Variational Inference for Hierarchical Dirichlet Process Based Nonparametric Models

Will Stephenson

Advisor: Erik Sudderth , Reader: Ben Raphael

Abstract

We examine two popular statistical models, the hidden Markov model and mixed membership stochastic blockmodel. Using the hierarchical Dirichlet process, we define nonparametric variants of these models. We develop a memoized online variational inference algorithm that uses a new objective function to properly penalize the addition of unneeded states in either model. Finally, we demonstrate that our models outperform competing methods in a wide array of tasks, including speaker diarization, academic co-authorship network analysis, and motion capture comprehension.

1 Introduction

Many real-world processes follow complex or ill-understood procedures and rules; networks of human relationships develop from a series of often byzantine social rules, and human exercise generates motion capture data that depends on the specifics of human physiology. We seek mathematical models for such processes that are able to explain observed data well while remaining concise. Many popular statistical models do so by assuming the existence of a number of unobserved, or *hidden*, states. The diverse nature of data is then explained by the diversity among the hidden states. The hidden state of human exercise might be the specific exercise being performed at any point in time; an entire sequence of motion capture data can then be understood as a series of particular exercises along with the specific dynamics of each exercise.

We focus on two particular models: the hidden Markov model (HMM) [16] and mixed membership stochastic blockmodel (MMSB) [1] which model time series and relational data, respectively. Both seek to understand the structure of data by defining some number of hidden states or communities. Given these states, they specify a data generation process. For example, audio data of human conversation may be understood as a series of speakers along with their individual speech patterns; a social network may be understood in terms of cliques of friends with high probability of intra-clique interaction and low probability of inter-clique interactions.

Crucially, training either of these models requires us to specify the number of states or communities. In many cases this number is difficult to estimate or explicitly unknown; for example, the speaker diarization task [19] requires us to segment audio data of meetings with an unknown number of speakers. To address this issue, we turn to *nonparametric* models, which assume an unbounded number of states, some finite subset of which are represented in the data. Specifically, we use the *hierarchical Dirichlet process* [18] to define the HDP-HMM [9] and HDP-MMSB [14].

Given a dataset and generic description of a model, we wish to infer the specific model parameters that best match the data. To perform this inference, we use *memoized variational inference* [12]. We derive a new variational objective function for HDP-based models and exploit the structure of the HDP to show it generalizes to both our models. These choices follow the work of [11] on HDP topic models and address issues with previous inference methods for the HDP-HMM and HDP-MMSB.

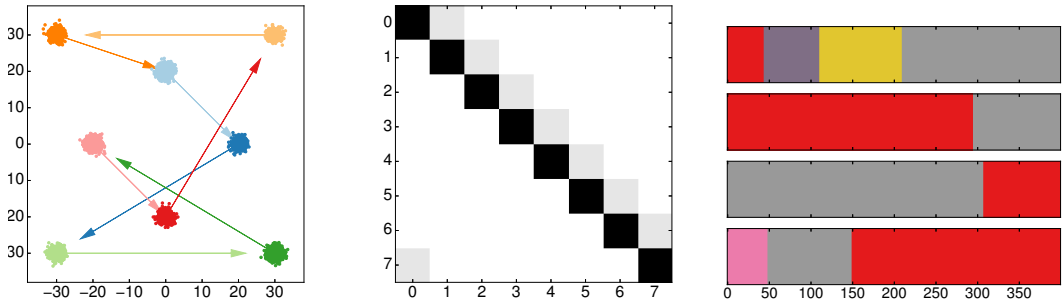

Figure 1: Toy HMM dataset from [7] with $K = 8$ hidden states, 2-dimensional gaussian emissions, and thirty-two sequences of length $T = 1000$. *Left*: Observed data $\{x_t\}$ with arrows indicating transitions between states. *Middle*: Transition matrix π with white corresponding to probability zero. The transition matrix is highly "sticky" in that it has a strong tendency to stay in the same hidden state. *Right*: Example state sequences $\{z_t\}$ for four of the thirty-two sequences.

Prior inference for the HDP-HMM has used either Gibbs sampling [9], which is slow to converge and limited in scalability, or stochastic variational inference with an objective function [13] that has been shown to be problematic [11]. The only prior work for inference in the HDP-MMSB also faces this latter issue [14].

2 Hidden Markov Models

A hidden Markov model (HMM) [16] models sequences of data $\{x_t\}$. In the example of motion capture, our observed $\{x_t\}$ is the position of a human's joints at evenly spaced intervals in time. The HMM assumes that each datapoint x_t is assigned an unobserved *hidden state* z_t that evolves according to a Markov chain. That is, if we represent the hidden state as an indicator vector, where $z_{tk} = 1$ if the HMM is in state k at time t and 0 otherwise, we can write:

$$p(z_{1k}) = \pi_{0k}, \quad p(z_{tm} \mid z_{t-1,\ell}) = \pi_{\ell m} \tag{1}$$

where π_0 is the *starting-state distribution* and π is the *transition distribution*. A symmetric Dirichlet prior is placed on each row of the transition matrix as well as the starting-state: $\pi_i \sim \text{Dirichlet}(\alpha)$.

Given the state z_t, the HMM specifies the probability distribution of the observed datapoint x_t. Formally, it assumes the existance of state specific emission distributions $\{F_k\}$ parameterized by $\{\phi_k\}$ such that if $z_{tk} = 1$, $x_t \sim F_k(\phi_k)$. Figure 5 shows the generative model for a HMM with gaussian emissions, and Figure 1 shows a synthetic dataset following this generative model. Although we only use gaussian and autoregressive gaussian emissions here, we note that any member of the exponential family can be used.

Extension to multiple sequences. In our experiments, we will consider modeling multiple sequences as being drawn from the same HMM. Although derivations are given only for single sequence datasets, the generalization to multiple sequences is extremely simple.

3 Mixed Membership Stochastic Blockmodels

Stochastic blockmodels seek to model graphs over N nodes where edges encode some sort of relationship between nodes. For example, nodes may represent people and directed edges answers to questions such as "does person i like person j?" We also may seek to model relationships that are inherently undirected, such as co-authorship of academic papers. Although one can consider undirected variants of the blockmodels below, we will only consider directed versions and transform undirected datasets by creating two directed edges for every undirected edge.

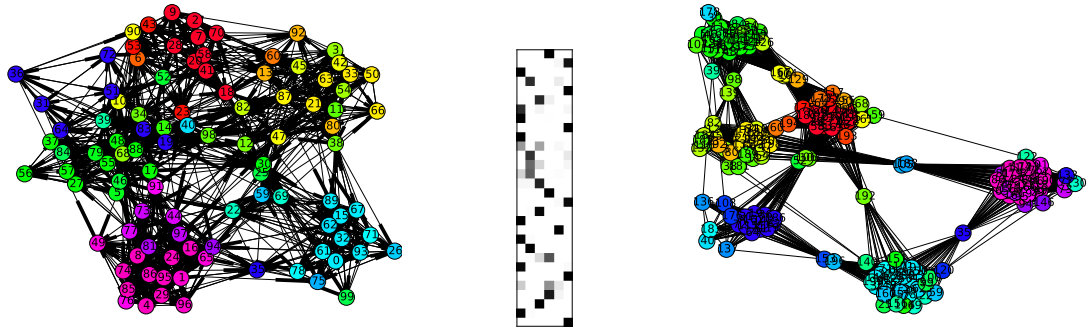

Figure 2: Toy MMSB dataset with $K = 6$ communities and $N = 100$ nodes. Nodes are colored by their individual distributions over communities π_i. *Left*: Actual directed graph generated by the MMSB. *Center*: First thirty community proportions $\pi_{1:30}$ drawn using $\alpha = .05$. *Right*: Graph with an edge drawn between nodes i and j if the probability of an edge $\pi_i^T \phi \pi_j$ is above some threshold. Usage of the model's hidden variables unveils a more detailed structure than indicated by the graph on the left.

Draw $(\mu_\ell, \Sigma_\ell) \sim$ Norm-Wishart$(\cdot)$, for $\ell = 1, \ldots, K$
Draw $\pi_\ell \sim$ Dirichlet(α) for $\ell = 0, \ldots, K$
Draw $z_1 \sim$ Cat(π_0)
Draw $x_1 \sim$ Gauss$(\mu_{z_1}, \Sigma_{z_1})$
for $t = 2 : T$
    Draw $z_t \sim$ Cat$(\pi_{z_{t-1}})$
    Draw $x_t \sim$ Gauss$(\mu_{z_t}, \Sigma_{z_t})$

Draw $\phi_{\ell m} \sim$ Beta(τ_1, τ_0), for $\ell, m = 1, \ldots, K$
for each node i :
    Draw $\pi_i \sim$ Dirichlet(α)
for each node i :
    for each node $j \neq i$:
        Draw $s_{ij} \sim$ Cat(π_i)
        Draw $r_{ij} \sim$ Cat(π_j)
        Draw $x_{ij} \sim$ Bernouli$(\phi_{s_{ij} r_{ij}})$

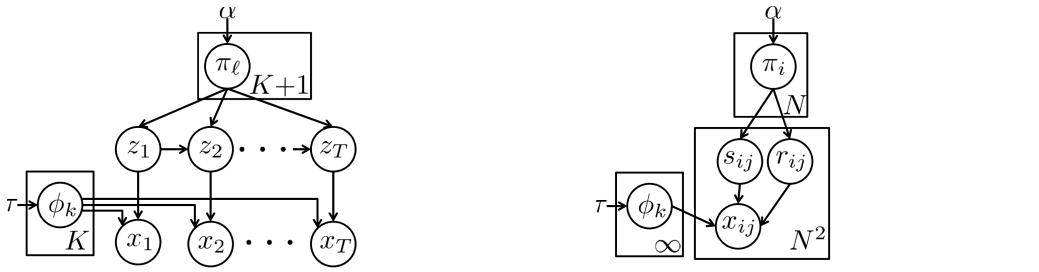

Figure 5: *Top left*: Generative model for a HMM with gaussian emissions. Here, Norm-Wishart$(\cdot)$ is the normal-Wishart distribution over mean, covariance pairs parameterized by a mean vector m, a scale matrix B, a degrees of freedom parameter ν, and scaling parameter κ'. *Top right*: Generative model for the MMSB with hyperparameters α, τ_1, and τ_0. *Bottom*: Graphical models encoding the above generative models.

Similar to the HMM, stochastic blockmodels seek to understand the structure of data through a set of hidden variables. Here, the hidden variables correspond to communities in the graph, given which, the probability of an interaction between any two nodes is specified. For example, two authors belonging to the "Physics" community are much more likely to co-author a paper together than a Physics author and an English author.

