Beam Sampling for the Infinite Hidden Markov Model

Jurgen Van Gael JV279@CAM.AC.UK
Yunus Saatci YS267@CAM.AC.UK
Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK

Yee Whye Teh YWTEH@GATSBY.UCL.AC.UK
Gatsby Computational Neuroscience Unit, University College London, WC1N 3AR, UK

Zoubin Ghahramani ZOUBIN@ENG.CAM.AC.UK
Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK

Abstract

The infinite hidden Markov model is a nonparametric extension of the widely used hidden Markov model. Our paper introduces a new inference algorithm for the infinite Hidden Markov model called *beam sampling*. Beam sampling combines slice sampling, which limits the number of states considered at each time step to a finite number, with dynamic programming, which samples whole state trajectories efficiently. Our algorithm typically outperforms the Gibbs sampler and is more robust. We present applications of iHMM inference using the beam sampler on changepoint detection and text prediction problems.

1. Introduction

The hidden Markov model (HMM) (Rabiner, 1989) is one of the most widely used models in machine learning and statistics for sequential or time series data. The HMM consists of a hidden state sequence with Markov dynamics, and independent observations at each time given the corresponding state. There are three learning related tasks associated with the HMM: inference of the hidden state sequence, learning of the parameters, and selection of the right model size.

Inference for the hidden state trajectory can be performed exactly using the forward-backward algorithm (Rabiner, 1989), a dynamic programming algorithm with $O(TK^2)$ computational costs where T is the number of time steps and K number of states.

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.

The standard approach to learning uses the Baum-Welch algorithm, a special instance of the EM algorithm (Dempster et al., 1977) which produces (locally) maximum likelihood (ML) parameters. Such ML learning of parameters can potentially lead to overfitting if the model size is inappropriate for the amount of data available. This can be partially mitigated using a more fully Bayesian learning procedure, e.g. using variational approximations (MacKay, 1997) or Markov chain Monte Carlo (MCMC) sampling (Scott, 2002). Such Bayesian approaches also produce estimates of the marginal probability of data, which can be used to select for the appropriate model size (or to average over model sizes if ones desires a more Bayesian analysis). Such model selection procedures can be computationally expensive since multiple HMMs of different sizes need to be explored.

A new twist on the problem of model selection has emerged in recent years with the increasing popularity of nonparametric Bayesian models. These are models of infinite capacity, a finite portion of which will be used to model a finite amount of observed data. The idea of searching/averaging over the space of finite models is replaced with Bayesian inference over the size of submodel used to explain data. Examples of successful applications of nonparametric Bayesian methods include Gaussian Processes (Rasmussen & Williams, 2005) for regression and classification, Dirichlet Process (DP) mixture models (Escobar & West, 1995; Rasmussen, 2000) for clustering heterogeneous data and density estimation, Indian Buffet Processes for latent factor analysis (Griffiths & Ghahramani, 2006), and defining distributions over non-trivial combinatorial objects such as trees (Teh et al., 2008).

The Infinite Hidden Markov Model (iHMM), otherwise known as the HDP-HMM, (Beal et al., 2002) is a non-

parametric Bayesian extension of the HMM with an infinite number of hidden states. Exact Bayesian inference for the iHMM is intractable. Specifically, given a particular setting of the parameters the forward-backward algorithm cannot be applied since the number of states K is infinite, while with the parameters marginalized out all hidden state variables will be coupled and the forward-backward algorithm cannot be applied either. Currently the only approximate inference algorithm available is Gibbs sampling, where individual hidden state variables are resampled conditioned on all other variables (Teh et al., 2006). Unfortunately convergence of Gibbs sampling is notoriously slow in the HMM setting due to the strong dependencies between consecutive time steps often exhibited by time series data (Scott, 2002).

In this paper we propose a new sampler for the iHMM called *beam sampling*. Beam sampling combines two ideas—slice sampling and dynamic programming—to sample whole state trajectories efficiently. Our application of slice sampling (Neal, 2003) is inspired by (Walker, 2007), who used it to limit the number of clusters considered when sampling assignment variables in DP mixtures to a finite number. We apply slice sampling to limit to a finite number the states considered in each time step of the iHMM, so that dynamic programming can be used to sample whole state trajectories efficiently. We call our proposal beam sampling due to its similarity to beam search, a heuristic procedure for finding the maximum a posteriori trajectory given observations in non-linear dynamical systems. The underlying idea in both is to limit the search to a small number of states so that a good trajectory can be found using reasonable computational resources. However, ours is a MCMC sampling method with guaranteed convergence to the true posterior.

We first present a self-contained description of the iHMM using the Hierarchical Dirichlet process (HDP) formalism (Teh et al., 2006) in Section 2, followed by a discussion of Gibbs sampling in Section 3. We introduce beam sampling in Section 4 and compare it against Gibbs sampling on both artificial and real datasets in Section 5. We find that beam sampling is (1) at least as fast if not faster than Gibbs sampling; (2) more robust than Gibbs sampling as its performance is not as dependent on initialization and hyperparameter choice; (3) handles non-conjugacy in the model more naturally; (4) straightforward to implement. We conclude in Section 6 with a discussion and suggestions for other cases in which beam sampling might prove useful. All software is available from *http://mlg.eng.cam.ac.uk/jurgen* to encourage more widespread adoption of the iHMM and the beam sampler.

