Multi-Task Learning for Sequential Data via iHMMs and the Nested Dirichlet Process

Kai Ni KN6@EE.DUKE.EDU
Lawrence Carin LCARIN@EE.DUKE.EDU
Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA

David Dunson DUNSON@STAT.DUKE.EDU
Biostatistics Branch, National Institute of Environmental Health Sciences, RTP, NC 27709 USA

Abstract

A new hierarchical nonparametric Bayesian model is proposed for the problem of multi-task learning (MTL) with sequential data. Sequential data are typically modeled with a hidden Markov model (HMM), for which one often must choose an appropriate model structure (number of states) before learning. Here we model sequential data from each task with an infinite hidden Markov model (iHMM), avoiding the problem of model selection. The MTL for iHMMs is implemented by imposing a nested Dirichlet process (nDP) prior on the base distributions of the iHMMs. The nDP-iHMM MTL method allows us to perform task-level clustering and data-level clustering simultaneously, with which the learning for individual iHMMs is enhanced and between-task similarities are learned. Learning and inference for the nDP-iHMM MTL are based on a Gibbs sampler. The effectiveness of the framework is demonstrated using synthetic data as well as real music data.

1. Introduction

Multi-task learning (MTL) (Caruana, 1997) has attracted significant interest in the machine learning community (Blei et al., 2004; Rasmussen, 2000; Thurn & O'Sullivan, 1996; Xue et al., 2007) and has been successfully applied to information retrieval (Blei et al., 2004) and computer vision (Thurn & O'Sullivan, 1996). Recent research on MTL has exploited new ideas in Bayesian hierarchical modeling (Gelman et al., 1995). In MTL, data from multiple tasks are learned collectively and data are appropriately shared among related tasks. Therefore the training data for each task are strengthened and the overall learning performance is improved. This is especially useful when there are limited training data from each task.

While most work in hierarchical Bayesian modeling addresses clustering multiple sets of data that are exchangeable within tasks, little has been done to solve the MTL problem for sequential data. Hidden Markov models (HMMs) have been widely used to analyze sequential data from a single source, addressing problems in speech recognition (Rabiner, 1989), music analysis (Logan & Salomon, 2001) and multi-aspect target detection (Runkle et al., 1999). In many cases, one may have limited sequential data for training. Rather than building HMMs for each task separately, it is preferable to identify relationships between tasks and share information appropriately, thus obtaining more accurate task-dependent models.

In the context of sequential-data analysis using HMMs, a key issue is to develop a methodology for finding the appropriate model complexity, i.e., defining the appropriate number of states. However, the data may not be represented by a single "correct" HMM structure, i.e., a fixed number of states. Rather than performing model selection (Stolcke & Omohundro, 1993) to select a fixed model structure, we employ a nonparametric Bayesian approach developed by Teh et al. (2006) in which the number of states is not fixed *a priori*. To address the problem of appropriately sharing data between tasks we utilize the nested Dirichlet process (nDP) (Rodriguez et al., 2006).

The nDP-iHMM introduced here represents a new hierarchical nonparametric Bayesian model for multi-task learning of sequential data. The sequential data from each task is modeled with an infinite hidden Markov model (iHMM) and the iHMMs are shared

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

by imposing an nDP prior on the base distributions of the iHMMs, leading to a general Bayesian learning algorithm for simultaneous iHMM task-level clustering and data-level modeling.

2. The Infinite Hidden Markov Model

Hidden Markov models (HMMs) have been widely used for modeling sequential data. A data sequence of length T generated by an HMM yields a sequence of observations $\boldsymbol{o} = \{o_1, o_2, \ldots, o_T\}$ and a sequence of hidden underlying states $\boldsymbol{s} = \{s_1, s_2, \ldots, s_T\}$, the latter following a Markov process. Consider an HMM with M states and C possible observations (we focus on a discrete HMM here for simplicity, but generalization to continuous HMMs is straightforward). The parameters of the model are $(\pi^M, A^{M\times M}, B^{M\times C})$, with π being the initial-state probability, A the transition matrix of $P(s_t|s_{t-1})$ and B the observation matrix of $P(o_t|s_t)$.

The conventional inference methods for HMMs are the expectation-maximization (EM) method implemented via the Baum-Welch algorithm (Rabiner, 1989), and the variational Bayesian method (Beal, 2003). However, in both methods the model structure must be specified initially, i.e. the number of states is fixed. Knowing the correct model complexity requires expensive model selection and in some applications there may exist no such fixed "correct" model (the limited sequential data for a given problem may be best represented via an ensemble of HMMs, with different numbers of states). We address this problem by using an HMM with a countably infinite state space, namely the *infinite hidden Markov model* (iHMM). Beal et al. (2002) first proposed the iHMM and provided an approximate sampling scheme for inference. Teh et al. (2006) demonstrated that the HDP can be used to recast the iHMM and provided a useful sampling scheme. Below we overview the HDP, and then describe how the HDP may be employed to develop an iHMM.