The mixed membership stochastic blockmodel (MMSB) allows nodes to participate in multiple communities. This comes from the intuition that actors are often multi-faceted; academics may write about more than one subject and people often participate in more than one circle of friends. To account for this, the MMSB stipulates that a node is allocated a different community for each of its possible interactions. In each directed relationship (i, j), node i is allocated a *source* variable s_{ij} and node j a *receiver* variable r_{ij}. We then draw $x_{ij} \sim \text{Bernouli}(\phi_{s_{ij}r_{ij}})$. For the full generative model, see Figure 5; Figure 2 shows a toy dataset drawn from this model.

Assortativity We primarily consider the assortative MMSB (aMMSB) [10], which changes the generative process of each x_{ij}:

$$x_{ij} \sim \begin{cases} \text{Bernouli}(\phi_k) & s_{ij} = r_{ij} = k \\ \text{Bernouli}(\varepsilon) & s_{ij} \neq r_{ij} \end{cases} \tag{2}$$

where ε is a given – typically small – constant, encoding the idea that the main source of interactions is intra-community. Although the full MMSB can represent more complicated relationships than the aMMSB, previous work has shown the difference to be minimal [10]. For this small cost in modeling ability, we get a large benefit in efficiency: as discussed in Appendix C, the computational cost of inference drops from quadratic to linear in the number of hidden communities, K.

4 Nonparametric Models

As described above, the HMM and MMSB are finite, or *parametric*, models in the sense that the number of hidden states or communities is some finite K. A *nonparametric* version of these models assumes the existence of an infinite number of hidden states. These models are extremely useful in situations when the dimension of the hidden structure is ambiguous or unknown. To create these nonparametric variants, we will require a collection of infinite, discrete distributions to serve as the distributions over the hidden states. To this end, we will use the *hierarchical Dirichlet process*, which generates collections of the more fundamental *Dirichlet process*.

4.1 Dirichlet Processes

A Dirichlet process [6] $DP(\gamma, H)$ with *base distribution* H and *concentration parameter* $\gamma > 0$, is a distribution over probability distributions. Formally, let Ω be the probability space underlying the distribution H. Then, we say that $G \sim DP(\gamma, H)$ if, for any finite partition $A_1, \ldots, A_N$ of Ω, the distribution of G's probability mass on this partition is given by:

$$(G(A_1), \ldots, G(A_N)) \sim \text{Dirichlet}(\gamma H(A_1), \ldots, \gamma H(A_N)) \tag{3}$$

Here, we can see the role of the concentration parameter γ: larger values of γ encourage G to more closely follow the base distribution H, whereas smaller values allow for more deviation.

A more informative – and computationally useful – definition of the Dirichlet process is the *stick-breaking construction* [17], which defines $G \sim DP(\gamma, H)$ in terms of *stick-breaking weights* $\{u_k\}_{k=1}^{\infty}$ and *stick-breaking pieces* $\{\beta_k\}_{k=1}^{\infty}$. Specifically, draws $G \sim DP(\gamma, H)$ can be constructed by:

$$u_k \sim \text{Beta}(1, \gamma), \quad \phi_k \sim H, \quad \beta_k \triangleq u_k \prod_{\ell=1}^{k-1} (1 - u_\ell) \tag{4}$$

$$G(x) = \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k}(x) \tag{5}$$

where $\delta_{\phi_k}(x)$ is the Dirac delta function centered at point ϕ_k. Eq. 5 shows the very important fact that draws from Dirichlet processes are discrete. This makes the Dirichlet process a natural choice for the distribution over hidden communities or states in many popular models, such as mixture models or simplifications of the MMSB from Section 3.

4.2 Hierarchical Dirichlet Processes

In models that require multiple distributions over hidden communities, such as the HMM or MMSB, simply using independent draws from $DP(\gamma, H)$ will create distributions with little to no similarity between them. To induce a level of similarity between these distributions, we turn to the *hierarchical Dirichlet process* [18], which defines a top-level distribution $G_0 \sim DP(\gamma, H)$ and any number of lower-level distributions $G_i \sim DP(\alpha, G_0)$, where $\alpha > 0$ is a second concentration parameter.

With G_0 defined as in Eq. 4-5, [18] shows that we can also write a stick-breaking representation for $G_i \sim DP(\alpha, G_0)$ with stick-breaking pieces π_i and stick-breaking weights v_i:

$$v_{ik} \sim \text{Beta}\left(\alpha u_k \textstyle\prod_{\ell=1}^{k-1}(1-u_\ell), \alpha \prod_{\ell=1}^{k-1}(1-u_\ell)\right), \quad \pi_{ik} \triangleq v_{ik} \textstyle\prod_{\ell=1}^{k-1}(1-v_{i\ell}) \tag{6}$$

$$G_i(x) = \sum_{k=1}^{\infty} \pi_{ik} \delta_{\phi_k}(x) \tag{7}$$

Eq. 6 gives us an explicit representation for each of the infinitely many entries of every π_i. However, it is more convenient to consider a finite partition of the entries of π_i: $[\pi_{i1}, \ldots, \pi_{iK}, \pi_{i>K}]$, where $\pi_{i>K} \triangleq \sum_{\ell=K+1}^{\infty} \pi_{i\ell}$. Using the definition of Dirichlet processes from Eq. 3, the distribution over this partition is:

$$[\pi_{i1}, \ldots, \pi_{iK}, \pi_{i>K}] \sim \text{Dirichlet}(\alpha\beta_1, \ldots, \alpha\beta_K, \alpha\beta_{>K}) \tag{8}$$

where $\beta_{>K} \triangleq \sum_{\ell=K+1}^{\infty} \beta_\ell$.

4.3 The HDP-HMM

We can now define the hierarchical Dirichlet process hidden Markov model (HDP-HMM) [2, 9], which allows for an infinite number of hidden states. The generative model is very similar to the parametric version described in Section 2; the hidden state z advances according to a Markov chain with (now infinite) transition matrix π, and observation x_t is sampled according to emission denstiy $F(\phi_{z_t})$. However, while the finite HMM places a Dirichlet(α) prior on each row of π, the HDP-HMM considers each of the infinitely many rows of π to be the stick-breaking pieces of some $G_i \sim DP(\alpha, G_0)$.

Sticky Hyperparameter As noted in [9], the HDP-HMM can prefer to explain data by rapidly switching between many hidden states. In some cases, this may lead to unrealistic segmentations of the data; for example, in a recording of a conversation between multiple people, we expect the same speaker to persist for at least a second. To this end, we use the *sticky hyperparameter* $\kappa \geq 0$ of [9], which alters the distribution over each π_ℓ given in Eq. 8:

$$[\pi_{\ell 1}, \ldots, \pi_{\ell >K}] \sim \text{Dirichlet}(\alpha\beta_1, \ldots, \alpha\beta_\ell + \kappa, \ldots, \alpha\beta_{>K}) \tag{9}$$

For large values of κ, π is encouraged to have significant mass on its diagonal entries. That is, the hidden state Markov chain is strongly biased towards self-transition.

4.4 The HDP-aMMSB

The HDP-aMMSB is also very similar its parametric counterpart; community memberships π now use the HDP as a prior, and ϕ is now infinitely large with $\phi_{kk} \sim \text{Beta}(\tau_1, \tau_0)$.

5 Variational Inference

For nonparametric models, exact inference of the model parameters is intractable. That is, for a model with parameters $\Theta = \{\Theta_1, \ldots, \Theta_M\}$ (e.g. the hidden states, transition matrix, and emission densities of a HMM) and dataset x, there is no closed form for the parameters Θ that maximize the log-posterior $\log p(\Theta \mid x)$. To resolve this issue, we turn to variational inference [4], which defines a *variational distribution*, $q(\Theta)$, that is a simplification of the true posterior $p(\Theta \mid x)$. We then seek to find the element of this family that minimizes the relative entropy to the true posterior:

$$q^*(\Theta) = \underset{q}{\operatorname{argmin}}\, D(q||p) \tag{10}$$

Standard variational methods give a coordinate ascent algorithm in which factor $q(\Theta_i)$ is updated by holding all $q(\Theta_j)$ for $i \neq j$ fixed. The update for each factor is given by

$$q^*(\Theta_i) \propto \exp\Big[\mathbb{E}_{j \neq i}[\log p(x, \Theta)]\Big] \tag{11}$$

where $\mathbb{E}_{j \neq i}$ indicates an expectation taken over all $q(\Theta_j)$ for $j \neq i$. To determine convergence of a variational algorithm, we examine convergence of a lower bound, $\mathcal{L}$, on the marginal evidence: $\mathcal{L} \leq \log p(x)$. Variational methods give a standard computation for $\mathcal{L}$:

$$\mathcal{L} = \mathbb{E}_q[\log p(x, \Theta) - \log q(\Theta)] \tag{12}$$

where the expectation is taken over all $q(\Theta_i)$. It is possible to show that the q corresponding to the global maximum of $\mathcal{L}$ is the exact solution to Eq. 10. Unfortunately, the coordinate ascent algorithm given by Eq. 11 is only guaranteed to give a local optimum of $\mathcal{L}$, and we are forced to use multiple random initializations in an attempt to find the optimal q.

5.1 Variational Inference for the HDP

We now present details for variational inference of the HDP-HMM and HDP-aMMSB. Although each contains some model-specific details, there is also a great deal of overlap due to their similar use of the HDP. In both cases, we approximate the posterior $p(u, \pi, \phi, \mathcal{Z} \mid x)$ by a factorized distribution $q(\cdot) = q(u)q(\pi)q(\phi)q(\mathcal{Z})$, where $\mathcal{Z}$ denotes the model's hidden variables: $\mathcal{Z} = \{z\}$ for the HDP-HMM and $\mathcal{Z} = \{s, r\}$ for the HDP-aMMSB. We first give the overlapping treatment of these factors, followed by the model-specific details.

Factor $q(\mathcal{Z})$. Although the exact structure of $q(\mathcal{Z})$ is model-specific, both models similarly *truncate* the distribution. We define a pre-specified truncation-level, K, and constrain $q(\mathcal{Z})$ to place zero probability on hidden states or communities taking on values greater than K. We do not similarly restrict $q(u), q(\pi)$, and $q(\phi)$; however our truncation of $q(\mathcal{Z})$ implies that they are optimally set to their prior under $p(\cdot)$ beyond the truncation level K. While this technically leaves these factors as infinite distributions, we can represent them implicitly, rather than actually allocating infinite memory.