2. The Infinite Hidden Markov Model

We start this section by describing the finite HMM, then taking the infinite limit to obtain an intuition for the infinite HMM, followed by a more precise definition. A finite HMM consists of a hidden state sequence $\boldsymbol{s} = (s_1, s_2, \ldots, s_T)$ and a corresponding observation sequence $\boldsymbol{y} = (y_1, y_2, \ldots, y_T)$. Each state variable s_t can take on a finite number of states, say $1 \ldots K$. Transitions between states are governed by Markov dynamics parameterized by the transition matrix $\boldsymbol{\pi}$, where $\pi_{ij} = p(s_t = j | s_{t-1} = i)$, while the initial state probabilities are $\pi_{0i} = p(s_1 = i)$. For each state $s_t \in \{1 \ldots K\}$ there is a parameter ϕ_{s_t} which parametrizes the observation likelihood for that state: $y_t | s_t \sim F(\phi_{s_t})$. Given the parameters $\{\pi_0, \boldsymbol{\pi}, \boldsymbol{\phi}, K\}$ of the HMM, the joint distribution over hidden states $\boldsymbol{s}$ and observations $\boldsymbol{y}$ can be written (with $s_0 = 0$):

$$p(\boldsymbol{s}, \boldsymbol{y} | \pi_0, \boldsymbol{\pi}, \boldsymbol{\phi}, K) = \prod_{t=1}^{T} p(s_t | s_{t-1}) p(y_t | s_t)$$

We complete the Bayesian description by specifying the priors. Let the observation parameters $\boldsymbol{\phi}$ be iid drawn from a prior distribution H. With no further prior knowledge on the state sequence, the typical prior for the transition (and initial) probabilities are symmetric Dirichlet distributions.

A naïve way to obtain a nonparametric HMM with an infinite number of states might be to use symmetric Dirichlet priors over the transition probabilities with parameter α/K and take $K \to \infty$. Such an approach has been successfully used to derive DP mixture models (Rasmussen, 2000) but unfortunately does not work in the HMM context. The subtle reason is that there is no coupling across transitions out of different states since the transition probabilities are given independent priors (Beal et al., 2002). To introduce coupling across transitions, one may use a hierarchical Bayesian formalism where the Dirichlet priors have shared parameters and given a higher level prior, e.g.

$$\begin{aligned} \boldsymbol{\pi}_k &\sim \mathsf{Dirichlet}\left(\alpha \boldsymbol{\beta}\right), \\ \boldsymbol{\beta} &\sim \mathsf{Dirichlet}\left(\gamma/K \ldots \gamma/K\right) \end{aligned} \tag{1}$$

where $\boldsymbol{\pi}_k$ are transition probabilities out of state k and $\boldsymbol{\beta}$ are the shared prior parameters. As $K \to \infty$, the hierarchical prior (1) approaches (with some alterations) a *hierarchical Dirichlet process* (Teh et al., 2006).

A hierarchical Dirichlet process (HDP) is a set of Dirichlet processes (DPs) coupled through a shared random base measure which is itself drawn from a DP (Teh et al., 2006). Specifically, each $G_k \sim \mathsf{DP}(\alpha, G_0)$ with shared base measure G_0, which can be understood as the mean of G_k, and concentration parameter $\alpha > 0$, which governs variability around G_0,

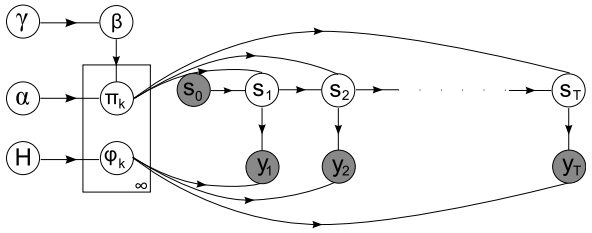

Figure 1. iHMM Graphical Model

with small α implying greater variability. The shared base measure is itself given a DP prior: $G_0 \sim \mathsf{DP}(\gamma, H)$ with H a global base measure. The stick-breaking construction for HDPs shows that the random measures can be expressed as follows: $G_0 = \sum_{k'=1}^{\infty} \beta_{k'} \delta_{\phi_{k'}}$ and $G_k = \sum_{k'=1}^{\infty} \pi_{kk'} \delta_{\phi_{k'}}$, where $\boldsymbol{\beta} \sim \mathsf{GEM}(\gamma)$ is the stick-breaking construction for DPs (Sethuraman, 1994), $\boldsymbol{\pi}_k \sim \mathsf{DP}(\alpha, \boldsymbol{\beta})$, and each $\phi_{k'} \sim H$ independently.

