

Center for Brains, Minds & Machines

arXiv:1411.7706v1 [stat.ML] 27 Nov 2014

CBMM Memo No. 027

February 8, 2022

A Nonparametric Bayesian Approach to Uncovering Rat Hippocampal Population Codes During Spatial Navigation

by

Scott W. Linderman, Matthew J. Johnson, Matthew A. Wilson, and Zhe Chen

Abstract: Rodent hippocampal population codes represent important spatial information about the environment during navigation. Several computational methods have been developed to uncover the neural representation of spatial topology embedded in rodent hippocampal ensemble spike activity. Here we extend our previous work and propose a nonparametric Bayesian approach to infer rat hippocampal population codes during spatial navigation. To tackle the model selection problem, we leverage a nonparametric Bayesian model. Specifically, to analyze rat hippocampal ensemble spiking activity, we apply a hierarchical Dirichlet process-hidden Markov model (HDP-HMM) using two Bayesian inference methods, one based on Markov chain Monte Carlo (MCMC) and the other based on variational Bayes (VB). We demonstrate the effectiveness of our Bayesian approaches on recordings from a freely-behaving rat navigating in an open field environment. We find that MCMC-based inference with Hamiltonian Monte Carlo (HMC) hyperparameter sampling is flexible and efficient, and outperforms VB and MCMC approaches with hyperparameters set by empirical Bayes.



This work was supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216.

1 Introduction

A fundamental goal in neuroscience is to understand how populations of neurons represent and transmit information about the external world. The hippocampus is known to encode information relevant to spatial navigation and episodic memory. Spatial representation of the environment is pivotal for navigation in rodents (O’Keefe and Nadel, 1978). One type of spatial representation is a topological map, which contains only relative ordering or connectivity information between spatial locations and is invariant to orientation or deformation. A relevant question of interest is: how can neurons downstream of the hippocampus infer representations of space from hippocampal spike activity without *a priori* place field information (namely, without the measurement of spatial correlates)? Several reports have been dedicated to the mathematical analysis of this problem (Curto and Itskov, 2008; Dabaghian et al., 2012); however, a data-driven approach for analyzing ensemble hippocampal spike data remains missing. This paper employs probabilistic modeling and inference methods to uncover the spatial representation (or topological map) based on the ensemble spike activity.

Bayesian statistical modeling is a consistent and principled framework for dealing with uncertainties about the observed data (Scott, 2002). The goal of Bayesian inference is to incorporate prior knowledge and constraints of the problem and to infer the posterior distribution of unobserved variables of interest (Gelman et al., 2013). In recent years, cutting-edge Bayesian methods have become increasingly popular for data analyses in neuroscience, medicine and biology (Mishchenko et al., 2011; Chen et al., 2011; Chen, 2013; Davidson et al., 2009; Kloosterman et al., 2014; Yau et al., 2011). Specifically, thanks to ever-growing computing power, Markov chain Monte Carlo (MCMC) methods have been widely used in Bayesian inference.

In our previous work (Chen et al., 2012a; 2014), we have developed a *parametric Bayesian* approach to uncover the neural representation of spatial topology embedded in rodent hippocampal population codes during spatial navigation. Here we extend the preceding work and consider a *nonparametric Bayesian* approach. The nonparametric Bayesian method brings additional flexibility to the probabilistic model, which allows us to model the complex structure of neural data (Teh and Jordan, 2010; Wood and Black, 2008; Yu et al., 2009; Shalchyan and Farina, 2014). Specifically, we leverage the so-called a hierarchical Dirichlet process-HMM (HDP-HMM) (Teh et al., 2006), which extends the finite-state hidden Markov model (HMM) with a nonparametric, HDP prior and derive corresponding Bayesian inference algorithm. We consider both deterministic and stochastic approaches for fully Bayesian inference. Based on deterministic approximation, we extend the work of (Chen et al., 2012a; Johnson and Willsky, 2014) and use a variational Bayes (VB) method for approximate Bayesian inference. For MCMC, we adapt a Gibbs sampling method (Teh et al., 2006; Johnson, 2014), and integrate it with a Hamiltonian Monte Carlo (HMC) method for hyperparameter inference (Neal, 2010). To the best of our knowledge, the application of the HDP-HMM to hippocampal ensemble neuronal spike trains and the HMC hyperparameter inference algorithm is novel.

We test the statistical model and inference methods with both simulation data and experimental data. The latter consists of a recording of rat dorsal hippocampal ensemble spike activity during open field navigation. Using a decoding analysis and predictive likelihood, we verify and compare the performance of the proposed Bayesian inference algorithms. We also discuss the results of model selection related to the sample size and the choice of concentration parameter or hyperparameters. Our methods provide an extended tool to analyze rodent hippocampal population codes, which may further empower us to explore important neuroscience questions about neural representation, learning and memory.

2 Methods: Modeling and Inference

2.1 Basic Probabilistic Model

In our previous work (Chen et al., 2012a), we used a finite m -state HMM to characterize the population spiking activity from a population of C hippocampal place cells. It was assumed that first, the animal’s spatial location during locomotion, modeled as a latent state process, followed a first-order discrete-state Markov chain $\mathcal{S} = S_{1:T} \equiv \{S_t\} \in \{1, \dots, m\}$, and second, the spike counts of individual place cells at time t , conditional on the hidden state S_t , followed a Poisson probability with their respective tuning curve functions $\mathbf{\Lambda} = \{\boldsymbol{\lambda}_c\} = \{\{\lambda_{c,i}\}\}$. Essentially,

we employed a Markov-driven population Poisson firing model with the following probabilistic models

$$\begin{aligned}
p(\mathbf{y}_{1:T}, S_{1:T} | \boldsymbol{\pi}, \mathbf{P}, \boldsymbol{\Lambda}) &= p(S_1 | \boldsymbol{\pi}) \prod_{t=2}^T p(S_t | S_{t-1}, \mathbf{P}) \prod_{t=1}^T p(\mathbf{y}_t | S_t, \boldsymbol{\Lambda}), \\
p(S_1 | \boldsymbol{\pi}) &= \text{Multinomial}(S_1 | \boldsymbol{\pi}), \\
p(S_t | S_{t-1}, \mathbf{P}) &= \text{Multinomial}(S_t | \mathbf{P}_{S_{t-1},:}), \\
p(\mathbf{y}_t | S_t, \boldsymbol{\Lambda}) &= \prod_{c=1}^C \text{Poisson}(y_{c,t} | \lambda_{c,S_t}).
\end{aligned} \tag{1}$$

where $\mathbf{P} = \{P_{ij}\}$ denotes an m -by- m state transition probability matrix, with P_{ij} representing the transition probability from state i to j (since $\sum_{k=1}^m P_{ik} = 1$, each row of matrix \mathbf{P} specifies a multinomial likelihood); $y_{c,t}$ denotes the number of spike counts from the c -th cell within the t -th temporal bin (for notation simplicity, from now on we assume $\Delta = 1$ such that the rate is defined in the unit bin size of 250 ms) and $\mathbf{y}_{1:T} = \{y_{c,t}\}_{C \times T}$ denotes time series of C -dimensional population response vector; and $\text{Poisson}(y_{c,t} | \lambda_{c,i})$ defines a Poisson distribution with the rate parameter $\lambda_{c,i}$ when $S_t = i$. Finally, $\log p(\mathbf{y}_{1:T} | \mathcal{S}, \boldsymbol{\theta})$ defines the observed data log likelihood given the hidden state sequence \mathcal{S} and all parameters $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \mathbf{P}, \boldsymbol{\Lambda}\}$ (where $\boldsymbol{\pi} = \{\pi_i\}$ denotes a probability vector for the initial state S_1).