Factor $q(\pi)$. In both cases, we set $q(\pi)$ to be a product of Dirichlet distributions $q(\pi_i) = \text{Dirichlet}(\hat{\theta}_{i1}, \ldots, \hat{\theta}_{i>K})$, with $\hat{\theta}_{i\ell} > 0$. Although π_i is still an infinitely large vector, we motivate this finite parameterization by considering Eq. 8. We will later find it useful to define two summaries of this distribution:

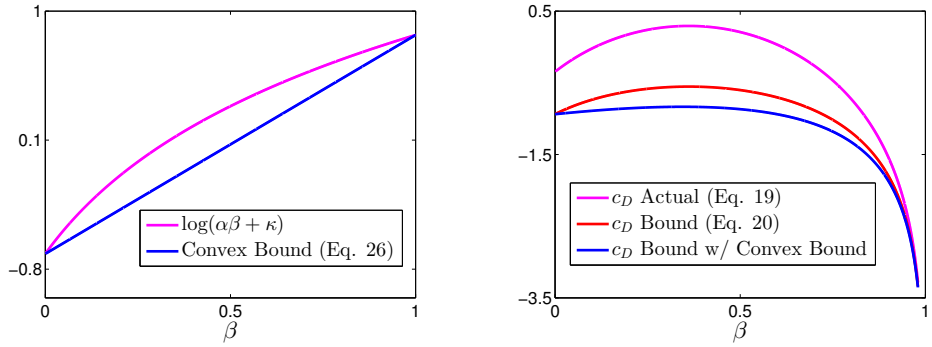

Figure 6: Figure showing plots of bounds in Eq. 20 and Eq. 26 for $\alpha = 1.5$, $\kappa = 0.5$, and a truncation level of $K = 1$. Note that the bound in Eq. 26 is very tight even for such an impractically small value of κ; the gap in the bound is visually unnoticable for more realistic values of κ.

$$\tilde{\pi}_{\ell m} \triangleq \exp \mathbb{E}_q[\log \pi_{\ell m}] = \exp\left[\psi(\hat{\theta}_{\ell m}) - \psi(\textstyle\sum_{m=1}^{K+1} \hat{\theta}_{\ell m})\right], \quad \tilde{\pi}_{\ell} \triangleq \sum_{m=1}^{K+1} \tilde{\pi}_{\ell m} \tag{13}$$

where ψ is the digamma function $\psi(x) \triangleq \frac{d}{dx} \log \Gamma(x)$. A direct application of Eq. 11 gives that $\hat{\theta}$ can be updated as:

$$\hat{\theta}_{\ell m} = \alpha \beta_m + U_{\ell m} \tag{14}$$

Where $U_{\ell m}$ is a model-specific *usage sufficient statistic* of $q(\mathcal{Z})$ interpretable as the expected number of times that $\pi_{\ell m}$ was "used" in generating the data. In the case of the HDP-HMM, it is the expected number of times the hidden state transitioned from ℓ to m; for the HDP-aMMSB, it is the expected number of times node ℓ was allocated to community m.

Factor $q(\phi)$. This factor is dependent on the emission density, F, and its prior, rather than choice of model or structure of the HDP. When F is a member of the exponential family, these updates have a standard form. As all of our emission densities F are standard members of the exponential family, we do not give further details here.

Factor $q(u)$. Much previous work on variational inference for HDP-based models used a point-mass for the distribution of the top-level stick, $q(u) = \delta_{u^*}(u)$ [5, 13, 15]. While computationally simple, this approach suffers from issues noted in [11], so we follow their approach of using a full distribution $q(u_\ell) = \text{Beta}(\hat{\rho}_\ell \hat{\omega}_\ell, (1 - \hat{\rho}_\ell)\hat{\omega}_\ell)$, where $\hat{\rho}_\ell \in (0, 1)$ and $\hat{\omega}_\ell > 0$.

This choice of distribution, however, forces us to numerically optimize $\mathcal{L}$ to recover the optimal parameters $\hat{\rho}, \hat{\omega}$. For both the HDP-aMMSB and basic HDP-HMM, details of this optimization are nearly identical to those presented in [11]. The addition of the sticky hyperparameter to the HDP-HMM introduces some changes, which are covered in Appendix B.

Objective Function. Variational methods seek to maximize a lower bound $\mathcal{L}$ on the marginalized log-likelihood, $\mathcal{L} \leq \log p(x)$. The objective function for both our models can be written as:

$$\begin{aligned} \mathcal{L} =& \mathbb{E}_q[\log p(x, u, \pi, \phi, \mathcal{Z}) - \log q(u, \pi, \phi, \mathcal{Z})] \\ =& \mathcal{L}_{data} + \mathcal{L}_{HDP} + \mathcal{L}_{alloc} + \mathbb{H}[q(\mathcal{Z})] \end{aligned} \tag{15}$$

where we define:

$$\mathcal{L}_{data} \triangleq \mathbb{E}_q \left[\log p(x \mid \mathcal{Z}, \phi) + \log \frac{p(\phi \mid \tau_1, \tau_0)}{q(\phi)} \right] \tag{16}$$

$$\mathcal{L}_{HDP} \triangleq \mathbb{E}_q \left[\log \frac{p(u \mid \gamma)}{q(u)} + \log \frac{p(\pi \mid u, \alpha)}{q(\pi)} \right] \tag{17}$$

$$\mathcal{L}_{alloc} \triangleq \mathbb{E}_q[\log p(\mathcal{Z} \mid \pi)] \tag{18}$$

and $\mathbb{H}[q(\mathcal{Z})]$ as the entropy of $q(\mathcal{Z})$. Here, we discuss the non-model-specific term $\mathcal{L}_{HDP}$. While most of its evaluation requires standard expectations of beta and Dirichlet distributions, difficulty arises in the term $\mathbb{E}_q[\log p(\pi)]$. The issue comes from the log-normalization constant, $c_D(\alpha\beta)$, of the log-Dirichlet distribution $\log p(\pi)$, where:

$$c_D(\alpha\beta) \triangleq \log \Gamma \left(\sum_{m=1}^{K+1} \alpha\beta_m \right) - \sum_{m=1}^{K+1} \log \Gamma(\alpha\beta_m) \tag{19}$$

In particular, $\mathbb{E}_q[c_D(\alpha\beta)]$, has no closed form due to our choice of $q(u)$. We place a lower bound on this term:

$$c_D(\alpha\beta) \geq K \log \alpha + \sum_{m=1}^{K+1} \log \beta_m, \tag{20}$$

a proof of which is given in Appendix A. The expectation of Eq. 20 has a closed form, giving us a tractable lower bound on $\mathcal{L}$.

5.2 Variational Inference for the HDP-HMM

Factor $q(z)$. We assume a form of $q(z)$ that retains the Markov dependencies from $p(z \mid \pi)$:

$$q(z) = q(z_1) \prod_{t=2}^{T} q(z_t \mid z_{t-1}) = \left[\prod_{\ell=1}^{K} \hat{r}_{1\ell}^{z_{1\ell}} \right] \left[\prod_{t=2}^{T} \prod_{\ell=1}^{K} \prod_{m=1}^{K} \left(\frac{\hat{s}_{t\ell m}}{\hat{r}_{t-1,\ell}} \right)^{z_{t-1,\ell} z_{tm}} \right], \tag{21}$$

where variational parameters $\hat{s}, \hat{r}$ parameterize the discrete distribution $q(z)$ by $\hat{s}_{t\ell m} \triangleq q(z_{t-1,\ell}, z_{tm})$ and $\hat{r}_{t\ell} \triangleq q(z_{t\ell})$. Note that, due to our truncation assumption, products run up to K rather than ∞. Following previous work on variational methods for HMMs [3], our update to $q(z)$ finds the joint configuration of all parameters $\hat{s}, \hat{r}$ that maximize our objective $\mathcal{L}$. To do so, we apply Eq. 11 to find that the optimal $q(z)$ satisfies:

$$q(z) \propto \left[\prod_{\ell=1}^{K} \tilde{\pi}_{0\ell}^{z_{1\ell}} \right] \left[\prod_{t=2}^{T} \prod_{\ell=1}^{K} \prod_{m=1}^{K} \tilde{\pi}_{\ell m}^{z_{t-1,\ell} z_{tm}} \right] \left[\prod_{t=1}^{T} \prod_{\ell=1}^{K} \exp\left(\mathbb{E}_q[\log p(x_t \mid \phi_\ell)] \right)^{z_{t\ell}} \right] \tag{22}$$

where $\tilde{\pi}_{\ell m}$ is as defined in Eq. 13. We can now use belief propagation – or more specifically, the forward-backward algorithm – to find the marginals $\hat{s}_{t\ell m} = q(z_{t-t,\ell}, z_{tm})$ and $\hat{r}_{t\ell} = q(z_{t\ell})$ of the distribution $q(z)$ satisfying this proportionality.

Factor $q(\pi)$. In the case of the HDP-HMM, $q(\pi)$ is a factor over an infinite number of distributions: $q(\pi) = \prod_{\ell=0}^{\infty} \text{Dirichlet}(\hat{\theta}_\ell)$; however, as noted in Section 5.1, our chosen truncation of $q(z)$ implies we need only update up to $\ell = K$, as terms with index $\ell > K$ will be set to their prior.

Following the generic update to $\hat{\theta}$ given in Eq. 14, we find that the usage sufficient statistics are given by:

$$U_{\ell m} = \begin{cases} \hat{r}_{1m} & \ell = 0 \\ \sum_{t=2}^{T} \hat{s}_{t\ell m} & \ell \neq 0 \end{cases} \tag{23}$$

where the case $\ell = 0$ corresponds to $\hat{\theta}_0$ parameterizing the factor over the starting-state distribution, $q(\pi_0)$. U_{0m} has the natural interpretation as the expected number of times the hidden state began in state m and $U_{\ell m}$ as the expected number of times the hidden state transitioned from state ℓ to m.

The sticky HDP-HMM requires only a slight modification to $q(\pi)$:

$$\hat{\theta}_{\ell m} = \alpha\beta_m + \kappa\, \mathbb{1}_{\{\ell=m\}} + U_{\ell m} \tag{24}$$

where $\mathbb{1}_{\{\ell=m\}} = 1$ if $\ell = m$ and 0 otherwise. Thus κ only has the effect of adding pseudocounts to the self-transition parameters $\hat{\theta}_{\ell\ell}$.

Objective Function. The addition of the sticky hyperparameter κ to the HDP-HMM requires some modifications to our lower bound on the term $\mathcal{L}_{HDP}$. With κ included, Eq. 20 becomes:

$$c_D(\alpha\beta + \kappa\, \mathbb{1}_{\{\ell\}}) \geq K \log\alpha - \log(\alpha+\kappa) + \log(\alpha\beta_\ell + \kappa) + \sum_{\substack{m=1 \\ m\neq\ell}}^{K+1} \log\beta_m \tag{25}$$

We are again left with an intractable expectation, as $\mathbb{E}_q[\log(\alpha\beta_\ell + \kappa)]$ has no closed form in terms of elementary functions[1]. To resolve this, we apply yet another lower bound that relies on the concavity of logarithms:

$$\log(\alpha\beta_\ell + \kappa) \geq \beta_\ell \log(\alpha+\kappa) + (1-\beta_\ell)\log\kappa \tag{26}$$

In practice, we either have $\kappa = 0$, in which case we do not apply the bound, or we have $\kappa > 100$, in which case the gap in the bound is completely negligible for reasonable values of α; a plot showing this is given in Figure 6.