Identifying each G_k as describing both the transition probabilities $\pi_{kk'}$ from state k to k' and the emission distributions parametrized by $\phi_{k'}$, we can now formally define the iHMM as follows:

$$\boldsymbol{\beta} \sim \mathsf{GEM}(\gamma), \qquad \boldsymbol{\pi}_k | \boldsymbol{\beta} \sim \mathsf{DP}(\alpha, \boldsymbol{\beta}), \qquad \phi_k \sim H, \quad (2)$$

$$s_t | s_{t-1} \sim \mathsf{Multinomial}(\boldsymbol{\pi}_{s_{t-1}}), \qquad y_t | s_t \sim F(\phi_{s_t}). \quad (3)$$

The graphical model corresponding to this hierarchical model is shown in figure 1. Thus $\beta_{k'}$ is the prior mean for transition probabilities leading into state k', and α governs the variability around the prior mean. If we fix $\boldsymbol{\beta} = (\frac{1}{K} \ldots \frac{1}{K}, 0, 0 \ldots)$ where the first K entries are $\frac{1}{K}$ and the remaining are 0, then transition probabilities into state k' will be non-zero only if $k' \in \{1 \ldots K\}$, and we recover the Bayesian HMM of (MacKay, 1997).

Finally we place priors over the hyperparameters α and γ. A common solution, when we do not have strong beliefs about the hyperparameters, is to use gamma hyperpriors: $\alpha \sim \mathsf{Gamma}(a_\alpha, b_\alpha)$ and $\gamma \sim \mathsf{Gamma}(a_\gamma, b_\gamma)$. (Teh et al., 2006) describe how these hyperparameters can be sampled efficiently, and we will use this in the experiments to follow.

3. The Gibbs Sampler

The Gibbs sampler was the first sampling algorithm for the iHMM that converges to the true posterior. One proposal builds on the direct assignment sampling scheme for the HDP in (Teh et al., 2006) by marginalizing out the hidden variables $\boldsymbol{\pi}, \boldsymbol{\phi}$ from (2), (3) and ignoring the ordering of states implicit in $\boldsymbol{\beta}$. Thus we only need to sample the hidden trajectory $\boldsymbol{s}$, the base DP parameters $\boldsymbol{\beta}$ and the hyperparameters α, γ. Sampling $\boldsymbol{\beta}, \alpha, \gamma$ is exactly the same as for the HDP so we refer to (Teh et al., 2006) for details.

In order to resample s_t, we need to compute the probability $p(s_t | s_{-t}, \boldsymbol{\beta}, \boldsymbol{y}, \alpha, H) \propto p(y_t | s_t, \boldsymbol{s}_{-t}, \boldsymbol{y}_{-t}, H) \cdot p(s_t | \boldsymbol{s}_{-t}, \boldsymbol{\beta}, \alpha)$. The first factor is the conditional likelihood of y_t given $\boldsymbol{s}$, $\boldsymbol{y}$ and H: $\int p(y_t | s_t, \boldsymbol{\phi}_{s_t}) p(\boldsymbol{\phi}_{s_t} | \boldsymbol{s}_{-t}, \boldsymbol{y}_{-t}, H) d\boldsymbol{\phi}_{s_t}$. This is easy to compute when the base distribution H and likelihood F from equations (2) and (3) are conjugate. For the second factor we can use the fact that the hidden state sequence is Markov. Let n_{ij} be the number of transitions from state i to state j excluding time steps $t-1$ and t. Let $n_{\cdot i}, n_{i \cdot}$ be the number of transitions in and out of state i. Finally, let K be the number of distinct states in $\boldsymbol{s}_{-t}$. Then we have that[1] $p(s_t = k | \boldsymbol{s}_{-t}, \boldsymbol{\beta}, \alpha) \propto$

$$\begin{cases} (n_{s_{t-1},k} + \alpha\beta_k) \frac{n_{k,s_{t+1}} + \alpha\beta_{s_{t+1}}}{n_{k\cdot} + \alpha} & \text{if } k \le K,\ k \ne s_{t-1} \\ (n_{s_{t-1},k} + \alpha\beta_k) \frac{n_{k,s_{t+1}} + 1 + \alpha\beta_{s_{t+1}}}{n_{k\cdot} + 1 + \alpha} & \text{if } k = s_{t-1} = s_{t+1} \\ (n_{s_{t-1},k} + \alpha\beta_k) \frac{n_{k,s_{t+1}} + \alpha\beta_{s_{t+1}}}{n_{k\cdot} + 1 + \alpha} & \text{if } k = s_{t-1} \ne s_{t+1} \\ \alpha\beta_k \beta_{s_{t+1}} & \text{if } k = K+1. \end{cases}$$

For each $1 \le t \le T$ we need to compute $\mathcal{O}(K)$ probabilities, hence the Gibbs sampler has an $\mathcal{O}(TK)$ computational complexity. Non-conjugate models can be handled using more sophisticated sampling techniques. In our experiments below, we used algorithm 8 from (Neal, 2000).

The Gibbs sampler's success is due to its straightforward implementation. However, it suffers from one major drawback: sequential and time series data are likely to be strongly correlated. For example, if we know the value of a stock at time t then we can be reasonably sure that it will be similar at time $t+1$. As is well known, this is a situation which is far from ideal for the Gibbs sampler: strong correlations in the hidden states will make it unlikely that individual updates to s_t can cause large blocks within $\boldsymbol{s}$ to be changed. We will now introduce the beam sampler which does not suffer from this slow mixing behavior by sampling the whole sequence $\boldsymbol{s}$ in one go.