The hidden variables $\mathcal{S} = \{S_{1:T}\}$ are treated as the missing data, $\mathbf{y}_{1:T}$ as the observed (incomplete) data, and their combination $\{S_{1:T}, \mathbf{y}_{1:T}\}$ as the complete data.

A Bayesian version of this model introduces prior distributions over the parameters. We use the following conjugate prior distributions,

$$\begin{aligned}
\boldsymbol{\pi} &\sim \text{Dir}(\alpha_0 \mathbf{1}), \\
\mathbf{P}_{i,:} &\sim \text{Dir}(\alpha_0 \mathbf{1}), \\
\lambda_{c,i} &\sim \text{Gamma}(a_c^0, b_c^0).
\end{aligned} \tag{2}$$

where Dir denotes the Dirichlet prior distribution, and $\text{Gamma}(a_c^0, b_c^0)$ denotes the gamma prior distribution with shape parameter a_c^0 and scale parameter b_c^0 .

2.2 HDP-HMM

Model selection is an important issue for statistical modeling and data analysis. We have previously proposed a *Bayesian deviance information criterion* to select the model size m of HMM (Chen et al., 2012a; 2014). Here we extend the finite-state HMM to an HDP-HMM, a nonparametric Bayesian extension of the HMM that allows for a potentially infinite number of hidden states (Beal et al., 2002). Namely, the HDP-HMM treats the priors via a stochastic process. Instead of imposing a Dirichlet prior distribution on the rows of the finite state transition matrix \mathbf{P} , we use a HDP that allows for a countably infinite number of states.

Specifically, we sample a distribution over latent states, G_0 , from a Dirichlet process (DP) prior, $G_0 \sim \text{DP}(\gamma, H)$, where γ is the concentration parameter and H is the base measure. One may sample from the DP via the “stick-breaking construction.” First we sample the stick-breaking weights, $\boldsymbol{\beta}$,

$$\tilde{\beta}_i \sim \text{Beta}(1, \gamma), \quad \beta_i = \tilde{\beta}_i, \quad \beta_i = \tilde{\beta}_i \prod_{k=1}^{j-1} (1 - \tilde{\beta}_k) \tag{3}$$

where $\sum_{j=1}^{\infty} \beta_j = 1$, and $\text{Beta}(a, b)$ defines a beta distribution with two positive shape parameters a and b .

The stick-breaking construction of (3) is generally denoted as $\boldsymbol{\beta} \sim \text{GEM}(\gamma)$, after Griffiths, Engen, and McCloskey (Ewens, 1990). The name “stick-breaking” comes from the interpretation of β_i as the length of the piece of a unit-length stick assigned to the i -th value. After the first $i - 1$ values having their portions assigned, the length of the remainder of the stick is broken according to a sample $\tilde{\pi}_i$ from a beta distribution, and $\tilde{\beta}_i$ indicates the portion of the remainder to be assigned to the i -th value. Therefore, the stick-breaking process $\text{GEM}(\gamma)$ also defines a DP—the smaller γ , the less (in a statistical sense) of the stick will be left for subsequent values.

After sampling $\boldsymbol{\beta}$, we next sample the latent state variables, in this case $\boldsymbol{\lambda}_c$, from the base measure H . Our draw from the $\text{DP}(\gamma, H)$ prior is then given by

$$G_0 = \sum_{j=1}^{\infty} \beta_j \delta_{\boldsymbol{\lambda}_c^{(j)}}. \tag{4}$$

Importantly, this distribution is discrete with probability one (Teh et al., 2006).

Given a countably infinite set of shared states, we may then sample the rows of the transition matrix, $\mathbf{P}_{i,:} \sim \text{DP}(\alpha_0, \boldsymbol{\beta})$. We place the same prior over $\boldsymbol{\pi}$. The base measure in this case is $\boldsymbol{\beta}$, a countably infinite vector of stick-breaking weights, that serves as the mean of the DP prior over the rows of \mathbf{P} . The concentration parameter, α_0 , governs how concentrated the rows are about the mean. Since the base measure $\boldsymbol{\beta}$ is discrete, each row of \mathbf{P} will be able to “see” the same set of states. By contrast, if we remove the HDP prior and treat each row of \mathbf{P} as an independent draw from a DP with base measure H , each row would see a disjoint set of states with probability one. In other words, the hierarchical prior is required to provide a discrete (but countably infinite) set of latent states for the HMM.

2.3 Overdispersed Poisson Model

An interesting consequence of this Bayesian model is that it naturally leads to a distribution of spike counts that is overdispersed relative to simple Poisson model, a feature that has been observed in neural recordings (Goris et al., 2014). Recent work has explored the negative binomial distribution as an alternative to Poisson model, since its two parameters allow for Fano factors greater than one. The negative binomial (NB) distribution can also be seen as a continuous mixture of Poisson distributions (i.e., a compound probability distribution) where the mixing distribution of the Poisson rate is a gamma distribution (Gelman et al., 2013). In other words, the NB distribution is viewed as a gamma-Poisson (mixture distribution): a $\text{Poisson}(\lambda)$ distribution whose rate λ is itself a gamma random variable. In our case, the gamma prior over firing rates leads to a negative binomial marginal distribution over $y_{c,t}$.

Though the marginal spike count at a particular time t may be marginally distributed according to a negative binomial distribution, it is not necessarily true that a sequence of time bins, $\mathbf{y}_{1:T}$, will be i.i.d. negative binomials. This arises from the correlations induced by state transition matrix. Instead, $\mathbf{y}_{1:T}$ will follow a finite mixture of Poisson distributions, with one component for each latent state. The mixture will be weighted by the marginal probability of the corresponding latent state. However, as the number of visited states grows, and the marginal probability of latent states becomes more uniform, the resulting marginal distribution over the sequence of spike counts inherits the over dispersed nature of the negative binomial distribution. This is particularly true of a HDP-HMM with a high concentration.

2.4 Markov Chain Monte Carlo (MCMC) Inference

Several MCMC-based inference methods have been developed for the HDP-HMM (Teh et al., 2006; Beal et al., 2002; Van Gael et al., 2008). Some of these previous works use a collapsed Gibbs sampler in which the transition matrix \mathbf{P} and the observation parameters $\boldsymbol{\Lambda}$ are integrated out (Teh et al., 2006; Van Gael et al., 2008). In this work, however, we use a “weak limit” approximation in which the DP prior is approximated with a symmetric Dirichlet prior. Specifically, we let

$$\begin{aligned} \boldsymbol{\beta} | \gamma &\sim \text{Dir}(\gamma/M, \dots, \gamma/M), \\ \boldsymbol{\pi} | \alpha_0, \boldsymbol{\beta} &\sim \text{Dir}(\alpha_0\beta_1, \dots, \alpha_0\beta_M), \\ \mathbf{P}_{i,:} | \alpha_0, \boldsymbol{\beta} &\sim \text{Dir}(\alpha_0\beta_1, \dots, \alpha_0\beta_M). \end{aligned} \tag{5}$$

where M denotes a truncation level for approximating the infinity (which is different from m in the finite-state setting). It can be shown that this prior will weakly converge to the DP prior as the dimensionality of the Dirichlet distribution approaches infinity (Johnson and Willsky, 2014; Ishwaran and Zarepour, 2002). With this approximation we can capitalize on forward-backward sampling algorithms to jointly update the latent states \mathcal{S} .