5.3 Variational Inference for the HDP-aMMSB

Factor $q(s, r)$. The first variational algorithm proposed for the MMSB [1] factorized its hidden communities: $q(s, r) = q(s)q(r)$; however, due to issues with local optima that this factorization creates, we follow [14] and define a single factor $q(s, r)$. We parameterize this distribution by $\hat{\eta}$ such that $\hat{\eta}_{ij\ell m} \triangleq q(s_{ij\ell}, r_{ijm})$.

In opposition to the HDP-HMM, the latent factor for the HDP-aMMSB has a closed form update:

$$\hat{\eta}_{ij\ell m} \propto \begin{cases} \frac{1}{Z_{ij}}\tilde{\pi}_{ik}\tilde{\pi}_{jk} f(w_k, x_{ij}) & \ell = m = k \\ \frac{1}{Z_{ij}}\tilde{\pi}_{i\ell}\tilde{\pi}_{jm} f(\varepsilon, x_{ij}) & \ell \neq m \end{cases} \tag{27}$$

where $\tilde{\pi}_{i\ell}$ is defined in Eq. 13, $f(y, x_{ij}) \triangleq \exp[x_{ij}\mathbb{E}_q[\log(y)] + (1-x_{ij})\mathbb{E}_q[\log(1-y)]]$, and Z_{ij} is the normalization constant required to make $\hat{\eta}_{ij}$ a proper probability distribution. Here, our truncation implies that $q(s_{ij\ell}) = q(r_{ijm}) = 0$ for ℓ or $m > K$, giving $\hat{\eta}_{ij\ell m} = 0$ for $\ell, m > 0$.

Factor $q(\pi)$. For the HDP-aMMSB, $q(\pi)$ is a distribution over N infinitely large community membership vectors π_i. Again, by our truncation of $q(s, r)$, we need only update the first K components of each distribution.

A derivation of the usage statistics $U_{i\ell}$ gives:

[1] This expectation does have a closed form in terms of ${}_3F_2(\cdot)$, the regularized generalized hypergeometric function; however, our lower bound is computationally easier to work with and very tight in practice.

$$U_{i\ell} = \sum_{j=1}^{N} \mathbb{E}_q[s_{ij\ell}] + \sum_{j=1}^{N} \mathbb{E}_q[r_{ji\ell}]$$
$$= \sum_{j=1}^{N} \sum_{m=1}^{K} (\hat{\eta}_{ij\ell m} + \hat{\eta}_{jim\ell}) \tag{28}$$

Again, the usage statistic $U_{i\ell}$ has a natural interpretation as the expected number of times $\pi_{i\ell}$ was "used": it is the expected sum over how often node i was in community ℓ as either a source or a receiver.

Computational Efficiency Although the assortative MMSB is a slightly less expressive model than the full MMSB, it is significantly more computationally efficient. In both the assortative and full MMSB, the matrix $\hat{\eta}$ is of size $O(N^2K^2)$, which can be prohibitively expensive to compute and store. However, due to the fact that the aMMSB only has K "real" community interactions (i.e. those for which $\ell = m$), we can achieve linear complexity in K by only computing terms $\hat{\eta}_{ij\ell m}$ for which $\ell = m$. The details of how this changes the above computations are given in Appendix C.

6 Variational Inference Algorithms

Section 5 outlines the basic variational updates and objective function calculations for our chosen variational approximations. We now introduce specific algorithms that contain various interleavings and variations of these updates.

6.1 Single Batch Variational Inference

Single batch variational inference is the most immediate application of the results from Section 5. We consider the entire dataset at once, and iterate between updating "local" parameters $q(\mathcal{Z})$ and "global" parameters $q(\pi)$, $q(\phi)$, and $q(u)$. We see that the local step updates those factors which scale with the amount of observed data, while the global step scales with the number of hidden states K (albeit $q(\pi)$ scales with the number of nodes in the graph for the HDP-aMMSB). Intuitively, information only "flows" between local variables through the global parameters. This can make batch algorithms require many local steps to converge, which can be prohibitively expensive for a large dataset.

6.2 Memoized Online Variational Inference

To address this issue, we use *memoized online* variational inference [12], which divides the dataset into many smaller batches. After performing a local step on a single batch, we update the global parameters of the entire model and move on to the next batch. Although this does not change the runtime of a single pass through the dataset, it decreases the overall number of passes required by promoting more rapid exchange of information between batches.

Batch Definitions. To use memoized inference, we need a way of segmenting data into batches. For the HDP-HMM, we set each batch to be a collection of sequences, where a local step corresponds to running the forward-backward algorithm on each sequence in the batch. Although we do not do so here, it is possible to define batches as small subsets of larger sequences [7].

For the HDP-aMMSB, we consider a batch to be some subset of nodes and all their outgoing relationships. That is, for a set of indicies $I \subseteq \{1, \ldots, N\}$, a batch is all pairs $\{(i, j) : i \in I, j \in \{1, \ldots, N\}, j \neq i\}$. This definition is primarily for convenience of implementation; more sophisticated strategies that define batches as arbitrary subsets of edges have previously found success [10, 14].

Merge and Delete Moves. Although not discussed here, a major benefit of memoized algorithms is their ability to vary the truncation level K as the algorithm progresses [12, 11]. This often allows them to find higher quality and more compact models than competing methods.

7 Experiments

We test both of our models on a variety of datasets. Our experiments can be re-created using our open source code[2] and the hyperparameter settings given in Appendix D

7.1 HDP-HMM

Methods. We assess the quality of our learned model by how accurately it predicts hidden state sequences. To do so, we compute the Hamming distance between the model's estimated state sequence and the ground-truth labels. The Viterbi algorithm is used to find the most likely state sequence z under $q(\cdot)$, and we then use the Munkres algorithm to permute the estimated labels to those of the ground-truth.

We compare our memoized online algorithm (`memo`) against the blocked Gibbs sampler (`sampler`) of [9], as well as the batch version of our algorithm (`batch`). For each of these methods, we use a k-means initialization to find hard assignments for each of the z_t. We then construct local parameters $\hat{r}$, $\hat{s}$ consistent with these assignments and initialize the global parameters for $q(\pi)$ and $q(\phi)$ accordingly.

Toy Data. We test our algorithm on the toy dataset of Figure 1 containing $K = 8$ states with 2-dimensional gaussian emissions. The dataset consists of thirty-two sequences, each of length $T = 1000$. Exact parameters for generating the dataset are taken from [7].

Our results are shown in Figure 7. Given the simplicity of this dataset, it is not surprising that the best runs of all three methods eventually reach a nearly perfect hamming distance of 0. However, we see that our variational algorithms accomplish this task significantly faster than the sampler, particularly when using the sticky hyperparameter. Although we set a relatively small batch size of four for our memoized algorithm, we still find that, on average, it converges faster than single batch inference.

Speaker Diarization. We next examine twenty-one unrelated audio recordings with an unknown number of speakers from the NIST 2007 speaker diarization challenge[3]. The audio recordings are unrelated in the sense that they do not necessarily contain the same speakers. We thus train and measure performance on each meeting separately. Our main point of comparison is the HDP-HMM blocked Gibbs sampler of [9], which previously obtained state-of-the-art results on this dataset.

We run our algorithm with a gaussian likelihood starting from ten random k-means initializations and use the same initialization for the sampler. In this case, we do not compare to single batch inference, as each meeting consists of only one sequence. Our results are shown in Figure 8; we find that our algorithm delivers similar performance to the sampler with nearly two orders of magnitude fewer passes through the dataset. Interestingly, although [9] first introduced the sticky hyperparameter κ for this task, our algorithm performs best with $\kappa = 0$.

Human Motion Capture. Finally, we test our HDP-HMM on six sequences of motion capture data of humans exercising[4], manually segmented into twelve different activities ("knee-raise," "squats," etc.) [8]. Each datapoint x_t corresponds to measurements of twelve joint positions taken over a .1 second window. We use an autoregressive gaussian likelihood, $\kappa = 0$, 300, and initialize using the k-means algorithm with $K = 20$ states. Results are shown in Figure 9.

Although the use of the sticky hyperparameter removes some rapid state swtiching, our results still seem poor when immediately compared to the ground-truth segmentations: we recover a hamming distance of .46 when $\kappa = 0$ and .43 when $\kappa = 300$. However, we find that this is due to our algorithm recovering finer-grained states than those given by the ground-truth labels. For example, within the label "knee-raise," our algorithm discovers three states corresponding to left knee-raises, right knee-raises and a somewhat noisy transition / rest state. We see this pattern repeated in other exercises; the HDP-HMM segments the "twist" exercise into arm twirling and back-bending states.

[2] `https://bitbucket.org/michaelchughes/bnpy-dev/`

[3] `http://www.nist.gov/speech/tests/rt/`

[4] Data can be found at `https://github.com/michaelchughes/mocap6dataset`

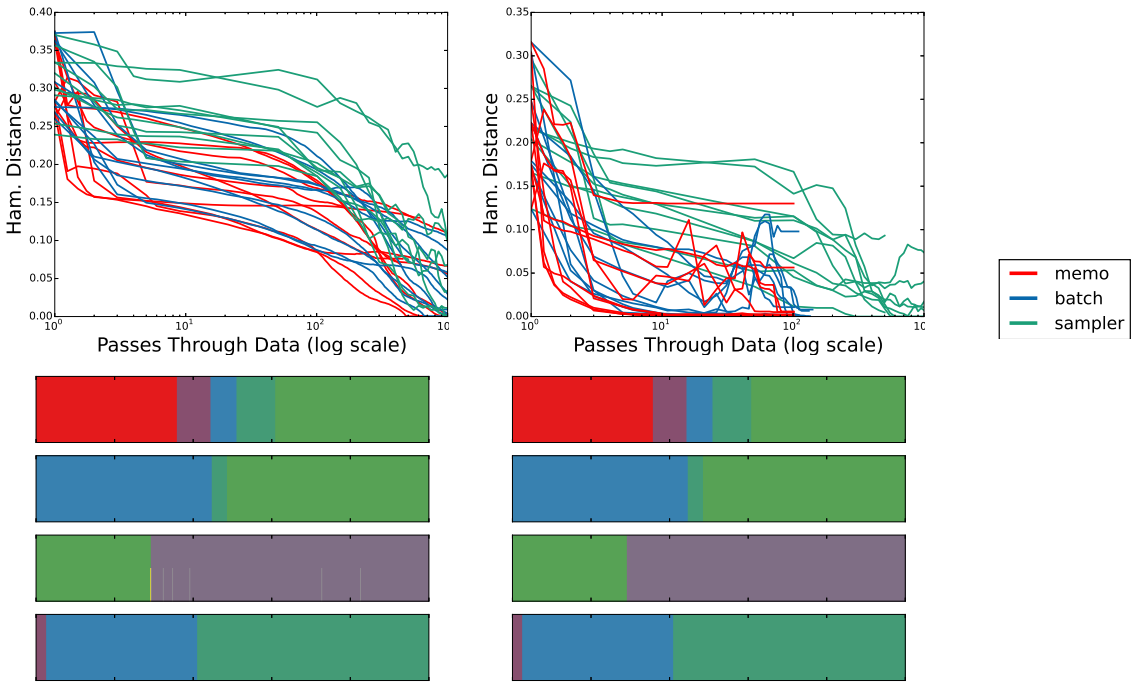

Figure 7: HMM toy dataset results. *Top*: Plot of hamming distance versus pass through data for the sampler and our single batch and memoized algorithms with $\kappa = 0$ (left) and $\kappa = 100$ (right). *Bottom*: Comparison of estimated state sequence labels (bottom half of each image) with true labels (top half of each image). Shown are four of the thirty-two estimated sequences from the best run of our memoized algorithm.