4. The Beam Sampler

The forward-backward algorithm does not apply to the iHMM because the number of states, and hence the number of potential state trajectories, are infinite. The idea of beam sampling is to introduce auxiliary variables $\boldsymbol{u}$ such that conditioned on $\boldsymbol{u}$ the number of trajectories with positive probability is finite. Now dynamic programming can be used to compute the conditional probabilities of each of these trajectories and thus sample *whole* trajectories efficiently. These

[1] Recall that we ignored the ordering of states in $\boldsymbol{\beta}$. In this representation the K distinct states in $\boldsymbol{s}$ are labeled $1 \ldots K$ and $K+1$ denotes a new state.

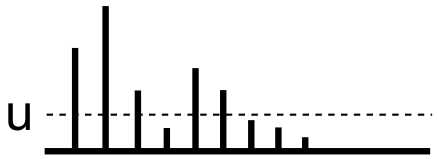

Figure 2. The auxiliary variable u partitions the probability distribution π (vertical bars) into a set of entries less than u and a set of entries larger than u.

auxiliary variables do not change the marginal distribution over other variables hence MCMC sampling will converge to the true posterior. This idea of using auxiliary variables to limit computation costs is inspired by (Walker, 2007), who applied it to limit the number of components in a DP mixture model that need be considered during sampling.

As opposed to the sampler in the previous section, the beam sampler does not marginalize out $\boldsymbol{\pi}$ nor $\boldsymbol{\phi}$. Specifically, the beam sampler iteratively samples the auxiliary variables $\boldsymbol{u}$, the trajectory $\boldsymbol{s}$, the transition probabilities $\boldsymbol{\pi}$, the shared DP parameters $\boldsymbol{\beta}$ and the hyperparameters α and γ conditioned on all other variables. In the following, we shall describe in more detail how to sample each set of variables, as well as how the auxiliary variables allow dynamic programming to be carried out over a finite number of trajectories without approximations.

Sampling $\boldsymbol{u}$: for each t we introduce an auxiliary variable u_t with conditional distribution $u_t \sim \mathsf{Uniform}(0, \pi_{s_{t-1}s_t})$ depending on $\boldsymbol{\pi}$, s_{t-1} and s_t.

Sampling $\boldsymbol{s}$: we sample the whole trajectory $\boldsymbol{s}$ given the auxiliary variables $\boldsymbol{u}$ and other variables using a form of forward filtering-backward sampling. The important observation here is that only trajectories $\boldsymbol{s}$ with $\pi_{s_{t-1}s_t} \geq u_t$ for all t will have non-zero probability given $\boldsymbol{u}$. There are only finitely many such trajectories[2] and as a result we can compute the conditional distribution over all such trajectories efficiently using dynamic programming.

First note that the probability density for u_t is $p(u_t|s_{t-1}, s_t, \boldsymbol{\pi}) = \frac{\mathbb{I}(0<u_t<\pi_{s_{t-1},s_t})}{\pi_{s_{t-1},s_t}}$, where $\mathbb{I}(C) = 1$ if condition C is true and 0 otherwise. We compute $p(s_t|y_{1:t}, u_{1:t})$ for all t as follows (we omitted the additional conditioning variables $\boldsymbol{\pi}$ and $\boldsymbol{\phi}$ for clarity):

$$\begin{aligned}
&p(s_t|y_{1:t}, u_{1:t}) \\
\propto &p(s_t, u_t, y_t|y_{1:t-1}, u_{1:t-1}), \\
= &\sum_{s_{t-1}} p(y_t|s_t)p(u_t|s_t, s_{t-1})p(s_t|s_{t-1}) \\
&\qquad p(s_{t-1}|y_{1:t-1}, u_{1:t-1}), \\
= &p(y_t|s_t) \sum_{s_{t-1}} \mathbb{I}(u_t < \pi_{s_{t-1},s_t}) p(s_{t-1}|y_{1:t-1}, u_{1:t-1}), \\
= &p(y_t|s_t) \sum_{s_{t-1}:u_t<\pi_{s_{t-1},s_t}} p(s_{t-1}|y_{1:t-1}, u_{1:t-1}). \qquad (4)
\end{aligned}$$

Note that we only need to compute (4) for the finitely many s_t values belonging to some trajectory with positive probability. Further, although the sum over s_{t-1} is technically a sum over an infinite number of terms, the auxiliary variable u_t truncates this summation to the finitely many s_{t-1}'s that satisfy both constraints $\pi_{s_{t-1},s_t} > u_t$ and $p(s_{t-1}|y_{1:t-1}, u_{1:t-1}) > 0$. Finally, to sample the whole trajectory $\boldsymbol{s}$, we sample s_T from $p(s_T|y_{1:T}, u_{1:T})$ and perform a backward pass where we sample s_t given the sample for s_{t+1}: $p(s_t|s_{t+1}, y_{1:T}, u_{1:T}) \propto p(s_t|y_{1:t}, u_{1:t})p(s_{t+1}|s_t, u_{t+1})$.

Sampling $\boldsymbol{\pi}$, $\boldsymbol{\phi}$, $\boldsymbol{\beta}$: these follow directly from the theory of HDPs (Teh et al., 2006), but we briefly describe these for completeness.