Previous work has typically been presented with Gaussian or multinomial likelihood models, with the acknowledgement that the same methods work with any exponential family likelihood when the base measure, H is a conjugate prior. Here we present the Gibbs sampling algorithm of Teh et al. (2006) for the HDP-HMM with independent Poisson observations for each cell, and derive Hamiltonian Monte Carlo (HMC) transitions to sample the cell-specific hyperparameters of the firing rate priors.

We begin by defining Gibbs updates for the neuronal firing rates $\boldsymbol{\Lambda}$. Since we are using gamma priors with independent Poisson observations, the model is fully conjugate and simple Gibbs updates suffice. We have,

$$\lambda_{c,i} | \mathbf{y}, \mathcal{S} \sim \text{Gamma} \left(\alpha_c^0 + \sum_{t=1}^T y_{c,t} \mathbb{I}[S_t = i], \beta_c^0 + \sum_{t=1}^T \mathbb{I}[S_t = i] \right). \tag{6}$$

Under the weak limit approximation the priors on $\mathbf{P}_{i,:}$ and $\boldsymbol{\pi}$ reduce to Dirichlet distributions, which are also conjugate with the finite HMM. Hence we can derive conjugate Gibbs updates for these parameters as well. They take the form:

$$\begin{aligned}\boldsymbol{\pi} | \alpha_0, \boldsymbol{\beta} &\sim \text{Dir}(\alpha_0 \boldsymbol{\beta} + \mathbf{1}_{S_1}), \\ \mathbf{P}_{i,:} | \alpha_0, \boldsymbol{\beta} &\sim \text{Dir}(\alpha_0 \boldsymbol{\beta} + \mathbf{n}_i), \\ n_{i,j} &= \sum_{t=1}^{T-1} \mathbb{I}[S_t = i, S_{t+1} = j],\end{aligned}\tag{7}$$

where $\mathbf{1}_j$ is a unit vector with a one in the j -th entry.

Conditioned upon the firing rates, the initial state distribution, and the transition matrix, we can jointly update the latent states of the HDP-HMM using a *forward message passing, backward sampling* algorithm. Details can be found in [Johnson \(2014\)](#); the intuition is that in the backward pass of the algorithm, we have a conditional distribution over S_t given $S_{t+1:T}$. We can iteratively sample from these distributions as we go backward in time to generate a full sample from $p(\mathcal{S} | \mathbf{P}, \boldsymbol{\pi}, \boldsymbol{\Lambda})$. Jointly sampling these latent states allows us to avoid issues with mixing when individually sampling states that are highly correlated with one another.

Finally, the Dirichlet parameters $\boldsymbol{\beta}$ and the concentration parameters α_0 and γ can be updated using standard techniques. For more information, see [Teh et al. \(2006\)](#). A single iteration of the final algorithm consists of an update for each parameter of the model. The aforementioned updates are based upon previous work; one novel direction that we explore in this work is the sampling of the hyperparameters of the gamma firing rate priors.

2.4.1 Setting Firing Rate Hyperparameters

We consider two approaches to setting the hyperparameters of the gamma priors for Poisson firing rates, namely, $\{\alpha_c^0, \beta_c^0\}$ for the c -th cell. Unfortunately they do not have a simple conjugate prior, but we can resort to other Bayesian techniques. First, these parameters may be estimated using an empirical Bayesian (EB) procedure, that is, by maximizing the marginal likelihood of the spike counts. For each cell, this may be easily done using standard maximum likelihood estimation for the negative binomial model. Alternatively, we can place a weak prior and sample the hyperparameters using HMC. For simplicity, we use an improper, uniform prior on $\log \alpha_c^0$ and $\log \beta_c^0$.

To implement HMC we must have access to both the log probability of the parameters as well as its gradient. Since both parameters are restricted to be positive, we instead reparameterize the problem in terms of their logs. For cell c , the conditional log probability equal to,

$$\begin{aligned}\mathcal{L} &= \log p(\log \alpha_c^0, \log \beta_c^0 | \boldsymbol{\Lambda}_{c,:}) \\ &= \sum_{i=1}^m \log p(\lambda_{c,i} | \alpha_c^0, \beta_c^0) + \text{const.} \\ &= \sum_{i=1}^m \alpha_c^0 \log \beta_c^0 - \log \Gamma(\alpha_c^0) + (\alpha_c^0 - 1) \log \lambda_{c,i} - \beta_c^0 \lambda_{c,i}.\end{aligned}\tag{8}$$

Taking gradients with respect to both parameters yields,

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \log \alpha_c^0} &= \sum_{i=1}^m [\log \beta_c^0 - \Psi(\alpha_c^0) + \log \lambda_{c,i}] \times \alpha_c^0, \\ \frac{\partial \mathcal{L}}{\partial \log \beta_c^0} &= \sum_{i=1}^m \left[\frac{\alpha_c^0}{\beta_c^0} - \lambda_{c,i} \right] \times \beta_c^0.\end{aligned}\tag{9}$$

With the parameter and hyperparameter inference complete, we evaluate the performance of our algorithm in terms of its predictive log likelihood on held out test data. We approximate the predictive log likelihood with samples from the posterior distribution generated by our MCMC algorithm. That is,

$$\begin{aligned}\log p(\mathbf{y}_{test} | \mathbf{y}_{1:T}) &= \log \sum_{\mathcal{S}_{test}} \int_{\boldsymbol{\Theta}} p(\mathbf{y}_{test}, \mathcal{S}_{test} | \boldsymbol{\theta}) p(\boldsymbol{\theta} | \mathbf{y}_{train}) d\boldsymbol{\theta}, \\ &\approx \log \frac{1}{N} \sum_{n=1}^N \sum_{\mathcal{S}_{test}} p(\mathbf{y}_{test}, \mathcal{S}_{test} | \boldsymbol{\theta}_n),\end{aligned}\tag{10}$$

where $\theta = (\mathbf{\Lambda}, \mathbf{P}, \boldsymbol{\pi})$ and $\theta_n \sim p(\theta | \mathbf{y}_{train})$. The summation over latent state sequences for the test data is performed with the message passing algorithm for HMMs.

2.5 Variational Bayes (VB) Inference

We build upon our previous work (Chen et al., 2012a; 2014) as well as the recent work of Johnson and Willsky (2014) to develop a variational inference algorithm for fitting the HDP-HMM to hippocampal spike trains. Our objective is to approximate the posterior distribution of the HDP-HMM with a distribution from a more tractable family. As usual, we choose a factorized approximation that allows for tractable optimization of the parameters of the variational model. Specifically, we let,

$$p(\mathcal{S}, \mathbf{\Lambda}, \mathbf{P}, \boldsymbol{\pi}, \boldsymbol{\beta} | \mathbf{y}_{1:T}) \approx q(\mathcal{S})q(\mathbf{\Lambda})q(\mathbf{P})q(\boldsymbol{\pi})q(\boldsymbol{\beta}). \quad (11)$$

Since the independent Poisson observations are conjugate with the gamma firing rate prior distributions, choosing a set of independent gamma distributions for $q(\mathbf{\Lambda})$ allows for simple variational updates.

$$\begin{aligned} q(\mathbf{\Lambda}) &= \prod_{i=1}^M \prod_{c=1}^C \text{Gamma}(\tilde{\alpha}_{c,i}, \tilde{\beta}_{c,i}), \\ \tilde{\alpha}_{c,i} &\leftarrow \alpha_c^0 + \sum_{t=1}^T y_{c,t} \mathbb{E}_q[\mathbb{I}[S_t = i]], \\ \tilde{\beta}_{c,i} &\leftarrow \beta_c^0 + \sum_{t=1}^T \mathbb{E}_q[\mathbb{I}[S_t = i]]. \end{aligned} \quad (12)$$