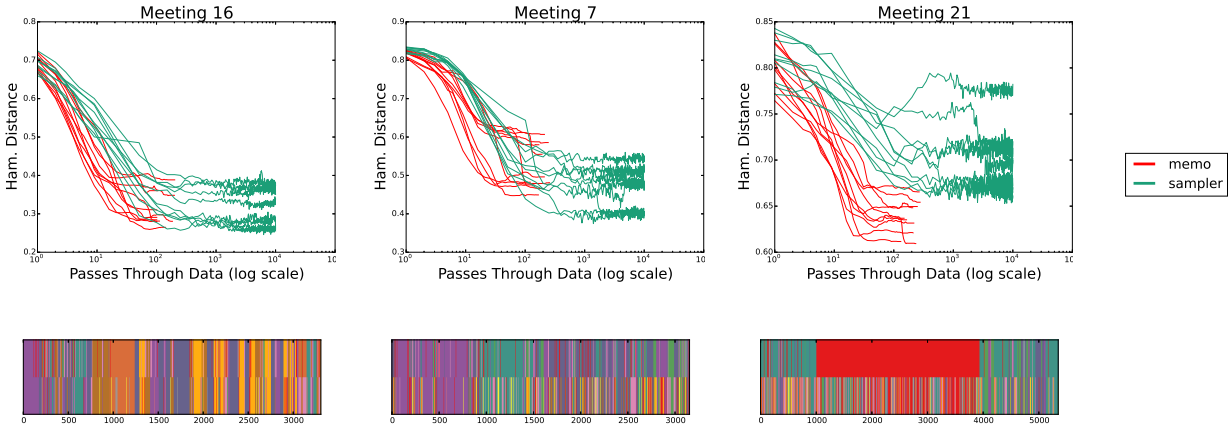

Figure 8: Speaker Diarization results. *Top*: Plots of hamming distance vs. pass through data for meetings 16, 7, and 21, representing best, intermediate and worst performance, respectively. *Bottom*: Comparison of estimated state sequence labels (bottom half of each image) with true labels (top half of each image).

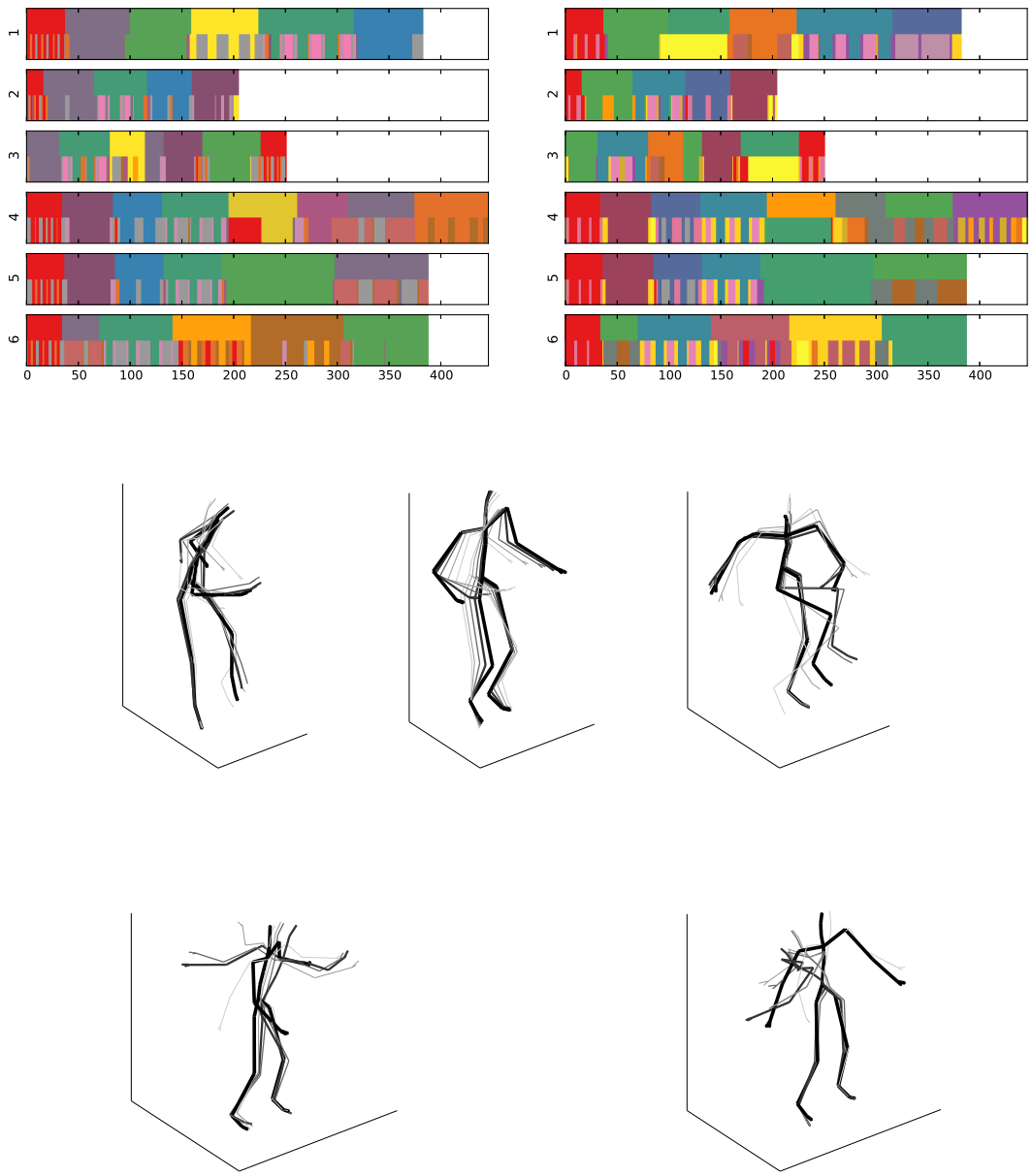

Figure 9: HDP-HMM results on the motion capture dataset. *Top Row*: Learned states of the runs with highest $\mathcal{L}$ for $\kappa = 0$ (left) and $\kappa = 300$ (right). True state sequence labels are shown at the top of each row and estimated on the bottom. *Middle Row*: Wire skeletons of the actual motion capture data showing our learned segmentation of the fourth activity ("knee-raises") in sequence five. We find that our forest-green state corresponds to right knee-raises (left), the brief yellow state to resting (center), and the pink state to right knee-raises (right). *Bottom Row*: Wire skeletons for the final activity in sequence five ("twist"). The HDP-HMM breaks this exercise into two distinct states: our gray state corresponds to horizontal arm-twirling (left), while our brown state is a back-bending motion (right).

7.2 HDP-aMMSB

Methods. Unlike the time-series datasets above, very few relational datasets are accompanied by ground-truth community assignments. To assess the quality of our learned models, we turn to held-out data prediction. Specifically, we hold out 10% of edges and a similar number of non-edges at training time. At various points during inference, we compute the probability of each held-out edge under our approximate posterior $q(\cdot)$:

$$\mathbb{E}_q[x_{ij}] = \mathbb{E}_q[\pi_i]^T \mathbb{E}_q[\phi] \mathbb{E}_q[\pi_j] \tag{29}$$

By thresholding this probability, we can compute the precision and recall of our model as:

$$\text{recall} = \frac{\text{Num. correct } x_{ij} = 1}{\text{True num. } x_{ij} = 1}, \quad \text{precision} = \frac{\text{Num. correct } x_{ij} = 1}{\text{Total predicted } x_{ij} = 1} \tag{30}$$

We compute this curve every twenty iterations and show its progression over the course of inference.

We initialize our model's global parameters by a random initialization for $q(\pi)$: $\hat{\theta}_{i\ell} \sim \text{Gamma}(5, 2)$ and set $q(\phi)$ equal to its prior under $p(\cdot)$. Finally, both of our experiments use the hyperparameter settings described in Appendix D.

Toy Data. We next test our model on the toy dataset from Figure 2 containing $K = 6$ communities with fairly sparse community memberships π_i sampled from Dirichlet$(.05)$. We divide the data into $B = 20$ batches, each containing 10 nodes, and hold out 10% of edges along with a similar number of non-edges. As shown in Figure 10, our algorithm recovers the structure of the graph with few errors.

Co-authorship Network. Finally, we train our model on a co-authorship network of physicists working in quantum cosmology and general relativity[5]. This network contains many disconnected subnetworks; we train on the largest, which contains $N = 4,158$ nodes and 26,850 edges. We run using $K = 20$ and $B = 300$ batches.

Our results shown in Figure 11 demonstrate that our model learns substantial communities within the graph; however, although we obtain high precision, our model fails to reach even 30% recall on the held-out interactions. This is due to the model's high edge probability within the "main" community (middle row of Figure 11) and low probability elsewhere. The main community contains just under a third of the observed edges, accounting for nearly all of our correctly recalled edges.

