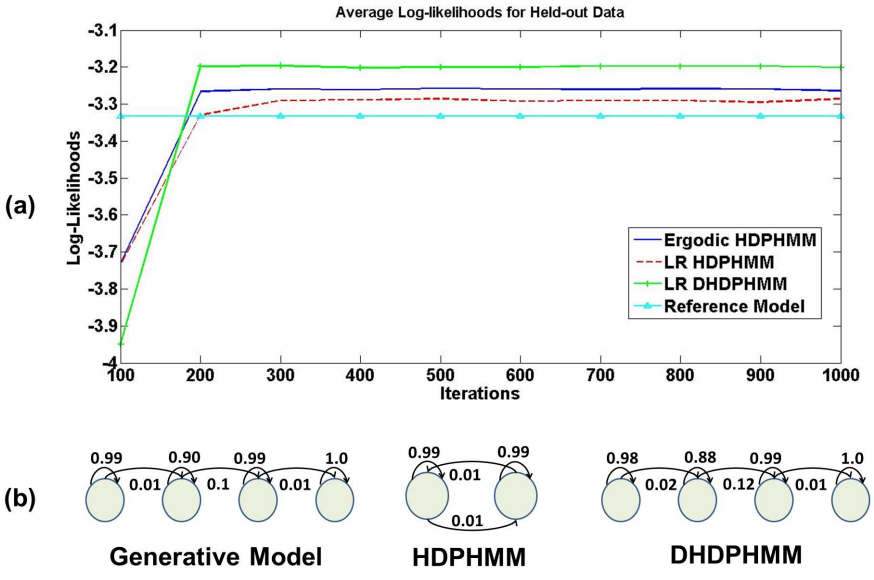

Fig. 4. Comparison of (a) log-likelihoods of the proposed models to an ergodic model, and (b) the corresponding model structures

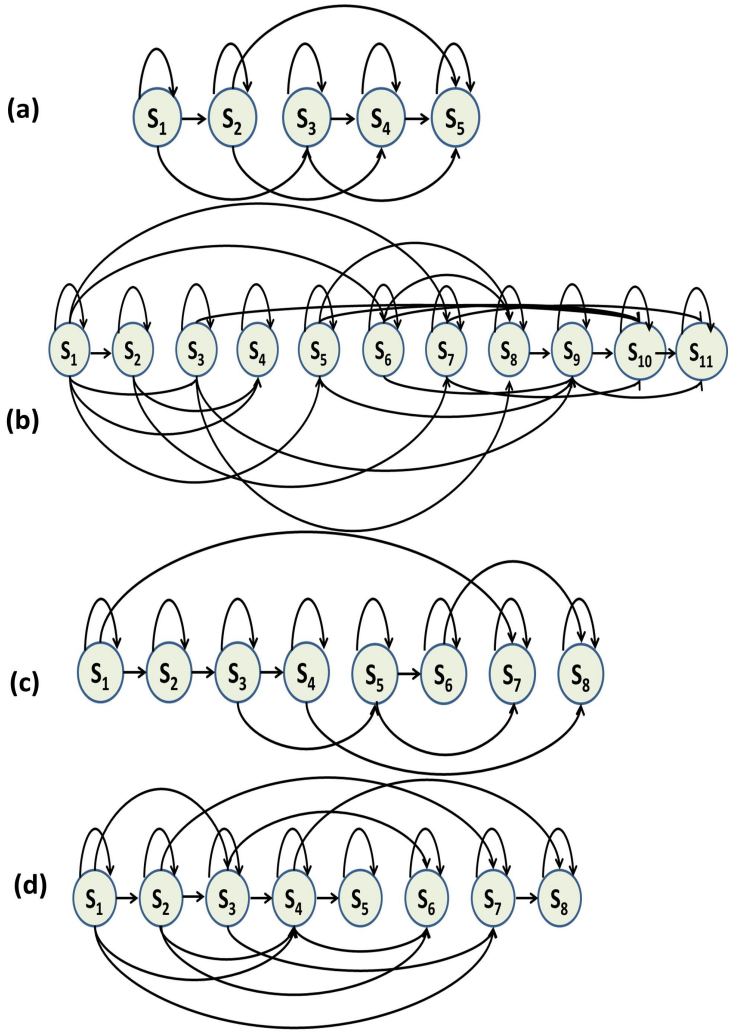

Fig. 5. An automatically derived model structure for a left-to-right DHDPHMM model (without the first and last non-emitting states) for (a) /aa/ with 175 examples (b) /aa/ with 2,256 examples (c) /sh/ with 100 examples and (d) /sh/ with 1,317 examples

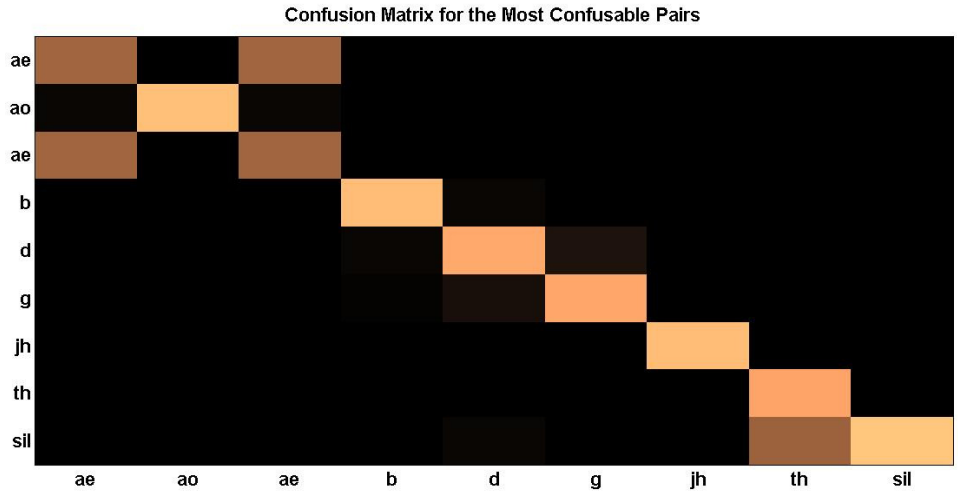

Fig. 6. Confusion matrix for phoneme classification for the most confusable pairs