2.1. Hierarchical Dirichlet Processes

The *hierarchical Dirichlet process* considers learning problems of multiple related groups of data, with each group described by an infinite mixture model, and the mixture components are shared across groups. Consider first a single group of observations $\{x_1, \cdots, x_N\}$, with each x_i generated from a distribution $x_i \sim F(\theta_i)$. The parameters θ_i are in turn drawn from an unknown mixture distribution G, which is assumed to be drawn from a *Dirichlet process* (Ferguson, 1973) $G \sim DP(\gamma, H)$, where γ is a positive real number and H is the base distribution for G. Such a model is known as a *Dirichlet process mixture model* (Escobar & West, 1995) with the number of mixture components unbounded and inferred automatically from the data. Ferguson (1973) showed that samples drawn from $DP(\gamma, H)$ are discrete with probability one, a property made explicit by the stick-breaking construction (Sethuraman, 1994)

$$G = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k^*} \quad \beta_k = \beta_k^{'} \prod_{l=1}^{k-1} (1-\beta_l^{'}) \quad \beta_k^{'} \overset{iid}{\sim} \text{Beta}(1, \gamma), \tag{1}$$

where $\delta_{\theta_k^*}$ is a discrete measure concentrated at θ_k^* and $\theta_k^* \overset{iid}{\sim} H$. A graphical representation of a DP mixture model is given in Fig. 1(a). Indicator variable z_i denotes the mixture component generating the data point $x_i \sim F(\theta_{z_i}^*)$, i.e., $\theta_i = \theta_{z_i}^*$. The sharing arises when several θ_i's use the same θ_k^*.

Now consider J groups of data, denoted $((x_{ji})_{i=1}^{N_j})_{j=1}^{J}$. To construct an HDP, a global probability measure $G_0 \sim DP(\gamma, H)$ is first drawn to define the base distribution for each data group, and then $G_j \sim DP(\alpha, G_0)$ is sampled independently for each group. The discreteness of G_0 (as shown in (1)) guarantees that the G_j's will reuse the same set of shared mixture components defined in G_0 but with different proportions (Teh et al., 2006):

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k^*} \quad G_j = \sum_{k=1}^{\infty} w_{jk} \delta_{\theta_k^*} \quad \boldsymbol{w}_j \sim \text{DP}(\alpha, \boldsymbol{\beta}), \tag{2}$$

where $\boldsymbol{w}_j$ is an infinite-dimensional vector of probabilities that sum to one almost surely. The HDP can be used to model J groups of coupled infinite mixture models. The graphical model of an HDP mixture is shown in Fig. 1(b), where datum x_{ji} in group j is generated by first drawing $\theta_{ji} \sim G_j$, then sampling $x_{ji} \sim F(\theta_{ji})$. Parameters $\{\theta_k^*\}_{k=1}^{\infty}$ are the set of shared mixture components drawn from G_0, and z_{ji} is the indicator variable for which $\theta_{ji} = \theta_{z_{ji}}^*$.

2.2. Learning an iHMM via HDP

An M-state HMM can be regarded as a set of M coupled finite mixture models (each with M shared mixture components). Given the hidden state (random variable) $s_{t-1} = j$, the conditional distribution of the next observation o_t is

$$p(o_t|s_{t-1} = j) = \sum_{i=1}^{M} a_{ji} b_i(o_t), \tag{3}$$

where $a_{ji} = p(s_t = i|s_{t-1} = j)$ is the probability of choosing the i^{th} state given s_{t-1} and $b_i(\cdot)$ is the i^{th}

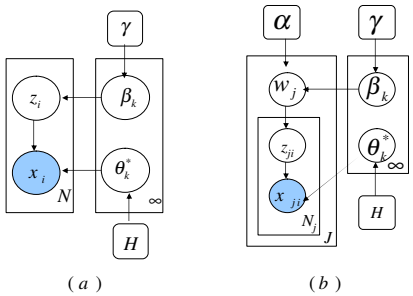

Figure 1. (a) Graphical representation of a Dirichlet process mixture model using (1). (b) Graphical representation of a hierarchical Dirichlet process (HDP) mixture model using (2).

state-dependent observation model. Thus the previous state s_{t-1} indexes a specific row of the transition matrix serving as the mixture weights for choosing current state s_t, and the state-dependent observation models serve as the mixture components generating o_t. Note that the HMM involves a set of mixture models, one for each possible visited state at time $t-1$. To deal with an infinite number of states, it is natural to apply a set of state-specific DPs, one for each value of the states. Furthermore, these DPs must be shared because they use the same set of states and observation models $\{b_i(\cdot)\}_{i=1}^{\infty}$. This is similar to the HDP mixture model but with a key distinction being that the datum (observation) belongs to a random group (indexed by the previous hidden state) in the iHMM rather than a fixed group in the HDP.