[5] `https://snap.stanford.edu/data/ca-GrQc.html`

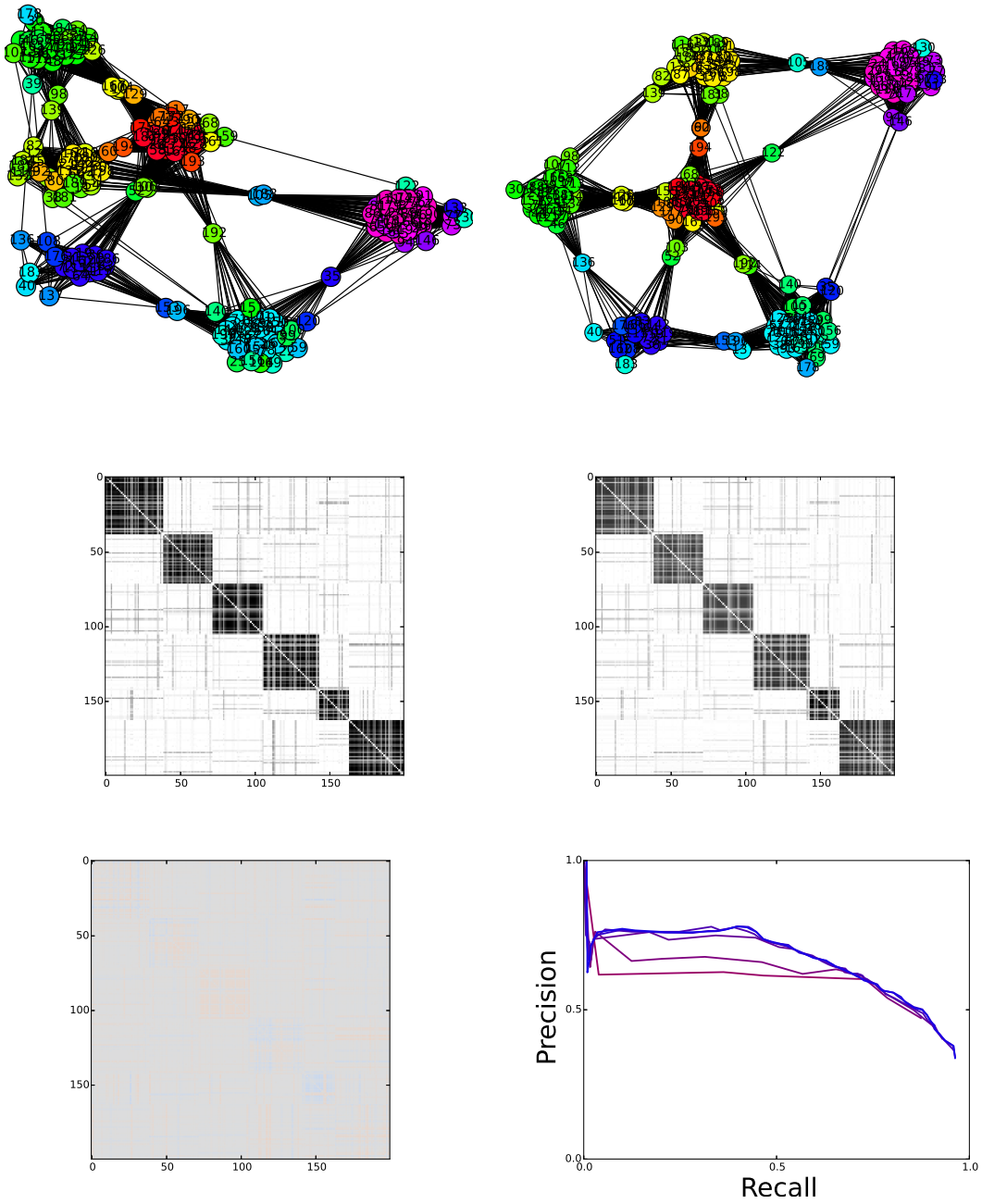

Figure 10: Results from the toy MMSB dataset of Figure 2. *Top Row:* Graph with edges drawn between nodes i and j if the probability of an edge $\pi_i^T \phi \pi_j$ is above a threshold using the true parameters π, ϕ (left) and the estimated parameters from our model with the best objective function $\mathcal{L}$ (right). *Middle Row:* Unthresholded adjacency matrix of the above graphs with node indices permuted by their most likely community. The main source of interaction is intra-community, represented by the strong block diagonal. Nodes with significant membership in more than one community give rise to the off-diagonal streaks. *Bottom Left*: Difference between the two adjacency matrices shown above. Red corresponds to too high an estimated probability and blue too low an estimate; the predominantly gray color indicates very little difference between the two. *Bottom Right*: Curves of precision vs. recall on held-out interactions evaluated every twenty iterations. Early iterations are colored red, while later iterations are shown in blue.

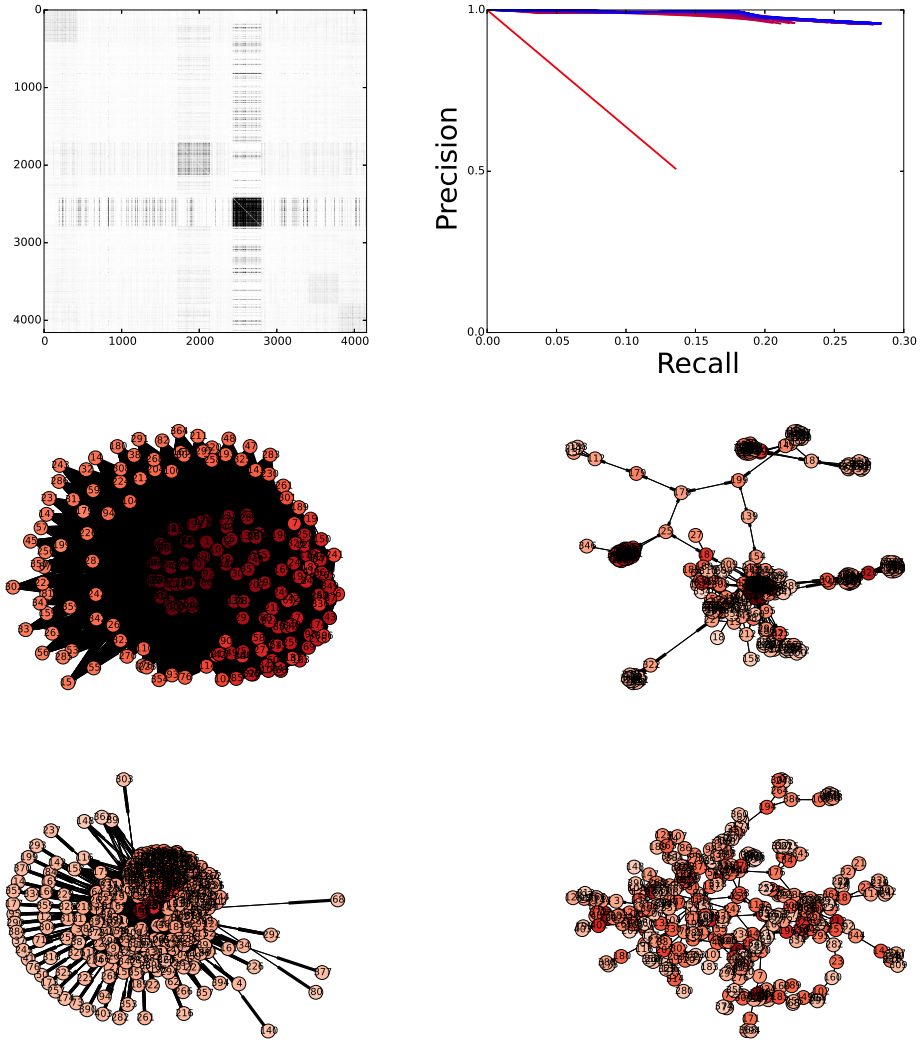

Figure 11: HDP-aMMSB results from the physics co-authorship network. *Top Left*: Matrix showing estimated edge probability between each pair of nodes, permuted by each node's most likely community. Five communities are faintly visible along the diagonal. *Top Right*: Precision vs. recall curves computed on held-out interactions. Red curves show earlier iterations, while blue curves show later iterations. *Middle Row*: Subgraphs of the most prominent community in the above edge probability matrix. Shown is the graph corresponding to the thresholded edge probability matrix (left) and actual observed adjacencies (right). Nodes belonging more strongly to this community are colored darker red. *Bottom Row*: Subgraphs of the second most prominent community (near center of the edge probability matrix). In both cases, we see a core of main community members is surrounded by less dedicated members.

8 Conclusions

We have developed memoized online variational inference algorithms for two nonparametric models: the HDP-HMM and HDP-aMMSB. We have shown their ability to recover structure in a wide variety of applications, even when the dimension of this structure is vague or unknown prior to inference. In the future, we hope to further improve our variational algorithms by adapting the work of [11], which varies the truncation level as inference progresses.

9 Acknowledgements

First and foremost, I would like to thank my advisor, Erik Sudderth, for taking so much time to give me help and guidance throughout the course of this project. I would also like to thank Michael Hughes, who, despite having the busy schedule of a graduate student, also helped me on countless occasions. Finally, I want to say thank you to the Brown Computer Science department for having so many amazing courses, faculty, and other students!

References

[1] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. 2009.

[2] Matthew J Beal, Zoubin Ghahramani, and Carl E Rasmussen. The infinite hidden markov model. 2001.

[3] Matthew James Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, University of London, 2003.

[4] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[5] Michael Bryant and Erik B. Sudderth. Truly nonparametric online variational inference for hierarchical Dirichlet processes. 2012.

[6] Thomas S Ferguson. A bayesian analysis of some nonparametric problems. 1(2):209–230, 1973.

[7] Nicholas Foti, Jason Xu, Dillon Laird, and Emily Fox. Stochastic variational inference for hidden Markov models. 2014.

[8] Emily B Fox, Michael C Hughes, Erik B Sudderth, and Michael I Jordan. Joint modeling of multiple time series via the beta process with application to motion capture segmentation. 8(3):1281–1313, 2014.

[9] Emily B Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. A sticky HDP-HMM with application to speaker diarization. 5(2A):1020–1056, 2011.

[10] Prem K Gopalan, Sean Gerrish, Michael Freedman, David M. Blei, and David M. Mimno. Scalable inference of overlapping communities. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2249–2257. Curran Associates, Inc., 2012.

[11] Michael C. Hughes, Dae Il Kim, and Erik B. Sudderth. Reliable and scalable variational inference for the hierarchical Dirichlet process. 2015.

[12] Michael C. Hughes and Erik B. Sudderth. Memoized online variational inference for Dirichlet process mixture models. 2013.

[13] Matthew J. Johnson and Alan S. Willsky. Stochastic variational inference for bayesian time series models. 2014.

[14] Dae Il Kim, Prem Gopalan, David Blei, and Erik Sudderth. Efficient online inference for bayesian nonparametric relational models. 2013.

[15] Percy Liang, Slav Petrov, Michael I Jordan, and Dan Klein. The infinite PCFG using hierarchical Dirichlet processes. 2007.

[16] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–286, February 1989.

[17] J. Sethuraman. A constructive definition of Dirichlet priors. 4:639–650, 1994.

[18] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. 101(476):1566–1581, 2006.

[19] Chuck Wooters and Marijn Huijbregts. Multimodal technologies for perception of humans. chapter The ICSI RT07s Speaker Diarization System, pages 509–519. Springer-Verlag, Berlin, Heidelberg, 2008.