Let n_{ij} be the number of times state i transitions to state j in the trajectory $\boldsymbol{s}$, where $i, j \in \{1 \dots K\}$, K is the number of distinct states in $\boldsymbol{s}$, and these states have been relabeled $1 \dots K$. Merging the infinitely many states not represented in $\boldsymbol{s}$ into one state, the conditional distribution of $(\pi_{k1} \dots \pi_{kK}, \sum_{k'=K+1}^{\infty} \pi_{kk'})$ given its Markov blanket $\boldsymbol{s}, \boldsymbol{\beta}, \alpha$ is

$$\mathsf{Dirichlet}\left(n_{k1} + \alpha\beta_1 \dots n_{kK} + \alpha\beta_K, \alpha \textstyle\sum_{i=K+1}^{\infty} \beta_i\right),$$

To sample $\boldsymbol{\beta}$ we introduce a further set of auxiliary variables m_{ij} which are independent with conditional distributions

$$p(m_{ij} = m|\boldsymbol{s}, \boldsymbol{\beta}, \alpha) \propto S(n_{ij}, m)(\alpha\beta_j)^m,$$

where $S(\cdot, \cdot)$ denotes Stirling numbers of the first kind. The shared DP parameter $(\beta_1 \dots, \beta_K, \sum_{k'=K+1}^{\infty} \beta_{k'})$ has conditional distribution

$$\mathsf{Dirichlet}\left(m_{\cdot 1} \dots m_{\cdot K}, \gamma\right),$$

where $m_{\cdot k} = \sum_{k'=1}^{K} m_{k'k}$. (Teh et al., 2006; Antoniak, 1974) gives more details.

Finally, each ϕ_k is independent of others conditional on $\boldsymbol{s}$, $\boldsymbol{y}$ and their prior distribution H, i.e. $p(\boldsymbol{\phi}|\boldsymbol{s}, \boldsymbol{y}, H) =$

[2] To see this, note that $u_t > 0$ with probability 1 for each t, since each $\pi_{kk'} > 0$ with probability 1. Given the auxiliary variable u_t, note further that for each possible value of s_{t-1}, u_t partitions the set of transition probabilities out of state s_{t-1} into two sets: a finite set with $\pi_{s_{t-1}k} > u_t$ and an infinite set with $\pi_{s_{t-1}k} < u_t$, as illustrated in figure 2. Thus we can recursively show that for $t = 1, 2 \dots T$ the set of trajectories $s_{1:t}$ with all $\pi_{s_{t'-1}s_{t'}} > u_t$ is finite.

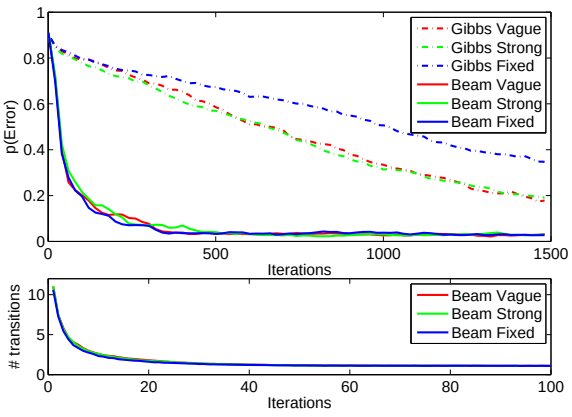

Figure 3. iHMM performance on strong negatively correlated data. The top plot shows the error of the Gibbs and beam sampler for the first 1500 iterations averaged over 20 runs. The bottom plot shows the average number of previous states considered in equation (4) for the first 100 iterations of the beam sampler.

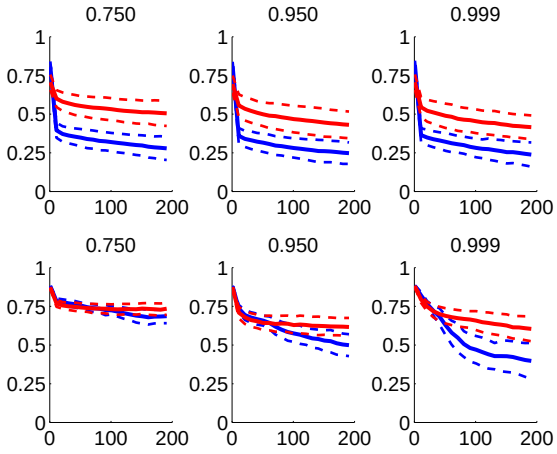

Figure 4. iHMM error on increasing positively correlated data. The blue curve shows the beam sampler while the red curve shows the Gibbs sampler performance. The dotted line show the one standard deviation error bars.

$\prod_k p(\phi_k|\boldsymbol{s},\boldsymbol{y},H)$. When the base distribution H is conjugate to the data distribution F each ϕ_k can be sampled efficiently. Otherwise we may resort to Metropolis-Hastings or other approaches. Note that in the non-conjugate case this is simpler than for Gibbs sampling. In the experimental section, we describe an application where the base distribution and likelihood are non-conjugate.

To conclude our discussion of the beam sampler, it is useful to point out that there is nothing special about sampling u_t from the uniform distribution on $[0, \pi_{s_{t-1},s_t}]$: by choosing a distribution over $[0, \pi_{s_t,s_{t-1}}]$ with higher mass near smaller values of u_t, we will allow more trajectories to have positive probability and hence considered by the forward filtering-backward sampling algorithm. Although this will typically improve mixing time, it also comes at additional computational cost. This brings us to the issue of the computational cost of the beam sampler: since for each timestep and each state assignment we need to sum over all represented previous states, the worst case complexity is $\mathcal{O}(TK^2)$. However, the sum in (4) is only over previous states for which the transition probability is larger than u_t; this means that in practice we might only need to sum over a few previous states. In our experiments below, we will give some empirical evidence for this "average case" behavior. Further, we have found that the drastically improved mixing of the beam sampler more than made up for the additional cost over Gibbs sampling. Finally, although we did not find any advantage doing so, it is certainly possible to interleave the beam sampler and the Gibbs sampler.