The HDP construction of the iHMM is shown in Fig. 2, with parameters defined as

$$\begin{aligned}
&\boldsymbol{o}_t \mid s_t, \{\theta_k^*\}_{k=1}^{\infty} \sim F(\theta_{s_t}^*) \qquad \{\theta_k^*\}_{k=1}^{\infty} \mid H \sim H \\
&s_t \mid s_{t-1}, \{\boldsymbol{w}_n\}_{n=1}^{\infty} \sim \text{Mult}(\boldsymbol{w}_{s_{t-1}}) \\
&\{\boldsymbol{w}_n\}_{n=1}^{\infty} \mid \alpha, \boldsymbol{\beta} \sim \text{DP}(\alpha, \boldsymbol{\beta}) \quad \boldsymbol{\beta} \mid \gamma \sim \text{Stick}(\gamma), \qquad (4)
\end{aligned}$$

where $\boldsymbol{w}_n$ corresponds to the row of transition matrix A, $F(\theta_k^*)$ corresponds to the observation model $b_k(\cdot)$ and Stick$(\cdot)$ represents the stick-breaking weights in (1). Each observation is represented with an L-dimensional feature vector $\boldsymbol{o}_t = [o_t^1, \cdots, o_t^L]$ and the feature vector is assumed to be generated from a signal model F.

A single HDP is used to construct an iHMM for a single task (one type of sequential data). If there are multiple tasks, it has been suggested (Teh et al., 2006) that one could put an additional level of the Bayesian hierarchy, letting a master Dirichlet process couple each of the iHMMs. While such a framework will allow sharing of the parameters between the tasks, it does not explicitly address the inter-relationships between the tasks, with this the motivation of the nDP.

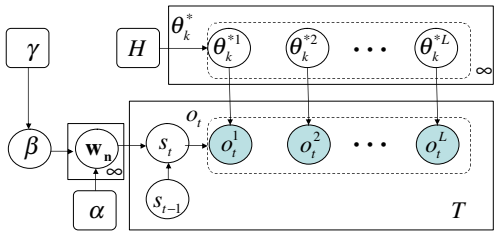

Figure 2. The infinite hidden Markov model interpreted as an HDP. For observation $\boldsymbol{o}_t$, s_{t-1} defines the mixture model to be used and s_t selects the mixture component according to infinite dimension weight vector $\boldsymbol{w}_{s_{t-1}}$.

3. Multi-Task Learning with Infinite Hidden Markov Models

Consider a learning problem in which we have sequential data collected from J different but possibly related tasks, $\mathcal{D} = \{\boldsymbol{O}_1, \ldots, \boldsymbol{O}_J\}$, where $\boldsymbol{O}_j = \{o_{j1}, \ldots, o_{jT}\}$ is the sequential data from task j. For example, each $\boldsymbol{O}_j$ may represent the observation sequence of features extracted from the j^{th} music clip. We here assume a single sequence with fixed length T for each task (music piece), but this is easily generalized to multiple sequences with different observation lengths.

Our goal is to build accurate HMMs for each of the tasks and also learn which tasks are similar (for example, to learn which pieces of music are similar). A naive way to learn the HMMs is to treat each task separately. However since these tasks may be related, and in some applications the sequential data are limited, the data from one task may potentially be useful to help build the model for other tasks. On the other hand, each task will likely have its own characteristics, thus simply pooling them and learning a single HMM is also inappropriate. We wish to exploit the sharing structure between tasks appropriately and use the shared information to enhance model learning. Furthermore, we wish to do this in a setting for which the problem of model selection (number of HMM states) is avoided. Finally, learning the appropriate inter-task sharing mechanisms is also of interest, because it gives insight into the relationships between the sequential tasks, with this important for information retrieval; for example, one may wish to learn which musical pieces are similar to one another.

We propose a new hierarchical nonparametric Bayesian model for sequential-data MTL. At the bottom level each task is modeled with an iHMM as described in Sec. 2.2, and at the top level the data in the tasks are shared appropriately by imposing a nested

Dirichlet process (nDP) prior on the base distributions of the iHMMs, yielding the nDP-iHMM. There are several appealing properties of the proposed model: (i) the problem of selecting an appropriate number of HMM states is avoided; (ii) the similarity measurement between tasks is obtained directly; (iii) the state-dependent observation models are shared among related tasks and model learning is improved; and (iv) upon sharing, each task still maintains its own transition matrix, which can be used to distinguish the tasks.

3.1. The Nested Dirichlet Process

The nested Dirichlet process (nDP) has been proposed by Rodriguez et al. (2006) to perform intra-task and inter-task clustering simultaneously (for the work presented here, the intra-task clustering is associated with learning the iHMM state structure). Suppose x_{ji}, for $i = 1, \ldots, N_j$, are observations from data group j, for $j = 1, \ldots, J$. The observations are assumed exchangeable within the group and drawn from a distribution $x_{ji} \sim F(\theta_{ji})$. Parameter θ_{ji} is drawn from G_j and all $\{G_j\}_{j=1}^J$ are linked via an nDP. The mathematical representation of the nDP mixture model is

$$x_{ji} \mid \theta_{ji} \sim F(\theta_{ji}), \quad \theta_{ji} \mid G_j \sim G_j, \quad i = 1, \ldots, N_j$$
$$G_j \sim \sum_{k=1}^{\infty} \pi_k \delta_{G_k^*}, \quad j = 1, \ldots, J \tag{5}$$
$$G_k^* = \sum_{l=1}^{\infty} \beta_{kl} \delta_{\theta_{kl}^*}, \tag{6}$$

with $\theta_{kl}^* \sim H$, $\boldsymbol{\beta}_k \sim \text{Stick}(\gamma)$ and $\boldsymbol{\pi} \sim \text{Stick}(\eta)$. The collection $\{G_1, \ldots, G_J\}$, used as the mixing distribution, is said to follow a *nested Dirichlet process* with parameters η, γ and H, and is denoted $\text{nDP}(\eta, \gamma, H)$.