Fig. 7. Error rate vs. amount of training data for LR DHDPHMM and LR HMM

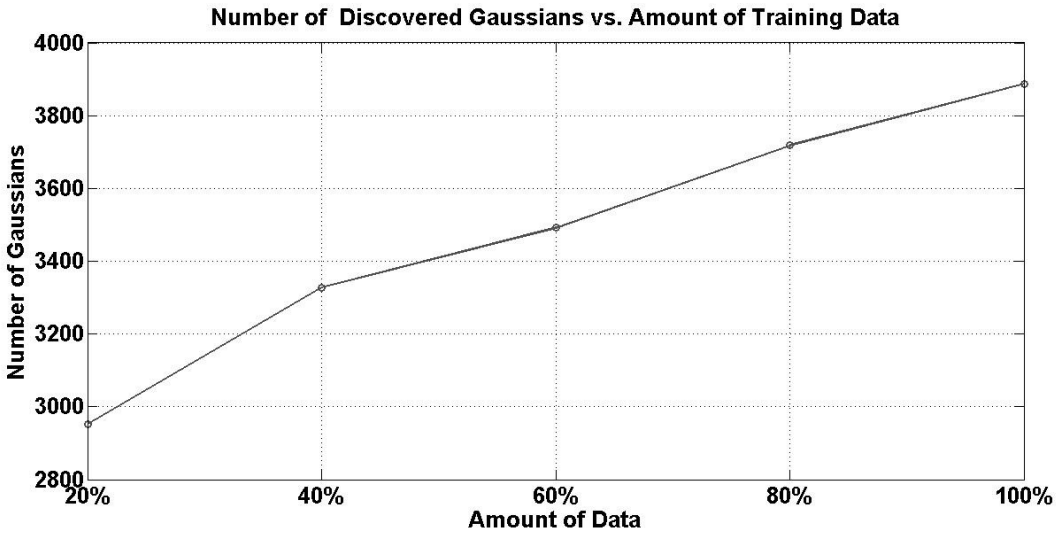

Fig. 8. Number of discovered Gaussians vs. amount of training data

TABLE 1
COMPARISON OF LR DHDPHMM WITH HDPHMM

Model	Dev Set (% Error)	Core Set (% Error)	No. Gauss.
LR HDPHMM 1	23.5%	24.4%	4628
LR HDPHMM 2	23.8%	25.1%	7281
Ergodic DHDPHMM	24.0%	25.4%	2704
Strictly LR DHDPHMM	39.0%	38.4%	2550
LR DHDPHMM	**20.5%**	**21.4%**	**3888**

TABLE 2
COMPARISON OF PHONEME RECOGNITION PERFORMANCE

Model	Discrim. Training	Dev Set (% Error)	Core Set (% Error)
HMM (10 Gauss.)	No	28.4%	28.7%
HMM (20 Gauss.)	No	26.1%	27.3%
HMM (40 Gauss.)	No	25.0%	26.1%
HMM/MMI (20 Gauss.) [20]	Yes	23.2%	24.6%
HCRF/SGD [20]	Yes	20.3%	21.7%
Large Margin GMMs [22]	Yes	–	21.1%
GMMs/Full Cov. [22]	No	–	26.0%
SVM [23]	Yes	–	22.4%
Data-driven HMM [24]	No	–	21.4%
LR DHDPHMM	**No**	**20.5%**	**21.4%**

TABLE 3
COMPARISON OF PHONEME RECOGNITION PERFORMANCE

Model	Discrim. Training	Context Modeling	% Error	% Correct	Subset
CI-HMM [25]	No	No	35.9%	–	TID7
CD-HMM 1[25]	No	Yes	26.2%	–	TID7
CD-HMM 2[26]	No	Yes	30.9%	–	Core
CD-HMM 3[13]	No	Yes	27.7%	–	Core
HMM MMI 1 [27]	Yes	No	32.5%	73.5%	Random
HMM MMI 2 / Full Cov. [27]	Yes	No	30.3%	74.4%	Random
Heterogeneous Class. [28]	Yes	Yes	24.4%	–	Core
Data-driven HMM [24]	N/A	Yes	26.4%	–	Core
Large Margin GMM [22]	Yes	No	30.1%	–	Core
CRF [29]	Yes	No	29.9%	73.2%	All
Tandem HMM [29]	Yes	Yes	30.6%	75.6%	All
CNN/CRF [30]	Yes	No	29.9%	–	Core
LR DHDPHMM	**No**	**No**	**29.7%**	**74.1%**	**Core**
LR DHDPHMM	**No**	**No**	**28.6%**	**75.1%**	**Dev**
LR DHDPHMM	**No**	**No**	**29.2%**	**74.7%**	**All**