A Derivation of $\mathcal{L}_{HDP}$ lower bound

A.1 Bound on cumulant function of Dirichlet

As in the main paper, we define the cumulant function c_D of the Dirichlet distribution as

$$c_D(\alpha\beta) = c_D(\alpha\beta_1, \alpha\beta_2, \ldots \alpha\beta_K, \alpha\beta_{K+1}) \triangleq \log\Gamma(\alpha) - \sum_{k=1}^{K+1} \log\Gamma(\alpha\beta_k) \tag{31}$$

where $\alpha > 0$ is a positive scalar, and $\beta = \{\beta_k\}_{k=1}^{K+1}$ is a vector of positive numbers that sum-to-one. The log-Gamma function $\log\Gamma(\cdot)$ has the following series representation[6] for scalar input $x > 0$:

$$-\log\Gamma(x) = \log x + \gamma x + \sum_{n=1}^{\infty}\left(\log\left(1+\frac{x}{n}\right) - \frac{x}{n}\right) \tag{32}$$

where $\gamma \approx .57721$ is the Euler-Mascheroni constant. Substituting this expansion into the definition of c_D, we find

$$\begin{aligned} c_D(\alpha\beta) = &-\log\alpha - \gamma\alpha - \sum_{n=1}^{\infty}\left(\log\left(1+\frac{\alpha}{n}\right) - \frac{\alpha}{n}\right) \\ &+ \sum_{k=1}^{K+1}\left[\log\alpha\beta_k + \gamma\alpha\beta_k + \sum_{n=1}^{\infty}\left(\log\left(1+\frac{\alpha\beta_k}{n}\right) - \frac{\alpha\beta_k}{n}\right)\right] \end{aligned} \tag{33}$$

Here, all the infinite sums are convergent. This allows some regrouping and cancellation to get:

$$\begin{aligned} c_D(\alpha\beta) = &-\log\alpha + \sum_{k=1}^{K+1}\log\alpha\beta_k \\ &+ \sum_{n=1}^{\infty}\left(\log\left(\prod_{k=1}^{K+1}\left(1+\frac{\alpha\beta_k}{n}\right)\right) - \log\left(1+\frac{\alpha}{n}\right)\right) \end{aligned} \tag{34}$$

Finally, via the binomial product expansion below, we see that the infinite sum is strictly positive.

$$\prod_{k=1}^{K+1}\left(1+\frac{\alpha\beta_k}{n}\right) = 1 + \sum_{k=1}^{K+1}\frac{\alpha\beta_k}{n} + \text{pos. const.} \quad \rightarrow \quad \prod_{k=1}^{K+1}\left(1+\frac{\alpha\beta_k}{n}\right) \geq \left(1+\frac{\alpha}{n}\right) \tag{35}$$

Thus, by simply leaving off the infinite sum from Eq. (35) we have a valid lower bound on $c_D(\cdot)$:

$$c_D(\alpha\beta) \geq -\log\alpha + \sum_{k=1}^{K+1}\log\alpha\beta_k \tag{36}$$

Expanding $\log\alpha\beta_k = \log\alpha + \log\beta_k$, we can further simplify to

$$c_D(\alpha\beta) \geq c_{sur}(\alpha,\beta) \triangleq K\log\alpha + \sum_{k=1}^{K+1}\log\beta_k \tag{37}$$

[6] http://mathworld.wolfram.com/LogGammaFunction.html

A.2 Bound on cumulant function with sticky hyperparameter.

Applying this bound to the case of the sticky HDP-HMM requires some further effort. Applying the above to this case gives an equation analogous to Eq. 37.

$$c_D(\alpha\beta_k + \delta_k\kappa) \geq K \log \alpha - \log(\alpha + \kappa) + \log(\alpha\beta_k + \kappa) + \sum_{\substack{m=1 \\ m \neq k}}^{K+1} \log(\beta_m) \tag{38}$$

Evaluating this term requires computing $\mathbb{E}_q[\log(\alpha\beta_k + \kappa)]$, which has no closed form in terms of elementary functions. Instead of calculating this directly, we use the concavity of logarithms to lower bound this term:

$$\begin{aligned} \log(\alpha\beta_k + \kappa) &\geq \beta_k \log(\alpha + \kappa) + (1 - \beta_k) \log(\kappa) \\ &= \beta_k \big(\log(\alpha + \kappa) - \log(\kappa) \big) + \log \kappa \end{aligned} \tag{39}$$

We justify this bound by noting that for even moderate κ (say, $\kappa > 10$), this inequality is very tight, as shown in Figure 6 in the main body of this thesis. Empirically, we find that κ almost always needs to be either zero, in which case we do not apply the bound, or in the low hundreds, in which case the gap in the bound is completely negligible.

Plugging this bound in Eq. 38, we find

$$\begin{aligned} c_D(\alpha\beta_k + \delta_k\kappa) \geq c_{sur-\kappa}(\alpha, \kappa, \beta, k) \triangleq K \log \alpha + \log(\kappa) - \log(\alpha + \kappa) + \sum_{\substack{m=1 \\ m \neq k}}^{K+1} \log(\beta_m) \\ + \beta_k \big(\log(\alpha + \kappa) - \log(\kappa) \big) \end{aligned} \tag{40}$$

This equation gives us a surrogate bound on the cumulant function for a single sticky transition vector. We next need to compute the sum of K sticky cumulant functions, plus one non-sticky cumulant function for the starting state.

Using our surrogate functions, we have

$$\begin{aligned} c_{sur}(\alpha, \beta) + \textstyle\sum_{k=1}^{K} c_{sur-\kappa}(\alpha, \kappa, \beta, k) = (K^2 + K) \log \alpha \\ + K(\log(\kappa) - \log(\alpha + \kappa)) \\ + \big(\log(\alpha + \kappa) - \log(\kappa) \big) \textstyle\sum_{k=1}^{K} \beta_k \\ + \textstyle\sum_{k=1}^{K+1} \log \beta_k + \sum_{k=1}^{K} \sum_{\substack{m=1 \\ m \neq k}}^{K+1} \log(\beta_m) \end{aligned} \tag{41}$$

In the last line, the first sum comes from the stating state's cumulant function, the second nested sum comes from the others. We can combine these two terms to find that

$$\begin{aligned} c_{sur}(\alpha, \beta) + \textstyle\sum_{k=1}^{K} c_{sur-\kappa}(\alpha, \kappa, \beta, k) = (K^2 + K) \log \alpha + K(\log(\kappa) - \log(\alpha + \kappa)) \\ + \big(\log(\alpha + \kappa) - \log(\kappa) \big) \textstyle\sum_{k=1}^{K} \beta_k \\ + \log \beta_{K+1} + K \textstyle\sum_{k=1}^{K+1} \log(\beta_k) \end{aligned} \tag{42}$$

Finally, we can rewrite these sums of surrogate cumulants in terms of u instead of β, since the transformation between them is deterministic. We find

$$\begin{aligned} c_{sur}(\alpha, \beta(u)) + \textstyle\sum_{k=1}^{K} c_{sur-\kappa}(\alpha, \kappa, \beta(u), k) = (K^2 + K) \log \alpha + K(\log(\kappa) - \log(\alpha + \kappa)) \\ + \big(\log(\alpha + \kappa) - \log(\kappa) \big) \textstyle\sum_{k=1}^{K} \beta_k(u) \\ + \textstyle\sum_{k=1}^{K} \big(K \log u_k + [K(K + 1 - k) + 1] \log(1 - u_k) \big) \end{aligned} \tag{43}$$

We can now easily compute expectations of Eq. 43, since $\mathbb{E}_q[\beta_k]$ and $\mathbb{E}_q[\log u_k]$ have known closed forms for $q(u_k) = \text{Beta}(\hat{\rho}_k\hat{\omega}_k, (1 - \hat{\rho}_k)\hat{\omega}_k)$.

B Global update for $q(u)$

Here, we derive the results needed to perform numerical optimization of $\hat{\rho}$ and $\hat{\omega}$, the variational beta parameters of the top-level stick-breaking weights $q(u_k) = \text{Beta}(\hat{\rho}_k\hat{\omega}_k, (1-\hat{\rho}_k)\hat{\omega}_k)$. We only show the results with the sticky hyperparameter; optimization for the case of $\kappa = 0$ and for the HDP-aMMSB is identical to the case for the HDP topic model presented in [11]. For all equations from that paper, you need only substitute in $K+1$ or N, respectively, for the number of documents D.

To begin our derivation, we collect terms in $\mathcal{L}_{HDP}$ that depend on $\hat{\rho}$ and $\hat{\omega}$. Note that we have dropped any additive terms constant with respect to $\hat{\rho}, \hat{\omega}$ in this expression, since they have no bearing on our numerical optimization problem.

$$\begin{aligned}\mathcal{L}_{obj}(\hat{\rho},\hat{\omega}) = \sum_{k=1}^{K}\Bigg(& - c_B(\hat{\rho}_k\hat{\omega}_k, (1-\hat{\rho}_k)\hat{\omega}_k) \\ & + \big(K+1-\hat{\rho}_k\hat{\omega}_k\big)\big(\psi(\hat{\rho}_k\hat{\omega}_k) - \psi(\hat{\omega}_k)\big) \\ & + \big(K(K+1-k)+1+\gamma-(1-\hat{\rho}_k)\hat{\omega}_k\big)\big(\psi((1-\hat{\rho}_k)\hat{\omega}_k) - \psi(\hat{\omega}_k)\big)\Bigg) \\ + \sum_{\ell=1}^{K}\mathbb{E}_q[\beta_\ell]\Bigg(&\log(\alpha+\kappa) - \log\kappa + \sum_{k=0}^{K}\alpha_k P_{k\ell}(\hat{\theta})\Bigg) \\ + \mathbb{E}_q[\beta_{K+1}]\Bigg(&\sum_{k=0}^{K}\alpha_k P_{k,K+1}(\hat{\theta})\Bigg)\end{aligned} \tag{44}$$

B.1 Constrained Optimization Problem

Our goal is to find the $\hat{\rho}, \hat{\omega}$ that maximize $\mathcal{L}_{obj}$. Remember that $\hat{\rho}, \hat{\omega}$ parameterize K Beta distributions, and so have certain positivity constraints. Thus, we need to solve a *constrained* optimization problem:

$$\begin{aligned}\text{argmax}_{\hat{\rho},\hat{\omega}} \quad & \mathcal{L}_{obj}(\hat{\rho},\hat{\omega}) \\ \text{subject to} \quad & 0 < \hat{\rho}_k < 1 \quad \hat{\omega}_k > 0, \quad k = 1,\ldots,K\end{aligned} \tag{45}$$

We now give the expressions for the gradient of the objective $\nabla\mathcal{L}_{obj}$, with respect to each entry of $\hat{\omega}$ and $\hat{\rho}$.