Equation (5) implies that the distribution G_j is a stick-breaking process, in which the atoms are themselves stick-breaking processes drawn from $DP(\gamma, H)$. Since $P(G_j = G_{j'}) = \frac{1}{1+\eta} > 0$, the model induces clustering in the space of distributions. Also, the stick breaking construction of G_k^* ensures that marginally, $G_j \sim DP(\gamma, H)$ for every j.

Rodriguez et al. (2006) showed that the prior correlation between two distributions G_j and $G_{j'}$ is $\text{Cor}(G_j, G_{j'}) = \frac{1}{1+\eta} = P(G_j = G_{j'})$. In addition, the correlation between draws from the process can be calculated from (5) and (6), yielding

$$\text{Cor}(\theta_{ji}, \theta_{j'i'}) = \begin{cases} \frac{1}{1+\gamma} & j = j' \\ \frac{1}{(1+\eta)(1+\gamma)} & j \neq j' \end{cases}. \tag{7}$$

The above indicates that the *a priori* correlation between observations coming from the same group is larger than the correlation between observations coming from different groups, which is an appealing feature. Therefore, the nDP model simultaneously enables clustering the observations across groups as well as clustering the distributions at the task level. This is different from the HDP, in which only data-level clustering across groups is considered (the task-level clustering in the HDP may be implied indirectly).

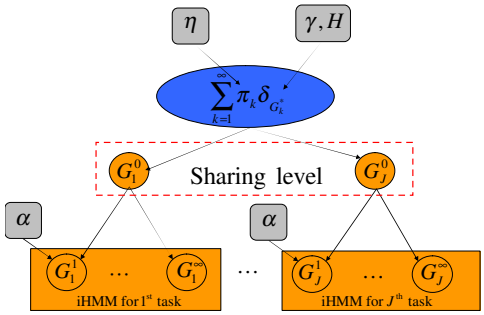

Figure 3. The graphical representation of the nDP-iHMM. The base distributions are shared via a nDP and the consequent iHMMs are independently generated given the base distribution.

3.2. The nDP-iHMM

While in the nDP exchangeable observations within each task are assumed, in iHMM-based MTL we have to consider sequential data for each task. In the nDP-iHMM model, the distribution of each task is now replaced by an iHMM and those iHMMs share an nDP prior. To be specific, the collection of base distributions $\{G_1^0, \ldots, G_J^0\}$ for the iHMMs are drawn from an nDP. A graphical model of the nDP-iHMM is shown in Fig. 3.

Two tasks j and j' share the same observation models (mixture components defined in the base distribution) if $G_j^0 = G_{j'}^0 = G_k^*$ for some k. Note that even though the base distribution G_j^0 and $G_{j'}^0$ are identical, the consequent iHMMs can still be different because the rows of transition matrices are random draws from the mixture weights of the base distribution. This property is reflected in the following equations:

$$G_j^n \mid (G_j^0 = G_k^*) = \sum_{l=1}^{\infty} w_{nkl}^j \delta_{\theta_{kl}^*}, \quad n = 1, \ldots, \infty$$
$$G_j^0 \sim \sum_{k=1}^{\infty} \pi_k \delta_{G_k^*}, \quad G_k^* = \sum_{l=1}^{\infty} \beta_{kl} \delta_{\theta_{kl}^*}, \tag{8}$$

where $\{\boldsymbol{w}_{nk}^j\}_{n=1}^{\infty} \sim \text{DP}(\alpha, \boldsymbol{\beta}_k)$ are rows of the transition matrix for the j^{th} iHMM given that the base distribution is equal to G_k^*.

The hyperparameters reflect the prior knowledge of

how similar the tasks are. As $\eta \to \infty$, each base distribution G_j^0 is assigned a distinct atom of a G_k^*, resulting in all the tasks being quite different, therefore separate iHMM learning is performed. On the other hand, as $\eta \to 0$, the prior information indicates all the base distributions almost use the same atom G^*, which corresponds to the special case of DP-iHMM. Moreover, as $\gamma \to 0$, each iHMM degenerates to a single-state HMM and the model reduces to parametric-based clustering.

3.3. Inference for nDP-iHMM

Let $\boldsymbol{O}_j = \{o_{j1}, \ldots, o_{jT}\}$ represent the observation sequence from task j and let indicator variable c_j denote the atom G_k^* for which $G_j^0 = G_{c_j}^*$. The nDP-iHMM mixture model can be written as follows

$$
\begin{aligned}
& o_{jt} \mid c_j, s_{jt}, \{\theta_{lk}^*\}_{l,k=1}^{\infty} \sim F(\theta_{s_{jt}c_j}^*) \\
& \{\theta_{lk}^*\}_{l,k=1}^{\infty} \mid H \sim H \\
& s_{jt} \mid c_j, s_{jt-1}, \{\boldsymbol{w}_{nk}^j\}_{n,k=1}^{\infty} \sim \text{Mult}(\boldsymbol{w}_{s_{jt-1}c_j}^j) \\
& \{\boldsymbol{w}_{nk}^j\}_{n,k=1}^{\infty} \mid c_j, \alpha, \{\boldsymbol{\beta}_k\}_{k=1}^{\infty} \sim \text{DP}(\alpha, \boldsymbol{\beta}_{c_j}) \\
& \{\boldsymbol{\beta}_k\}_{k=1}^{\infty} \mid \gamma \sim \text{Stick}(\gamma) \\
& c_j \mid \boldsymbol{\pi} \sim \text{Mult}(\boldsymbol{\pi}) \qquad \boldsymbol{\pi} \mid \eta \sim \text{Stick}(\eta). \qquad (9)
\end{aligned}
$$