Following Johnson and Willsky (2014), we use a ‘‘direct assignment’’ truncation for the HDP (Bryant and Sudderth, 2012; Liang et al., 2007). In this scheme, a truncation level M is chosen a priori and $q(\mathcal{S})$ is limited to support only states $S_t \in \{1, \dots, M\}$. The advantage of this approximation is that conjugacy is retained with $\mathbf{\Lambda}$, \mathbf{P} , and $\boldsymbol{\pi}$, and the variational approximation $q(\mathcal{S})$ reduces to:

$$\begin{aligned} q(\mathcal{S}) &= \text{HMM}(\tilde{\mathbf{P}}, \tilde{\boldsymbol{\pi}}, \tilde{\mathbf{\Lambda}}), \\ \tilde{\mathbf{P}} &= \exp\{\mathbb{E}_q[\ln \mathbf{P}]\}, \\ \tilde{\boldsymbol{\pi}} &= \exp\{\mathbb{E}_q[\ln \boldsymbol{\pi}]\}, \\ \tilde{\mathbf{\Lambda}} &= \exp\{\mathbb{E}_q[\ln p(\mathbf{y} | \mathbf{\Lambda})]\}. \end{aligned} \quad (13)$$

Expectations $\mathbb{E}_q[S_t = i]$ can then be computed using standard message passing algorithms for HMMs.

With the direct assignment truncation, the variational factors for $\mathbf{P}_{i,:}$ and $\boldsymbol{\pi}$ take on Dirichlet priors. Unlike in the finite HMM, however, these Dirichlet priors are now over $M + 1$ dimensions since the final dimension accounts for all states $i > M$. Under the HDP prior we had $P_{i,:} \sim \text{DP}(\alpha_0 \cdot \boldsymbol{\beta})$, and under the truncation the DP parameter becomes $\alpha_0 \cdot \boldsymbol{\beta}_{1:M+1}$. Again, leveraging the conjugacy of the model, we arrive at the following variational updates:

$$\begin{aligned} q(\mathbf{P}) &= \prod_{i=1}^M \text{Dir}(\tilde{\boldsymbol{\alpha}}_P^{(i)}), \\ (\tilde{\boldsymbol{\alpha}}_P^{(i)})_j &\leftarrow \alpha_0 \beta_j + \mathbb{E}_q[\mathbb{I}[S_t = i] \cdot \mathbb{I}[S_{t+1} = j]]. \end{aligned} \quad (14)$$

We use an analogous update for $\boldsymbol{\pi}$.

The principal drawback of the direct assignment truncation is that the prior for $\boldsymbol{\beta}$ is no longer conjugate. This could be avoided with the fully conjugate approach of Hoffman et al. (2013), however, this results in extra bookkeeping and the duplication of states. Instead, following Johnson and Willsky (2014); Bryant and Sudderth (2012); Liang et al. (2007), we use a point estimate for this parameter by setting $q(\boldsymbol{\beta}) = \delta_{\boldsymbol{\beta}^*}$ and use gradient ascent with backtracking line search to update this parameter during inference.

There are a number of hyperparameters to set for the variational approach as well. The hyperparameters α_c^0 and β_c^0 of gamma prior on firing rates can be set with empirical Bayes, as above. We resort to cross validation in order to set the Dirichlet parameter α_0 and the GEM parameter γ .

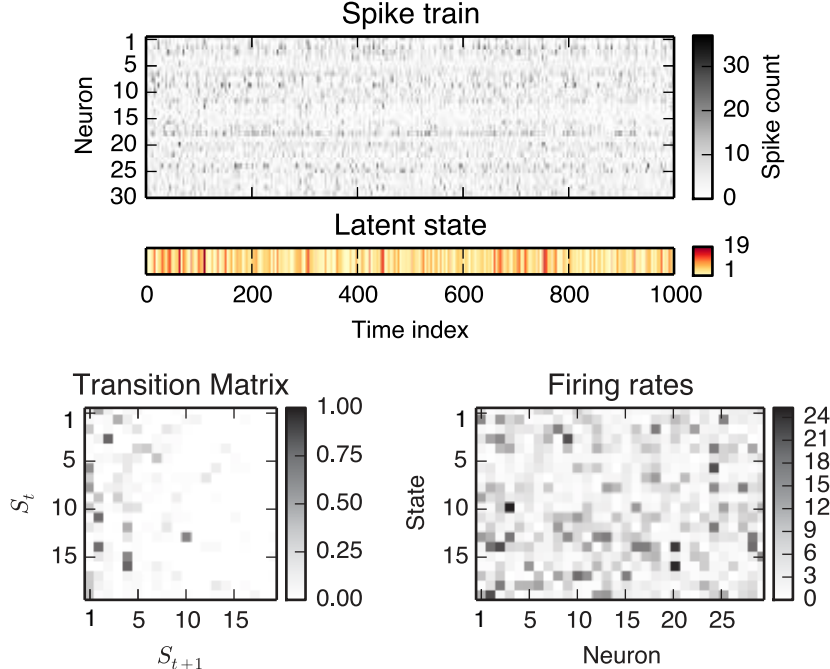


Figure 1: An example of a synthetic dataset drawn from an HDP-HMM. From the observed spike trains (top), we infer the latent state sequence (middle), the transition matrix \mathbf{P} (lower left), and the firing rate vectors $\boldsymbol{\lambda}_i$ (rows in lower right) specific to each state.

Finally, in order to compute predictive log likelihoods on held out test data, we draw multiple samples $\{(\boldsymbol{\Lambda}, \mathcal{S}, \mathbf{P}, \boldsymbol{\pi}, \boldsymbol{\beta})_n\}_{n=1}^N$ from the variational posterior and approximate the predictive log likelihood as,

$$\begin{aligned} \ln p(\mathbf{y}_{test} | \mathbf{y}_{1:T}) &\approx \ln \mathbb{E}_q [p(\mathbf{y}_{test} | \mathcal{S}, \boldsymbol{\Lambda}, \mathbf{P}, \boldsymbol{\pi}, \boldsymbol{\beta})] \\ &\approx \ln \frac{1}{N} \sum_{n=1}^N p(\mathbf{y}_{test} | (\mathcal{S}, \boldsymbol{\Lambda}, \mathbf{P}, \boldsymbol{\pi}, \boldsymbol{\beta})_n). \end{aligned} \quad (15)$$

3 Results

The inference algorithms were implemented based upon the PyHSMM framework of Johnson (2014). The codebase was written in Python with C offloads for the message passing algorithms. We extended the codebase to perform hyperparameter inference using the methods described above, and expanded it to tailor to neural spike train analysis. Our code is publicly available (<https://github.com/slinderman>).

3.1 Synthetic Data

First, we simulate synthetic spike count data using an HDP-HMM with $C = 30$ neurons, $T = 1000$ time bins, and Dirichlet concentration parameters $\alpha_0 = 4.0$ and $\gamma = 8.0$. These yield state sequences that tend to visit 15-30 states. All of neuronal firing rate parameters are drawn from a gamma distribution: $\text{Gamma}(1, 0.2)$ (with mean 5 and standard deviation 5).