Gradient for $\hat{\omega}$ Taking the derivative of Eq. 44 with respect to each entry $\hat{\omega}_m$ of $\hat{\omega}$, for $m \in 1, 2, \ldots K$, is:

$$\begin{aligned}\frac{\partial\mathcal{L}_{obj}}{\partial\hat{\omega}_m} = & \big(K+1-\hat{\rho}_m\hat{\omega}_m\big)\big(\hat{\rho}_m\psi_1(\hat{\rho}_m\hat{\omega}_m) - \psi_1(\hat{\omega}_m)\big) \\ & + \big(K(K+1-m)+1+\gamma-(1-\hat{\rho}_m)\hat{\omega}_m\big)\big((1-\hat{\rho}_m)\psi_1((1-\hat{\rho}_m)\hat{\omega}_m) - \psi_1(\hat{\omega}_m)\big)\end{aligned} \tag{46}$$

where $\psi_1 \triangleq \frac{d^2}{dx^2}\log\Gamma(x)$ is the trigamma function.

Gradient for $\hat{\rho}$ We first define Δ as a $K \times K+1$ matrix of partial derivatives of $\mathbb{E}_q[\beta_k]$ with respect to $\hat{\rho}$:

$$\Delta_{mk} \triangleq \frac{\partial}{\partial\hat{\rho}_m}\mathbb{E}_q[\beta_k] = \begin{cases} -\frac{1}{1-\hat{\rho}_m}\mathbb{E}_q[\beta_k] & m < k \\ \frac{1}{\hat{\rho}_m}\mathbb{E}_q[\beta_k] & m = k \\ 0 & m > k \end{cases} \tag{47}$$

Now, the derivative of $\mathcal{L}_{obj}$ with respect to each entry $\hat{\rho}_m$ of the vector $\hat{\rho}$, for $m \in 1, 2, \dots K$, is:

$$\begin{aligned}\frac{\partial \mathcal{L}_{obj}}{\partial \hat{\rho}_m} &= \hat{\omega}_m\big(K+1-\hat{\rho}_m\hat{\omega}_m\big)\psi_1(\hat{\rho}_m\hat{\omega}_m) \\ &\quad - \hat{\omega}_m\big(K(K+1-m)+1+\gamma-(1-\hat{\rho}_m)\hat{\omega}_m\big)\psi_1((1-\hat{\rho}_m)\hat{\omega}_m) \\ &\quad + \textstyle\sum_{\ell=1}^{K}\Delta_{m\ell}\Big(\log(\alpha+\kappa)-\log\kappa+\sum_{k=0}^{K}\alpha_k P_{k\ell}(\hat{\theta})\Big) \\ &\quad + \Delta_{m,K+1}\big(\sum_{k=0}^{K}\alpha_k P_{k,K+1}(\hat{\theta})\big)\end{aligned} \tag{48}$$

B.2 Unconstrained Optimization Problem

In practice, we find that it is more numerically stable to first transform our constrained optimization problem above into an unconstrained problem, and then solve the unconstrained problem via a modern gradient descent algorithm (L-BFGS).

Our transformation follows exactly the steps outlined in the Supplement of [11], where the cosntrained objective for the HDP topic model was transformed into unconstrained variables. See that Supplement document for mathematical details, or our public source code[7] for practical details.

C Efficient updates for the HDP-aMMSB

This section provides details of how to compute updates for the HDP-aMMSB in $O(N^2K)$ time, as opposed to the naive $O(N^2K^2)$ time. This method was first noted in [14]; we provide a single, cohesive explanation here. The key observation is that we only need to compute the N^2K diagonal entries of $\hat{\eta}$ using Eq. 27; that is, we only compute and store $\hat{\eta}_{ij\ell m}$ for $\ell = m$. We also need to compute and store Z_{ij}, the normalization constant that makes $\hat{\eta}_{ij}$ sum to 1. We can do so in $O(K)$ time by noting:

$$\begin{aligned}Z_{ij} \triangleq \sum_{\ell,m}^{K}\hat{\eta}_{ij\ell m} &= \sum_{k}^{K}\tilde{\pi}_{ij}\tilde{\pi}_{jk}f(\phi_k, x_{ij}) + \sum_{\ell\neq m}^{K}\tilde{\pi}_{i\ell}\tilde{\pi}_{jm}f(\varepsilon, x_{ij}) \\ &= \sum_{k}^{K}\tilde{\pi}_{ik}\tilde{\pi}_{jk}\Big(f(\phi_k, x_{ij}) - f(\varepsilon, x_{ij})\Big) + \tilde{\pi}_i\tilde{\pi}_j f(\varepsilon, x_{ij})\end{aligned} \tag{49}$$

where we have substituted in Eq. 27 in for $\hat{\eta}_{ij\ell m}$. This restricted computation of $\hat{\eta}$ makes updates to $q(\pi)$ and computation of $\mathcal{L}$ slightly more complicated. We outline the relevant terms below. A helpful identity that follows from the definition of $\hat{\eta}$ and $\tilde{\pi}$ is:

$$\sum_{\substack{m \\ m\neq\ell}}^{K}\hat{\eta}_{ij\ell m} = \frac{1}{Z_{ij}}f(\varepsilon, x_{ij})\tilde{\pi}_{i\ell}(\tilde{\pi}_j - \tilde{\pi}_{j\ell}) \tag{50}$$

Updates to $q(\pi)$. From Eq. 28, we have that the usage sufficient statistic for the HDP-aMMSB is $U_{i\ell} = \sum_{j=1}^{N}\sum_{m=1}^{K}(\hat{\eta}_{ij\ell m} + \hat{\eta}_{jim\ell})$ We can compute $U_{i\ell}$ given only the diagonal $\hat{\eta}_{ij\ell\ell}$ and normalization constants by:

$$U_{i\ell} = \sum_{j}^{N}\Big(\hat{\eta}_{ij\ell\ell} + \frac{1}{Z_{ij}}\tilde{\pi}_{i\ell}(\tilde{\pi}_j - \tilde{\pi}_{j\ell})\Big) + \sum_{j}^{N}\Big(\hat{\eta}_{ji\ell\ell} + \frac{1}{Z_{ji}}\tilde{\pi}_{i\ell}(\tilde{\pi}_j - \tilde{\pi}_{j\ell})\Big) \tag{51}$$

[7] https://bitbucket.org/michaelchughes/bnpy

Term $\mathbb{H}[q(s,r)]$. The entropy $\mathbb{H}[q(s,r)]$ corresponds to the term $\mathbb{E}_q[\log q(s,r)]$ in $\mathcal{L}$. We calculate it by:

$$\begin{aligned}
\mathbb{H}[q(s,r)] &\triangleq \sum_{i,j}^{N}\sum_{\ell,m}^{K} \hat{\eta}_{ij\ell m} \log \hat{\eta}_{ij\ell m} \\
&= \sum_{i,j}^{N}\sum_{\ell,m}^{K} \hat{\eta}_{ij\ell m} \Big(\mathbb{E}_q[\log \pi_{i\ell}] + \mathbb{E}_q[\log \pi_{jm}] + \log f(\cdot, x_{ij}) - \log Z_{ij} \Big) \\
&= \sum_{i,j} \Bigg[\sum_{\ell}^{K} \mathbb{E}_q[\log \pi_{i\ell}] \sum_{m}^{K} \hat{\eta}_{ij\ell m} + \sum_{m}^{K} \mathbb{E}_q[\log \pi_{jm}] \sum_{\ell}^{K} \hat{\eta}_{ij\ell m} \\
&\qquad + \log f(\varepsilon, x_{ij}) + \sum_{k}^{K} \hat{\eta}_{ijkk} \Big(\log f(\phi_k, x_{ij}) - \log f(\varepsilon, x_{ij}) \Big) - \log Z_{ij} \Bigg] \\
&= \sum_{i,j}^{N} \Bigg[\log f(\varepsilon, x_{ij}) + \sum_{k}^{K} \hat{\eta}_{ijkk} \Big(\log f(\phi_k, x_{ij}) - \log f(\varepsilon, x_{ij}) \Big) - \log Z_{ij} \qquad (52) \\
&\qquad + \sum_{\ell}^{K} \log \tilde{\pi}_{i\ell} \Big(\hat{\eta}_{ij\ell\ell} + \frac{1}{Z_{ij}} \tilde{\pi}_{i\ell} f(\varepsilon, x_{ij}) (\tilde{\pi}_j - \tilde{\pi}_{j\ell}) \Big) \qquad (53) \\
&\qquad + \sum_{m}^{K} \log \tilde{\pi}_{jm} \Big(\hat{\eta}_{ijmm} + \frac{1}{Z_{ij}} \tilde{\pi}_{jm} f(\varepsilon, x_{ij}) (\tilde{\pi}_i - \tilde{\pi}_{im}) \Big) \Bigg] \qquad (54)
\end{aligned}$$

We can use the previously cached statistic $U_{i\ell}$ by noting the similarity between Eq. 53 -54 and Eq. 51:

$$\begin{aligned}
\mathbb{H}[q(s,r)] = \sum_{i,j}^{N} &\Bigg[\log f(\varepsilon, x_{ij}) + \sum_{k}^{K} \hat{\eta}_{ijkk} \Big(\log f(\phi_k, x_{ij}) - \log f(\varepsilon, x_{ij}) \Big) - \log Z_{ij} \Bigg] \qquad (55) \\
&+ \sum_{\ell}^{K} \log \tilde{\pi}_{i\ell} U_{i\ell} \qquad (56)
\end{aligned}$$

D Hyperparameter Settings for Experiments

The hyperparameter settings given here can be used with our source code[8] to reproduce the experiments in Section 7 of the main document.

D.1 HDP-HMM: Toy Dataset

We use a full-covariance Gaussian likelihood.

−-gamma 10
–alpha 0.5
–startAlpha 5
–stickyKappa 0 or 100
–nu D+2
–ECovMat eye
–sF 1.0
–kappa 1e-5

[8] https://bitbucket.org/michaelchughes/bnpy-dev/

D.2 HDP-HMM: Speaker Diarization

We use a full covariance Gaussian likelihood.

−-gamma 10
–alpha 1.0
–startAlpha 1.0
–stickyKappa 0
–nu 1000
–ECovMat covdata
–sF 0.75
–kappa 0.0001

D.3 HDP-HMM: Motion Capture

We use a first order AR Gaussian likelihood.

−-gamma 10
–alpha 0.5
–startAlpha 5
–stickyKappa 300
–nu D+2
–ECovMat diagcovfirstdiff
–sF 0.5
–VMat same
–sV 0.5
–MMat eye

D.4 HDP-aMMSB Experiments

Both of our HDP-aMMSB experiments use a 1-D bernouli likelihood with a beta prior Beta(λ_1, λ_0) and identical hyperparameter settings:

−-gamma 10
–alpha 1.0
–epsilon 1e-5
–lam1 10.0
–lam0 0.0