The nDP-iHMM inference for a J-task MTL problem is based on a Gibbs sampler. We truncate the top level stick-breaking representation to K components ($K = 20$ in our experiments) (Ishwaran & James, 2001), and the initialization includes nDP mixture weights $\{\pi_k\}_{k=1}^K$, center index c_j, hidden state sequences $\{s_{jt}\}_{t=1}^T$, iHMM parameters $\{\text{iHMM}_{jk}\}_{k=1}^K$ for $j = 1, \ldots, J$ and hyperparameters η, γ and α. We put Gamma priors on the hyperparameters and repeat the following steps until the Gibbs sampler converges:

1. Draw new center index c_j' according to $p(c_j' = k) \propto \pi_k \cdot p(\boldsymbol{O}_j|\text{iHMM}_{jk})$ for every task j.

2. For those tasks whose center indices are changed $c_j' \neq c_j$, generate new hidden state sequence s_{jt} using the parameter $\text{iHMM}_{jc_j'}$. Recalculate the unique HMM transition matrix A_j for $j = 1, \ldots, J$ and the shared HMM observation matrix B_k for $k = 1, \ldots, K$. Both of the A's and B's are represented as counting matrices.

3. Sample new hidden state sequence s_{jt} in the way similar to the iHMM inference

$$
\begin{aligned}
& p(s_{jt} = l \mid \boldsymbol{S}_j^{-t}, \boldsymbol{O}_j, c_j) = \\
& p(s_{jt} = l | s_{jt-1} = r, s_{jt+1} = q, o_{jt}, c_j = k) \propto \\
& \begin{cases} (n_{jrl}^{-rs'_{jt}} + \alpha\beta_{kl}) \frac{\alpha\beta_{kq} + n_{lq}}{\alpha + \sum_{l'=1}^{L} n_{ll'}} f_l^{-o_{jt}}(o_{jt}), & \text{if } l \in 1 \sim \text{L}; \\ \alpha\beta_{kl}\beta_{kq} f_{l^{\text{new}}}^{-o_{jt}}(o_{jt}), & \text{if } l = l^{\text{new}}, \end{cases} \qquad (10)
\end{aligned}
$$

with $\boldsymbol{S}_j^{-t}$ being the hidden state sequence excluding s_{jt}, s'_{jt} the previous sampled value of s_{jt}, n_{jrl} the count of transitions from state value r to state value l in task j, and β_{kl} the mixing weight for state l given the task is using G_k^*.

4. Sample new A's by counting the transitions using s_{jt}'s. Sample new B's (mixture components) according to its posterior distribution. Generate new iHMM_{jk} parameters from the pair of A_j and B_k.

5. Sample nDP mixture weight π_k from $\pi_k = v_k \prod_{i=1}^{k-1}(1 - v_i)$, $v_k \sim \text{Beta}(1 + m_k, \eta + \sum_{s=k+1}^{K} m_s)$, $k = 1, \ldots, K-1$, $v_K = 1$, where $m_k = \sum_{j=1}^{J} I(c_j = k)$.

6. Sample η in the way similar to DP. Sample γ and α in the way similar to iHMM.

This Gibbs sampler involves simple steps for sampling from standard distributions and is quite easy to implement in general problems, as long as the sample size is not massive. Convergence tended to be rapid and mixing good in examples we have considered.