An example of one such synthetic dataset is shown in Fig. 1. The states have been ordered according to their occupancy (i.e., how many times they are visited during the simulation), such that the columns of the transition matrix exhibit a decrease in probability as the incoming state number, S_{t+1} , increases. This is a characteristic of the HDP-HMM, and indicates the tendency of the model to reuse states with high occupancy.

We compare five combinations of model, inference algorithm, and hyperparameter selection approaches: (i) HMM with the correct number of states, fit by MCMC and HMC; (ii) HMM with the correct number of states, fit by VB with hyperparameters set by EB; (iii) HDP-HMM fit by MCMC with HMC for hyperparameter selection; (iv) HDP-HMM fit by MCMC with EB for hyperparameters; and (v) HDP-HMM fit by VB with hyperparameters set by EB. For the MCMC methods, we set gamma priors over α_0 and γ and use Gibbs sampler as described above;

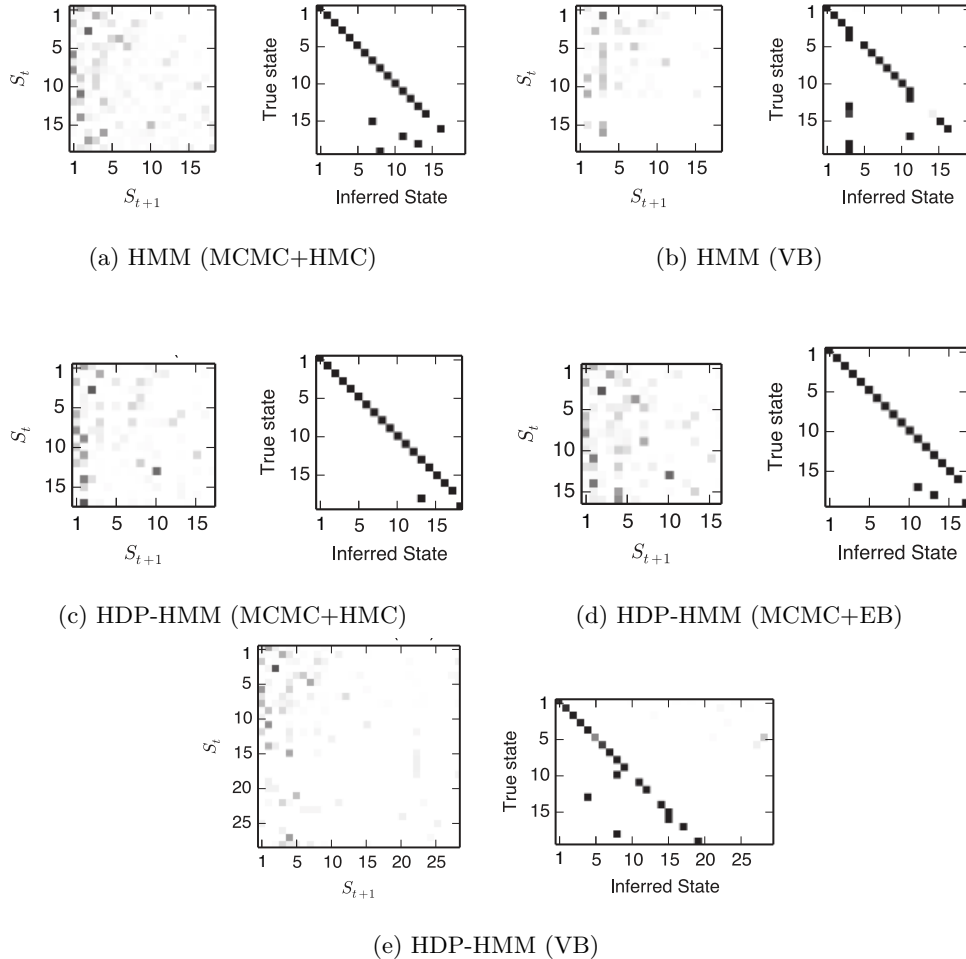


Figure 2: A comparison of inferred state transition matrices and their alignment with the true transition matrix using five combinations of model and inference algorithm. The true state transition matrix is shown in Fig. 1, and is drawn from an HDP-HMM. We find that the MCMC-based inference algorithms yield state estimates that have better correspondence with the true states. The HDP-HMM fit using MCMC combined with HMC for hyperparameter sampling achieves the best match, as evidenced by the nearly one-to-one mapping between true and inferred states (a diagonal matrix indicates a perfect match). The VB algorithms find state sequences with many-to-many maps between true and inferred states. A complete description of the models and inference algorithms is given in the main text.

for the VB methods, we set α_0 and γ to their true values. Alternatively, they can be selected by cross validation. We set both the weak limit approximation for MCMC and the direct assignment truncation level for VB to $M = 80$.

We collect 300 samples from the MCMC algorithms and use the last 50 for computing predictive log likelihoods. For visualization, we use the final sample to extract the transition matrix and the firing rates. The number of samples and the amount of burn-in iterations were chosen by examining the log probability and parameter traces for convergence. It is found that the MCMC algorithm converges within tens of iterations. For further convergence diagnosis of a single Gibbs chain, one may use the autocorrelation tools suggested in (Raftery and Lewis, 1992; Cowles and Carlin, 1996).

We run the VB algorithm for 100 steps to guarantee convergence of the variational lower bound. Again, this is assessed by examining the variational lower bound and is found to converge to a local maxima within tens of iterations.

We use two criteria for result assessment with simulation data. The first criterion is based on the state-state map. Ideally, if two state sequences are consistent, the scatter plot of two state sequences will have a one-to-one mapping. The second criterion is the model’s predictive log likelihood (unit: bits/spike) on a held out sequence of $T_{test} = 200$ time steps. We compare the predictive log likelihood to that of a set of independent Poisson processes.

Table 1: Predictive log likelihood (bits/spike) comparison for 10 simulated datasets, measured in bits per spike improvement over a baseline of independent, homogeneous Poisson processes (the best result in each data set is marked in bold font).

	HMM (MCMC+HMC)	HMM (VB)	HDP-HMM (MCMC+HMC)	HDP-HMM (MCMC+EB)	HDP-HMM (VB)
Dataset 1	0.321	0.286	0.323	0.323	0.299
Dataset 2	0.348	0.255	0.361	0.359	0.350
Dataset 3	0.317	0.294	0.318	0.318	0.297
Dataset 4	0.385	0.329	0.385	0.377	0.353
Dataset 5	0.331	0.310	0.332	0.333	0.333
Dataset 6	0.323	0.298	0.333	0.331	0.329
Dataset 7	0.307	0.274	0.320	0.320	0.317
Dataset 8	0.320	0.223	0.319	0.313	0.319
Dataset 9	0.261	0.251	0.262	0.261	0.259
Dataset 10	0.327	0.272	0.337	0.338	0.335

Their rates and the corresponding predictive log likelihood are given by,

$$\hat{\lambda}_c = \frac{1}{T_{train}} \sum_{t=1}^{T_{train}} y_{c,t}, \quad (16)$$

$$\log p(\mathbf{y}_{test} | \mathbf{y}_{train}) = \sum_{c=1}^C \left[-T_{test} \hat{\lambda}_c + \sum_{t=1}^{T_{test}} y_{c,t} \log \hat{\lambda}_c \right]. \quad (17)$$

The improvement obtained by a model is measured in bits, and is normalized by the number of spikes in the test dataset in order to obtain comparable units for each of the test datasets.