5. Experiments

We evaluate the beam sampler on two artificial and two real datasets to illustrate the following properties: (1) the beam sampler mixes in much fewer iterations than the Gibbs sampler; (2) the actual complexity per iteration of the beam sampler is only marginally more than the Gibbs sampler; (3) the beam sampler mixes well regardless of strong correlations in the data; (4) the beam sampler is more robust with respect to varying initialization and prior distribution; (5) the beam sampler handles non conjugate models naturally; (6) the iHMM is a viable alternative to the finite HMM. All datasets and a Matlab version of our software are available at *http://mlg.eng.cam.ac.uk/jurgen*.

5.1. Artificial Data

Our first experiment compares the performance of the iHMM on a sequence of length 800 generated by a 4 state HMM. The hidden state sequence was almost cyclic (1-2-3-4-1-2-3-...) with a 1% probability of self transition: i.o.w the true distribution of hidden states is strong negatively correlated. We use a multinomial output distribution with the following emission matrix

$$\begin{bmatrix} 0.0 & 0.5 & 0.5 \\ 0.6666 & 0.1666 & 0.1666 \\ 0.5 & 0.0 & 0.5 \\ 0.3333 & 0.3333 & 0.3333 \end{bmatrix}.$$

Next we run the Gibbs and beam sampler 20 times from a random initialization with every state randomly chosen between 1 and 20. We test the performance of both samplers using three different hyperparameter settings: (1) vague gamma hyperpriors for α and

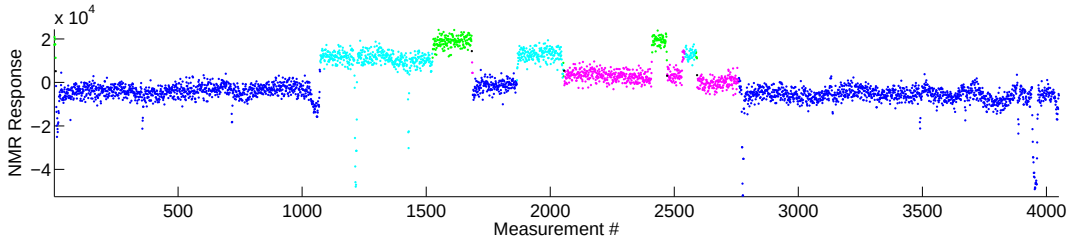

Figure 5. The 40'th sample of the beam sampler with every state represented by a different color on the well-log dataset.

γ (Gamma$(1,1)$ and Gamma$(2,1)$ respectively); (2) strong gamma hyperpriors for α and γ (Gamma$(6,15)$ and Gamma$(16,4)$ respectively); (3) fixed hyperparameters $\alpha = 0.4, \gamma = 3.8$. The latter were chosen using the values the beam and Gibbs samplers converged to. At every iteration, we greedily compute an assignment of sample states to true states to maximize overlap and use the resulting Hamming distance as our error measure. The top plot in figure 3 clearly shows that the beam sampler discovers the underlying structure much faster than the Gibbs sampler. Also, the beam sampler is insensitive to the prior while the performance of the Gibbs sampler becomes worse as we strengthen our prior beliefs. The bottom plot of figure 3 shows how many states are summed over in equation (4) averaged per timestep, per state. We find that after only about 20 iterations, the beam sampler on average considers a little more than one state. This implies that the actual complexity of the beam sampler is closer to $\mathcal{O}(TK)$ rather than the worst case complexity of $\mathcal{O}(TK^2)$. Although this behavior is dependent on the choice of distribution for the auxiliary variable u_t and the sparsity of the transition matrix, we have verified that this behavior is consistent also for larger iHMM's.

Our second experiment illustrates the performance of the beam sampler on data generated from HMM's with increasing positive correlation between the hidden states. We generated sequences of length 4000 from a 4 state HMM with self-transition probabilities increasing from 0.75 to 0.95 and finally 0.999. In one experiment (top plot of figure 4) we generated normal distributed observation from an informative output model with means $-2.0, 4.0, 1.0, -0.5$ and standard deviation 0.5, in another experiment (bottom plot of figure 4) we generated normal distributed observations from a less informative output model with means $-1.0, 0.5, -0.5, 0.0$ and standard deviation 0.5. We initialize the experiment as above and set the base distribution for the state means to be a 0 mean normal with 2.0 standard deviation. Then, we greedily compute the error compared to ground truth and average the results over 60 different random starting positions. The top row shows that with an informative prior, both the Gibbs and beam sampler can reduce the initial error by at least 50% independent of the correlation between hidden states. When the output model is less informative however and there is little correlation between the hidden states, the learning problem is hardest: the lower left plot shows that both the beam and Gibbs sampler discover structure only slowly. When the correlation increases, the learning problem should become easier. However, as the lower right plot shows, although the beam sampler mixes increasingly well, the Gibbs sampler suffers from slow random walk behavior.