4. Experiments

4.1. Synthetic Data

We apply the nDP-iHMM for discovering the relationships between 12 synthetic data sets. Each data set contains 50 sequences of length 20, generated from a distinct discrete HMM. For all HMMs, the number of states $M = 2$ and the codebook size $C = 8$. The parameters of the HMMs have the form of $\{\boldsymbol{\pi}_j = \bar{\boldsymbol{\pi}}_j, \boldsymbol{A}_j = \bar{\boldsymbol{A}}_j + \boldsymbol{\epsilon}_j^A, \boldsymbol{B}_j = \bar{\boldsymbol{B}}_j + \boldsymbol{\epsilon}_j^B\}$, where $\boldsymbol{\epsilon}_j^A = \begin{bmatrix} \epsilon_{j,11}^A & \epsilon_{j,12}^A \\ \epsilon_{j,21}^A & \epsilon_{j,22}^A \end{bmatrix}$ and $\boldsymbol{\epsilon}_j^B = \begin{bmatrix} \epsilon_{j,11}^B & \epsilon_{j,12}^B & \epsilon_{j,13}^B & \epsilon_{j,14}^B & \epsilon_{j,15}^B & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & \epsilon_{j,24}^B & \epsilon_{j,25}^B & \epsilon_{j,26}^B & \epsilon_{j,27}^B & \epsilon_{j,28}^B \end{bmatrix}$. Each non-zero element in $\boldsymbol{\epsilon}_j^A$ and $\boldsymbol{\epsilon}_j^B$ is independently drawn from a uniform distribution on the interval [0, 0.05]. For the first three HMMs, $j = 1, 2, 3$, they use the same $\bar{\boldsymbol{\pi}}_j = \begin{bmatrix} 0.6950 \\ 0.3050 \end{bmatrix}$, $\bar{\boldsymbol{A}}_j = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$, and $\bar{\boldsymbol{B}}_j = \begin{bmatrix} 0.05 & 0.1 & 0.7 & 0.1 & 0.05 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.05 & 0.1 & 0.7 & 0.1 & 0.05 \end{bmatrix}$, implying that the first three tasks have the same HMM parameters except that a small distortion is added. Similarly, for $j = 4, 5, 6, 7$, $\bar{\boldsymbol{\pi}}_j = \begin{bmatrix} 0.8724 \\ 0.1276 \end{bmatrix}$, $\bar{\boldsymbol{A}}_j = \begin{bmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{bmatrix}$, and $\bar{\boldsymbol{B}}_j =$

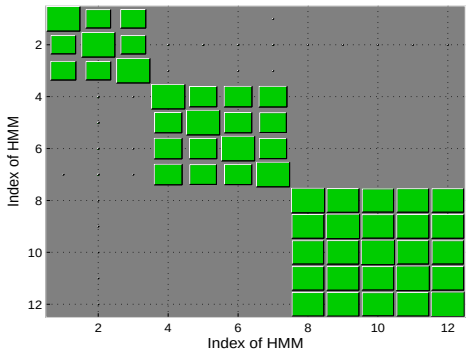

Figure 4. Hinton diagram of between-task similarities for the synthetic problem.

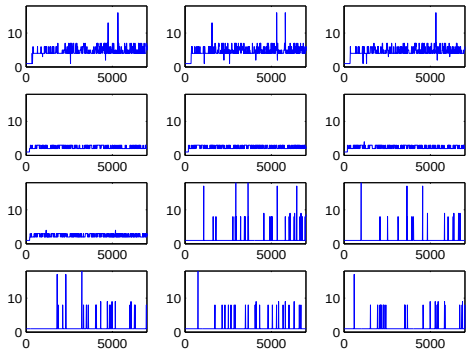

Figure 5. The evolution of center indices versus Gibbs iterations for the 12 tasks.

$\begin{bmatrix} 0.0 & 0.0 & 0.05 & 0.1 & 0.7 & 0.1 & 0.05 & 0.0 \\ 0.0 & 0.05 & 0.1 & 0.7 & 0.1 & 0.05 & 0.0 & 0.0 \end{bmatrix}$;
$\bar{\boldsymbol{\pi}}_j = \begin{bmatrix} 0.4729 \\ 0.5271 \end{bmatrix}$, $\bar{\boldsymbol{A}}_j = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$, and $\bar{\boldsymbol{B}}_j = \begin{bmatrix} 0.05 & 0.0 & 0.0 & 0.0 & 0.05 & 0.1 & 0.7 & 0.1 \\ 0.1 & 0.05 & 0.0 & 0.0 & 0.0 & 0.05 & 0.1 & 0.7 \end{bmatrix}$ for $j = 8, \ldots, 12$. Therefore there are three clusters among these 12 tasks.

The nDP-iHMM is applied to clustering the tasks. The H is set to be a Dirichlet distribution with parameters all equal to 1 and $Ga(1, 1)$ is imposed on hyperparameters η, γ and α. We initialize all tasks with the same center indices and let the nDP-iHMM infer the true underlying relationships between the tasks. The result is shown by the Hinton diagram plotted in Fig. 4. In the Hinton diagram, the size of the green box is proportional to the degree of similarity between two tasks. The similarity measure corresponds to the posterior probability that two tasks are grouped together, which can be calculated by the proportion of Gibbs sampling draws where tasks are assigned to the same cluster. Note that this approach relies on soft probabilistic clustering, so that the posterior mean estimate of the base distribution for two tasks will always differ, but these estimates will converge as the posterior probability of clustering increases. The evolution of center indices are shown in Fig. 5. It can be seen that the nDP-iHMM clusters the tasks well even when we initialize with the same center indices.

4.2. Music Data

To demonstrate application of the nDP-iHMM on real data, we consider the problem of music analysis. In this experiment, we have ten 1-minute music clips extracted from different pieces. The reason for choosing part of the piece instead of the whole piece is that we are confident on the similarities of those clips. In this way we are able to control the ground truth for the real application using our proposed model. These ten clips were chosen deliberately with the following intended clustering: 1) clip 1 is unique in style and instrumentation; 2) clips 2 and 3, 4 and 5, 6 and 7, and 9 and 10 are intended to be paired together; 3) clip 8 is also unique, but is of the same format (instrumentation) as clips 6 and 7 (the names of the pieces are given in Fig. 6).