Figure 2 shows the inferred state transition matrices (left half of each panel) and the corresponding state-state map between the true and inferred states (right half) for each of the model, inference algorithm, and initialization combinations. If the mapping is one-to-one, the map will be square and each row will have only one nonzero entry. To facilitate visualization in the presence of permutation ambiguity, we have reordered the inferred states to make the resulting map as diagonal as possible. We use a simple algorithm to match inferred states to true states based on their overlap. We begin by marking all true and inferred states as “unmatched,” then we find the pair of true and inferred states with the highest overlap, mark them as “matched,” remove them from the list of unmatched states, and repeat until either the true or inferred states have all been matched. We use this greedy mapping to permute the inferred states for presentation. Since the HDP-HMM may infer a different number of states than are present in the true model, the map may not be a square matrix (e.g., Fig. 2e). Again, the true states are ordered according to their occupancy so that true state with the lowest index is the most visited state.

Table 1 summarizes the predictive log likelihood comparison among 10 independent runs. For 7 of the 10 datasets, the HDP-HMM fit based on MCMC and HMC performs best, though in some cases the difference between the HDP-HMM when using HMC versus EB for hyperparameter selection is miniscule. In the cases where the HDP-HMM (MCMC+HMC) is not the best, its performance differs by at most 0.001 bits/spike.

Though computation cost is often a major factor with Bayesian inference, with the optimized PyHSMM package, the models can be fit to the synthetic data in under five minutes on an Apple MacBook Air. The runtime necessarily grows the number of neurons and the truncation limit on the number of latent states. As the model complexity grows, we must also run our MCMC algorithm for more iterations, which often motivates the use of variational inference algorithms instead. Given our optimized implementation and the performance improvements yielded by MCMC, we opted for a fully-Bayesian approach using MCMC with HMC for hyperparameter sampling in our subsequent experiments.

Figure 3 shows example traces from the MCMC combined with HMC algorithm for the HDP-HMM running on synthetic dataset 1. This is the same data from which Fig. 2 is generated. The first 5 iterations have been omitted to highlight the variation in the latter samples (the first few iterations rapidly move away from the initial conditions). We see that the log likelihood of the data rapidly converges to nearly that of the true model, and the number of states quickly converges to $m = 19$. Referring back to Fig. 2, we see that this set of states has combined two of the less occupied true states into one. Note that the nuisance parameters α_0 and γ do not converge to the true values within 300 iterations— this could be due to the fact that the solution is not sensitive to these parameters or the presence of local optima. However, even the concentration parameters are different from the true values, they are still consistent with the inferred state transition matrix.

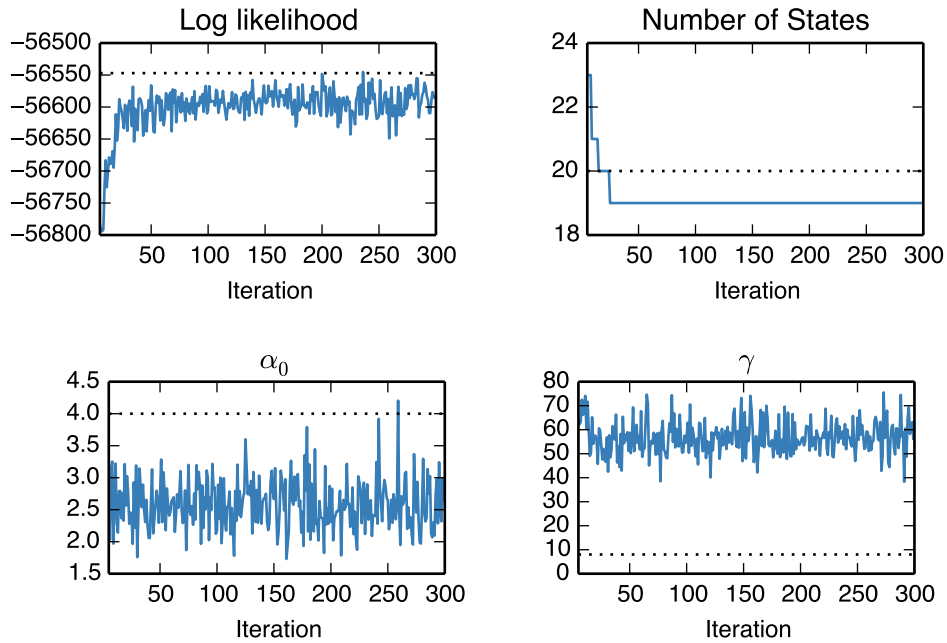


Figure 3: MCMC state trajectories for an HDP-HMM fit to the synthetic dataset shown in Fig. 1. True values are shown by the dotted black lines. The first five iterations of the Markov chain are omitted since they differ greatly from the final states. The chain quickly converges to nearly the correct number of states and achieves close to the true log likelihood.

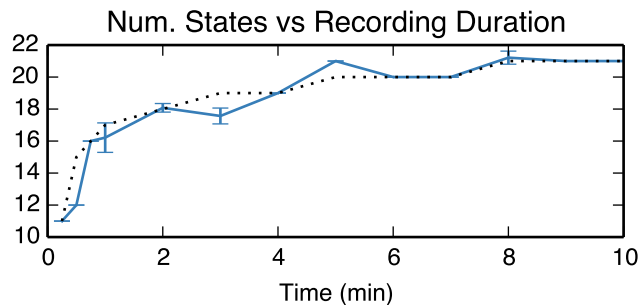


Figure 4: Analysis of number of inferred states as a function of length of the recording. We simulated a synthetic dataset from an HDP-HMM with 50 neurons for 10 minutes with bins of size 250 ms. At the end of the simulation, the model had visited 50 states (dotted black line). Fitting the HDP-HMM with MCMC+HMC on increasing subsets of the data shows a similar trend in number of inferred states, plateauing at around 20-21 states given 5 minutes of data.

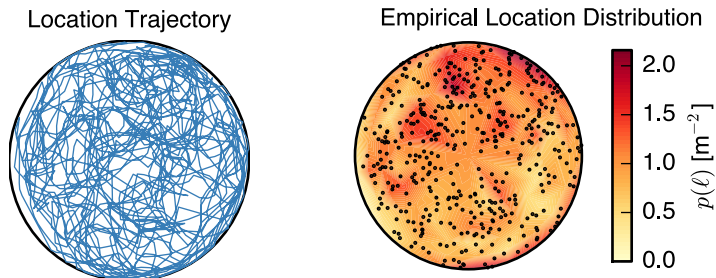


Figure 5: One rat’s behavioral trajectory (left) and spatial occupancy (right) in the open field environment.

Finally, we considered how the number of inferred states varies as a function of recording duration. We simulated 2400 sample points (which is equivalent to 10-min recording with 250 ms temporal bins) from an HDP-HMM with the same parameters as above. The simulation resulted in 21 states visited over the total dataset. We then fit the model with a HDP-HMM with MCMC plus HMC on increasing subset size of the data. The model was initialized with a $\text{Gamma}(1.0, 1.0)$ prior on α_0 and a $\text{Gamma}(8.0, 1.0)$ prior on γ . Thus, the prior means are equal to the underlying concentration parameters. Figure 4 shows the number of inferred states (blue line) and the true number of states (dotted black line) as a function of recording duration. As hoped, the number of inferred states parallels the true number of states; this is indeed the case when five minutes of data are used.

3.2 Rat Hippocampal Neuronal Ensemble Data

Next, we apply the proposed methods to experimental data of the rat hippocampus. Experiments were conducted under the supervision of the Massachusetts Institute of Technology (MIT) Committee on Animal Care and followed the NIH guidelines. The micro-drive arrays containing multiple tetrodes were implanted above the right dorsal hippocampus of male Long-Evans rats. The tetrodes were slowly lowered into the brain reaching the cell layer of CA1 two to four weeks following the date of surgery. Recorded spikes were manually clustered and sorted to obtain single units using a custom software (XClust, M.A.W.).