5.2. Well Data

The next experiment illustrates the performance of the iHMM on a changepoint detection problem. The data consists of 4050 noisy measurements of nuclear-response of rock strata obtained via lowering a probe through a bore-hole. Figure 5 illustrates this datasets. The data has been previously analyzed in (Ruanaidh & Fitzgerald, 1996) by eliminating the forty greatest outliers and running a changepoint detection algorithm with a fixed number of changepoints. This approach works well as this one-dimensional dataset can be inspected visually to make a decision on whether to throw away datapoints and get a rough idea for the number of changepoints. However, we believe that with a nonparametric model, we can automatically adapt the number of changepoints. Moreover, by setting up a noise model with fat tails, we hope to automatically handle the outlier problem.

We model the mean of the nuclear-response for every segment. First we normalize the data to have zero mean; then we specify a zero mean normal distribution for the base distribution H. We choose the variance of this normal to be the empirical variance of the dataset. For the output model, we let F correspond to a Student-t distribution with $\nu = 1$, also known as the Cauchy distribution. We set the scale parameter for the Cauchy distribution to twice the empirical standard deviation for the dataset. Since the Cauchy likelihood is not conjugate with respect to the normal base distribution, we modified the Gibbs sampler based on algorithm 8 in (Neal, 2000). We use the aux-

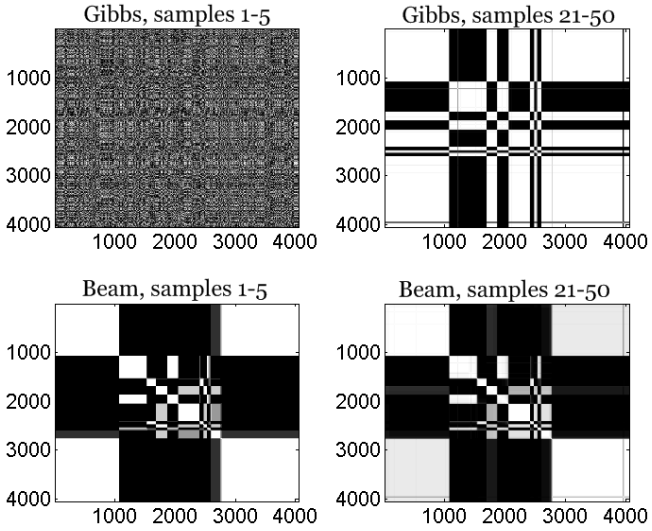

Figure 6. The left plots show how frequent two datapoints were in the same cluster averaged over the first 5 samples. The right plots show how frequently two datapoints were in the same cluster averaged over the last 30 samples.

iliary variable sampling scheme discussed in (Gelman et al., 2004) to resample the segment means.

Figure 5 shows the results of one sample from the beam sampler: the iHMM segments the dataset reasonably well and robustly handles the outliers. To compare the Gibbs and beam samplers, we compute 50 samples after a burnin of 5000 iterations with 1000 iterations in between each sample. For every pair of datapoints we compute the probability that they are in the same segment, averaged over the first five samples (left plots in figure 6) and the last thirty samples (right plots in figure 6). First, note that after the first 10000 iterations, the Gibbs sampler hasn't discovered any structure while the beam sampler has. This supports our claim that the beam sampler mixes faster than the Gibbs sampler. Moreover, we expect that the Gibbs sampler will have trouble to reassign the state assignment for whole segments because of slow random walk behavior. The beam sampler on the other hand resamples whole hidden state sequences and should be able to reassign whole segments more easily. The right plots of figure 6 confirm our expectation: a careful inspection of both plots shows that the Gibbs sampler is visually more black-white indicating that either two datapoints are always in the same cluster or never in the same cluster; the beam sampler, on the other hand, has gray areas which indicate that it averages over different assignments of the segments: e.g. the Gibbs plot (upper right) suggests that the leftmost segment and rightmost segment are *always* in the same state, while the beam sampler plot (bottom right) indicates that only part of the time, the left and rightmost segments are in the same state (90% of the time).

5.3. Alice in Wonderland

Another application domain for HMMs is the area of text prediction. One such task is that of predicting sequences of letters in text taken from *Alice's Adventures in Wonderland*. We compare the performance of a finite HMM trained using variational Bayes (as described in (MacKay, 1997)) with two iHMMs trained using beam sampling and Gibbs sampling. Both samplers had a burn-in of 1000 iterations and an additional 10000 iterations to collect 50 samples of hidden state sequences from the posterior (i.e. we sample every 200 iterations).