We wish for the nDP-iHMM to learn the relationships between the clips, i.e., the similarities of these clips. Meanwhile, we wish to learn an accurate iHMM for each of the clips simultaneously. Each music clip is sampled at 22 kHz and 10-dimensional Melfrequency cepstral coefficient (MFCC) (Logan & Salomon, 2001) features are extracted for every 25 ms non-overlapping frame. The feature vectors across all the ten clips are concatenated to perform vector quantization (VQ) (Linde et al., 1980), mapping each feature vector to a code within a VQ codebook of size 32. In our experiment, we choose a sequence of 1 second windows, or 40 observations. Therefore each music clip is transformed into 60 data sequences with 40 observations inside each sequence.

We compare three methods for iHMM model learning: (i) the proposed nDP-iHMM method, (ii) DP-iHMM, for which a master level DP is used to couple all the iHMMs, and (iii) STL-iHMM – the single task-learning method, for which each clip is analyzed in isolation.

We compare the performance of the three methods by evaluating the average of testing-sequence likelihood averaged first within clip and then over all the clips. For each clip, a certain number of training sequences are selected and the remaining sequences are used as testing data for that clip-dependent iHMM. The training data are chosen from the middle of the clip, because there may be a long quiet period in the two ends. To have a comprehensive comparison, different training set sizes are considered: $2, 4, \ldots, 10, 20$ and

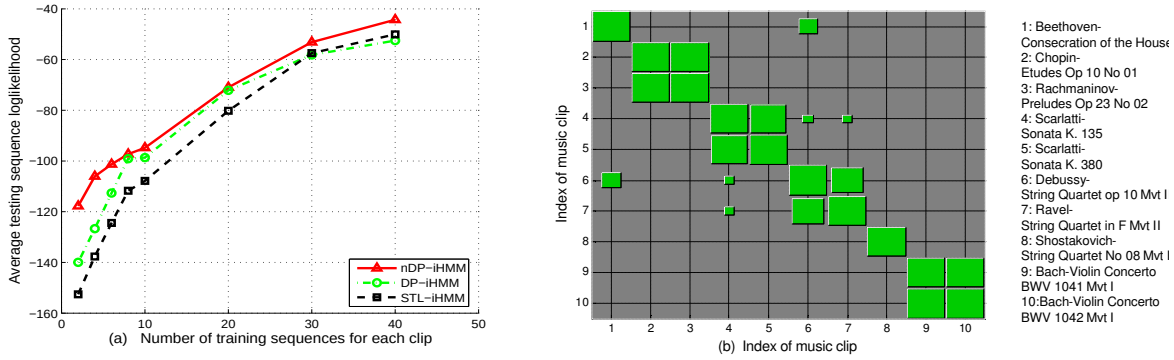

Figure 6. (a) The average testing sequence likelihood using nDP-iHMM, DP-iHMM and STL-iHMM; (b) the Hinton diagram of between-clip similarities based on sampled center indices in nDP-iHMM for the case of 20 training sequences.

30 sequences are used. All three methods are implemented via Gibbs samplers and the Raftery and Lewis test (Raftery & Lewis, 1992) is performed to determine the number of iterations needed for convergence. All results shown in Fig. 6 are based on 20709 samples obtained after a burn-in period of 2098 iterations. The setting of hyperparameters are the same as in Sec. 4.1. Figure 6(a) shows that the proposed nDP-iHMM method consistently outperforms the other two methods, and the improvement is more dramatic when there is a small amount of training data available. This is because the nDP-iHMM performs data-level clustering and the clip-level clustering simultaneously. This property is reflected by the Hinton diagram shown in Fig. 6(b). It is clear that the nDP-iHMM captures the between-clip similarity quite well.

The STL-iHMM and DP-iHMM do not provide a direct measure of inter-task similarity, as provided by the nDP. However, one may use an appropriate distance measure to compute the similarity of the learned iHMMs. For this purpose, we use a distance measure similar to that considered by Aucouturier and Pachet (2002). The distance between two iHMMs is defined as

$$D(\text{iHMM}_i, \text{iHMM}_j) = \frac{1}{2K}\sum_{i=1}^{K}[\log p(\boldsymbol{S}_i|\text{iHMM}_i) - \log p(\boldsymbol{S}_i|\text{iHMM}_j)] + \frac{1}{2K}\sum_{i=1}^{K}[\log p(\boldsymbol{S}_j|\text{iHMM}_j) - \log p(\boldsymbol{S}_j|\text{iHMM}_i)], \quad (11)$$

where $\boldsymbol{S}_i$'s are sequences simulated from iHMM_i and $\boldsymbol{S}_j$'s are sequences simulated from iHMM_j. The similarity between clip i and clip j is then calculated as

$$Sim(i,j) = exp(-\frac{|D(\text{iHMM}_i, \text{iHMM}_j)|^2}{\sigma^2}), \quad (12)$$

where the variance σ^2 is arbitrary. We compute similarities of clips using (12) for the case of 6 training sequences, and plot the Hinton diagrams for all the three methods in Figs. 7(a), (b) and (c), respectively. We observe that the nDP-iHMM (Fig. 7(a)) does the best in discovering the sharing structure of the music clips with limited training data.