For demonstration purpose, an ensemble spike train recording of $C = 47$ cells was collected from a single rat for a duration of 9.8 minutes. Once stable hippocampal units were obtained, the rat was allowed to freely forage in an approximately circular open field environment (radius: ~ 60 cm). To identify the period of rodent locomotion during spatial navigation, we used a velocity threshold (>10 cm/s) to select the RUN epochs and merged them together. One animal’s RUN trajectory and spatial occupancy are shown in Fig. 5 (left and right panels, respectively). The empirical probability of a location, $p(\ell)$, is determined by dividing the arena into 220 bins of equal area (11 angular bins and 20 radial bins) and counting the fraction of time points in which the rat is in the corresponding bin.

In the experimental data analysis, we focus on nonparametric Bayesian inference for HDP-HMM. For all methods, we increase the truncation level to a large value of $M = 200$. To discover the model order of the variational solutions, we use the number of states visited by the most likely state sequence under the variational posterior. The MCMC algorithms yield samples of state sequences from which the model order can be directly counted.

For the purpose of result assessment, we plot the state space map (Fig. 6, top left), which shows the mean value of the spatial position that each state represented. The size of the black dot is proportional to the occupancy of the state. We also plot the empirical location distribution for the top three states as measured by occupancy (Fig. 6, top right). In the bottom of Fig. 6, we show the estimated animal’s spatial trajectories (polar coordinate) in black, along with the reconstructed location in from the HDP-HMM with MCMC and HMC in blue. To reconstruct the position, we use the mean of each latent state’s location distribution weighted by the marginal probability of that state under the HDP-HMM. That is,

$$\hat{r}_t = \sum_{i=1}^m \bar{r}_i \Pr(\mathcal{S}_t = i), \quad \hat{\theta}_t = \sum_{i=1}^m \bar{\theta}_i \Pr(\mathcal{S}_t = i), \quad (18)$$

where \bar{r}_i and $\bar{\theta}_i$ denote the average location of the rat while in inferred state i (corresponding to the black dots in Fig. 6, top leftmost panel). Note that the animal’s position is not used in model inference, only during result assessment. In the illustrated example (HDP-HMM with MCMC+HMC), the mean reconstruction error in Euclidean distance is 9.07 cm.

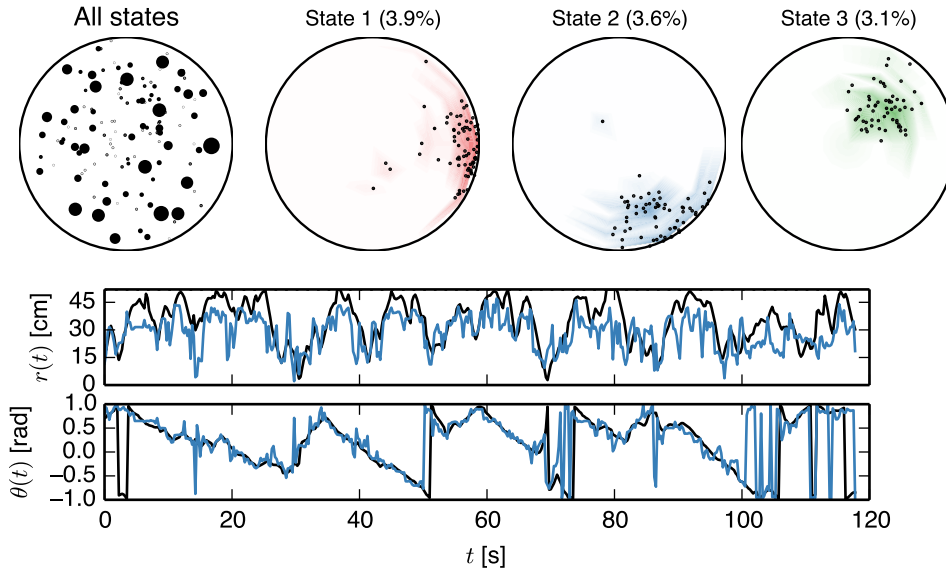


Figure 6: Estimation result from HDP-HMM (MCMC+HMC) for the rat hippocampal spike train. (Top left) Estimated state space map, where the mean value of the spatial position for each latent state is shown by a black dot. The size of the dot is proportional to the occupancy of the state. (Top right) Probability distributions over location corresponding to the top three latent states, measured by state occupancy. The small black dots indicate the location of the animal while in that state, and are used to compute the empirical distribution over location indicated by colored shading. (Bottom) The true trajectory (in polar coordinates) is shown in black and the reconstructed trajectory is shown in blue. For each time bin, we use the mean location of the latent states to determine an estimate of the animal’s location.

Table 2: A comparison of HDP-HMM models and algorithms on the rat hippocampal data. Performance is measured in mean decoding error and predictive log likelihood on two minutes of held out test data (the best result is marked in bold font).

	HDP-HMM (MCMC+HMC)	HDP-HMM (MCMC+EB)	HDP-HMM (VB)
Decoding error (cm)	9.07 ± 5.14	9.50 ± 5.73	10.41 ± 5.69
Predictive log likelihood (bits/spike)	0.591 ± 0.008	0.530 ± 0.010	0.581 ± 0.005

As the parameter sample traces in Fig. 7 show, the Markov chains from HDP-HMM (MCMC+HMC) and HDP-HMM (MCMC+EB) converge in around 7500 iterations. After this point, the total number of states stabilizes to 125.5 ± 2.0 , but the number of states that account for 95% of the time bins stabilizes to 77.1 ± 1.8 . The concentration parameters α_0 and γ converge within a similar number of iterations. Finally, we show the transition matrix \mathbf{P} and firing rate matrix $\mathbf{\Lambda}$ obtained from the final sample.

We perform a quantitative comparison between the three nonparametric models in terms of both decoding error and predictive log likelihood. For both tests, we train the models on the first 7.8 minutes of data and test on the final two minutes of data for prediction. The results are summarized in Table 2. We find that the HDP-HMM (MCMC+HMC) again outperforms the competing models in both measures, though the differences in decoding performance are not statistical significant.

Looking into the inferred states, we can reconstruct the “place fields” of the hippocampal neurons, that is the distribution over locations of the rat given that a neuron fired. To do so, we combine the state-location maps of Fig. 6 (top right) with the firing rate of the neuron of cell c in those states and weight by the marginal probability of the latent state. Together, these give rise to the inferred neuron’s place field, where, again, the position data was only used in reconstruction but not in the inference procedure. Four pairs of inferred and true place fields are shown in Fig. 8. On the left is the inferred place field; on the right is the true place field computed using the locations of the rat when cell c fired shown by black dots.