The training data for each HMM (whether finite or infinite) was taken to be a single sequence of 1000 characters from the first chapter of the book. There were 31 different observation symbols (26 letters ignoring case plus space and basic punctuation characters). The test data was taken to be the subsequent 4000 characters from the same chapter. For all finite HMMs we analyzed performance on models with the number of hidden states ranging from 1 to 50. For VB, we note that the true predictive distribution is intractable to compute. Therefore, we used the posterior parameter distributions to sample 50 candidate parameter settings, and used these to compute an approximate predictive log-likelihood. For the iHMMs, we sampled 50 hidden state sequences from the stationary distribution after convergence and used these samples to compute an approximate predictive log-likelihood. For the VB-HMM we set the prior pseudo-counts for the transition matrix to $4/K$ across all states and the prior pseudo-counts for the emission matrix to 0.3 across all symbols. Accordingly, we set the hyperprior for the iHMMs such that $a_\alpha = 4$ and $b_\alpha = 1$ and $H \sim \text{Dirichlet}\,(()\,0.3, \cdots 0.3)$. The results for VB and the iHMMs were averaged over 50 and 20 independent

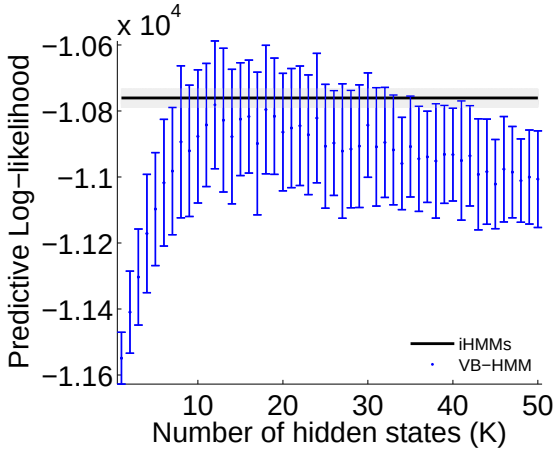

Figure 7. Comparing VB-HMM with the iHMM.

runs respectively. The plot includes error bars corresponding to 2 standard deviations.

Figure 7 illustrates the estimated predictive log-likelihoods for the finite VB-HMM and the two iHMMs trained using beam and Gibbs sampling. We find that the iHMMs have superior predictive power when compared to the VB-HMM, even when we select the best number of hidden states (around $K = 16$). Both the iHMMs converged to a posterior distribution over hidden state sequences with around 16 states, showing that nonparametric Bayesian techniques are an effective way to handle model selection. The final performance of the Gibbs and beam sampler were not found to be significantly different as we set the number of iterations high enough to ensure that both algorithms converge. Indeed, the aim of this experiment is not to compare the performance of individuals iHMM sampling schemes, rather, it is to further illustrate the relative effectiveness of using models of infinite capacity.

6. Conclusion

In this paper we introduced the beam sampler, a new inference algorithm for the iHMM that draws inspiration from slice sampling and dynamic programming to sample whole hidden state trajectories efficiently. We showed that the beam sampler is a more robust sampling algorithm than the Gibbs sampler. We believe that the beam sampler is the algorithm of choice for iHMM inference because it converges faster than the Gibbs sampler and is straightforward to implement. Moreover, it conveniently allows us to learn non-conjugate models. To encourage adoption of the iHMM as an alternative to HMM learning, we have made the software and datasets used in this paper available at *http://mlg.eng.cam.ac.uk/jurgen*.

The beam sampler idea is flexible enough to do inference for various extensions of the iHMM: our current work involves an adaptation of the beam sampler to an extension of the iHMM that handles inputs, effectively resulting in a nonparametric generalization of the input-output HMM (Bengio & Frasconi, 1995). We believe this is a promising model for nonparametric Bayesian learning of POMDPs. Another project currently underway is to use the beam sampler for efficiently learning finite, but very large hidden Markov models. Finally, we are exploring the possibilities of using the embedded HMM construction (Neal et al., 2004) as an alternative for the beam sampler for efficient inference in the iHMM.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. JVG is supported by a Microsoft Research PhD scholarship; ZG is also in the Machine Learning Department, CMU.

References

Antoniak, C. E. (1974). Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The Annals of Statistics*, *2*, 1152–1174.

Beal, M. J., Ghahramani, Z., & Rasmussen, C. E. (2002). The infinite hidden markov model. *NIPS*, *14*.

Bengio, Y., & Frasconi, P. (1995). An input output hmm architecture. *NIPS*, *7*.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, *39*, 1–38.

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

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2004). *Bayesian data analysis*. CRC Press. 2rev ed edition.

Griffiths, T. L., & Ghahramani, Z. (2006). Infinite latent feature models and the indian buffet process. *NIPS*, *18*.

MacKay, D. J. C. (1997). *Ensemble learning for hidden markov models*. Technical report, Cavendish Laboratory, University of Cambridge, 1997.

Neal, R. M. (2000). Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, *9*, 249–265.

Neal, R. M. (2003). Slice sampling. *The Annals of Statistics*, *31*, 705–741.

Neal, R. M., Beal, M. J., & Roweis, S. T. (2004). Inferring state sequences for non-linear systems with embedded hidden markov models. *NIPS*, *16*.

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

Rasmussen, C. E. (2000). The infinite gaussian mixture model. *NIPS*, *12*.

Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian processes for machine learning*. The MIT Press.

Ruanaidh, J., & Fitzgerald, W. J. (1996). *Numerical bayesian methods applied to signal processing*. Springer-Verlag New York Inc.

Scott, S. L. (2002). Bayesian methods for hidden Markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, *97*, 337–351.

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

Teh, Y. W., III, H. D., & Roy, D. (2008). Bayesian agglomerative clustering with coalescents. *NIPS*, *20*.

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

Walker, S. G. (2007). Sampling the dirichlet mixture model with slices. *Communications in Statistics - Simulation and Computation*, *36*, 45.