5. Conclusion

We have proposed a new hierarchical Bayesian model for multi-tasking learning with sequential data. The infinite hidden Markov model (iHMM) is used to model each task, solving the fundamental problem of model selection in HMMs. A nested Dirichlet process (nDP) is then imposed as a prior for the iHMMs, providing task-level clustering as well as data-level clustering (here the data-level clustering corresponds to the HMM states). The clustered iHMMs share the same base distribution (observation matrix) but have different transition matrices, which results in unique generative models for each task. Inference for the nDP-iHMM is based on a Gibbs sampler and promising nDP-iHMM results have been demonstrated on both simulated data and real (music) data.

An important area for future research involves development of techniques to improve computational efficiency. While the learned sharing mechanism between the musical pieces is very encouraging, the MCMC sampler is too expensive computationally for practical implementation. Therefore, future research will consider more approximate but efficient inference engines, such as variational Bayesian analysis.

References

Aucouturier, J. J., & Pachet, F. (2002). Music similarity measures: Whats the use? *International Sym-*

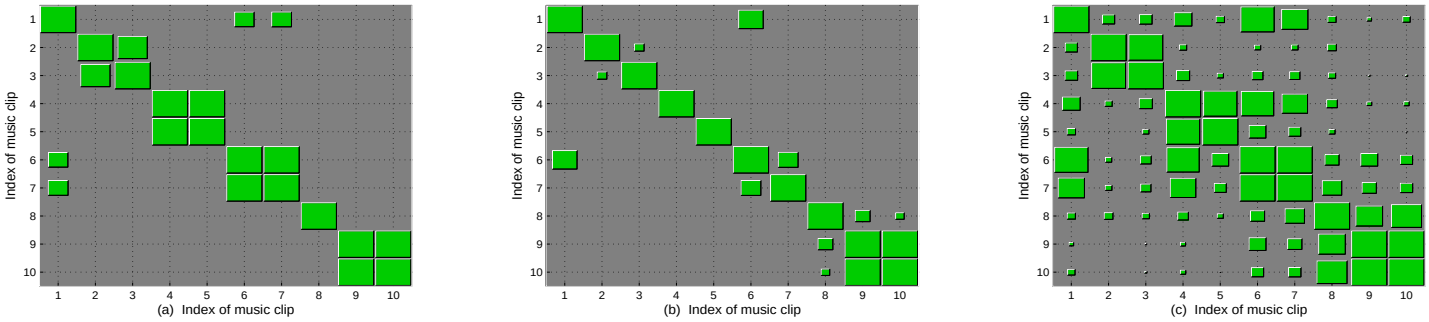

Figure 7. (a) the between-clip similarity matrix computed by (12) using nDP-iHMM results for the case of 6 training sequences; (b) and (c) are same as (a) but using DP-iHMM and STL-iHMM respectively.

posium on Music Information Retrieval (ISMIR).

Beal, M. J. (2003). *Variational algorithms for approximate Bayesian inference*. Doctoral dissertation, Gatsby Computational Neuroscience Unit, University College London.

Beal, M. J., Ghahramani, Z., & Rasmussen, C. (2002). The infinite Hidden markov model. *Neural Information Processing Systems*.

Blei, D. M., Griffiths, T. L., Jordan, M. I., & Tenenbaum, J. B. (2004). Hierarchical topic models and the nested Chinese restaurant process. *Neural Information Processing Systems*.

Caruana, R. (1997). Multitask learning. *Machine Learning*, *28*, 41–75.

Escobar, M. D., & West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, *90*, 577–588.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, *1*, 209–230.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubim, D. B. (Eds.). (1995). *Bayesian data analysis*. Chapman and Hall.

Ishwaran, H., & James, L. F. (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, *96*, 161–173.

Linde, Y., Buzo, A., & Gray, R. M. (1980). An algorithm for vector quantizer design. *IEEE Trans. Communications*, *COM-28*, 84–95.

Logan, B., & Salomon, A. (2001). A music similarity function based on signal analysis. *IEEE International Conference on Multimedia and Expo*.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*, 257–286.

Raftery, A. E., & Lewis, S. M. (1992). How many iterations in the Gibbs sampler? *Bayesian Statistics*, *4*, 763–773.

Rasmussen, C. (2000). The infinite Gaussian mixture model. *Neural Information Processing Systems*.

Rodriguez, A., Dunson, D. B., & Gelfang, A. E. (2006). The nested Dirichlet process. *Journal of the American Statistical Association*, *submitted*.

Runkle, P., Bharadwaj, P. K., Couchman, L., & Carin, L. (1999). Hidden Markov models for multi-aspect target classification. *IEEE Transactions on Signal Processing*, *47*, 2035–2040.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, *4*, 639–650.

Stolcke, A., & Omohundro, S. (1993). Hidden Markov model induction by Bayesian model merging. *Advances in Neural Information Processing Systems*.

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, *101(476)*, 1566–1581.

Thurn, S., & O'Sullivan, J. (1996). Discovering structure in multiple learning tasks: The TC algorithm. *The 13th International Conference on Machine Learning*.

Xue, Y., Liao, X., Carin, L., & Krishnapuram, B. (2007). Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research*, *Jan.*, 35–63.