In addition, we can evaluate the model in terms of the information latent states convey about the rat’s position in the circular environment. To do so, we divide the environment into 121 bins of equal area and treat the rat’s position as a discrete random variable. Likewise, we treat the latent state as a discrete random variable, and we

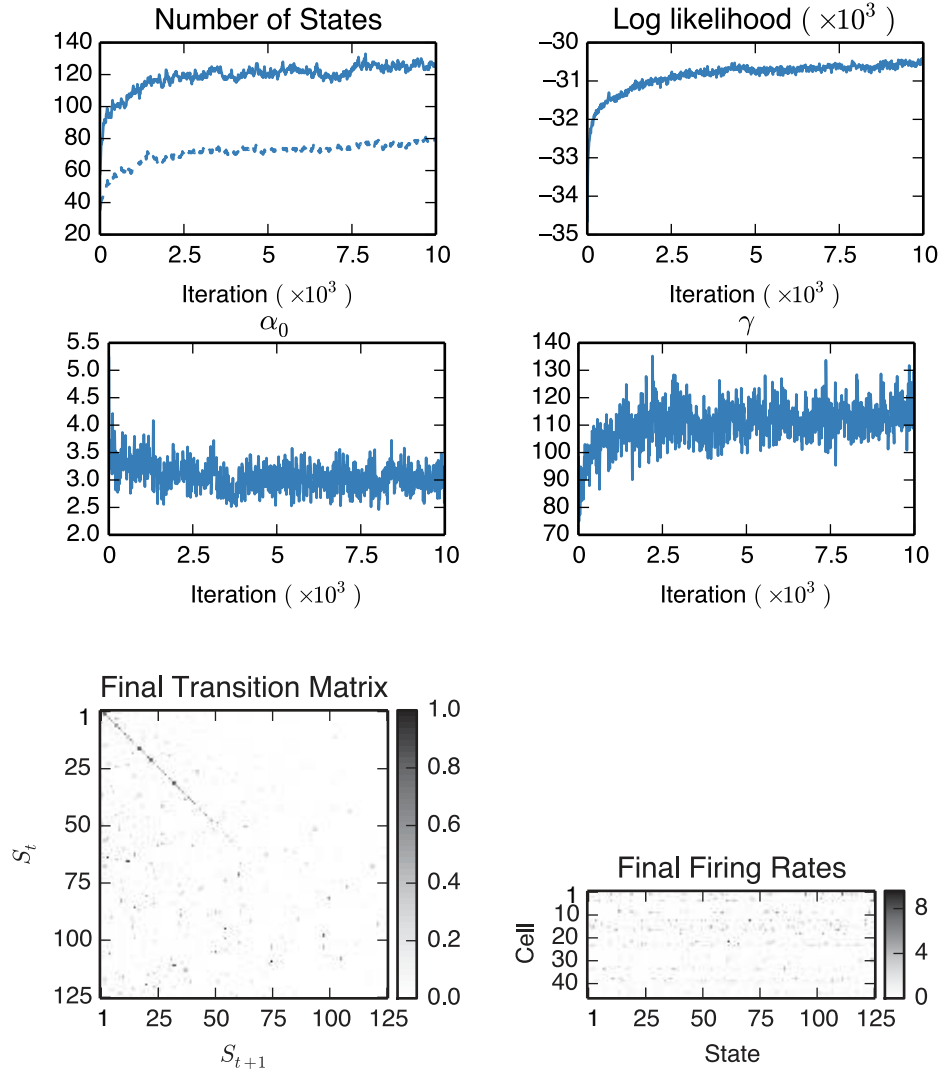


Figure 7: Estimation result from HDP-HMM (MCMC+HMC) for the rat hippocampal data. The total number of states (solid blue) slowly increases as states are allocated for a small number of time bins. The number of states accounting for 95% of the data (dashed line) converges after 7500 iterations. The log likelihood of the training data grows consistently as highly specific states are added. The concentration parameters, α_0 and γ converge after 2500 iterations. In the bottom we show the final state transition matrix and firing rate samples drawn from the last iteration.

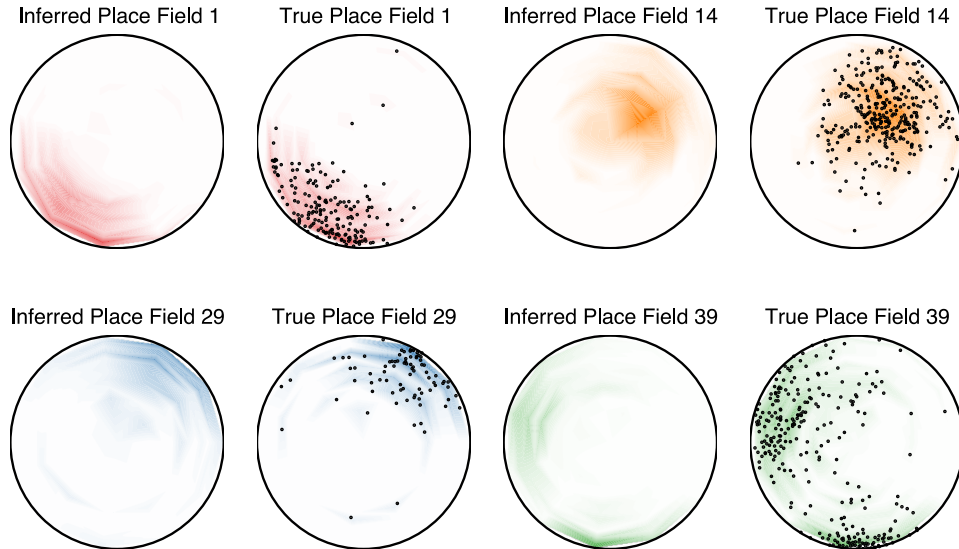


Figure 8: Pairs of inferred and true place fields for four randomly selected neurons. The inferred place field for cell c is a combination of location distributions for each state i weighted by the inferred firing rates $\lambda_{c,i}$, whereas the true place field for cell c is a histogram of locations in which cell c fires. The black dots show the locations used for each histogram. We see that the inferred place fields closely match the true place fields. With adequate data, we expect a higher latent state dimensionality to yield higher spatial resolution in the inferred place fields.

compute the discrete mutual information between these two variables. The left panel of Fig. 9 shows how this mutual information increases with increasing MCMC iteration. As the model refines its estimates of the latent states, the latent state sequence carries greater information about position. We also investigate the information content of each individual state by constructing a binary random variable indicating whether or not the model is in state i and measuring its mutual information with the rat’s position. The result is shown in the right panel of Fig. 9, where the latent states are ordered in decreasing order of occupancy. As expected, states that are more frequently occupied carry the most information about the rat’s position.

4 Extensions and Discussion

4.1 Hidden Semi-Markovian Models

A striking feature of the inferred transition matrix in Fig. 7 is that the first 60 states, those which account for about 95% of the time bins, exhibit strong self transitions. This is a common feature of time series and has been addressed by a number of augmented Markovian models. In particular, hidden semi-Markovian models (HSMMs) explicitly model the duration of time spent in each state separately from the rest of the transition matrix (Johnson and Willsky, 2013). Building this into the model allows the Dirichlet or HDP prior over state transition vectors to explain the rest of the transitions, which are often more similar. Alternatively, “sticky” HMMs and HDP-HMMs accomplish a similar effect (Fox et al., 2008). Our preliminary results (not shown) has already suggested that some improvement can be found in adopting this approach.

4.2 Statistical and Computational Considerations

In Bayesian estimation, we have seen a great advantage in nonparametric Bayesian formalism (i.e., HDP-HMM vs. HMM) regarding automatic model selection. This is especially important for sparse sample size or short recording in some neuroscience applications, where cross-validation on data is infeasible.

For any statistical estimation, we need to consider the “bias vs. variance” problem. In VB inference, there is a potential estimate bias due to bound optimization (since we optimize the lower bound of the marginal likelihood). In addition, because of the mean-field approximation, the parameter’s variance tends to be underestimated. In MCMC inference, the estimate is asymptotically unbiased, however, if the Markov chain mixes slowly, the estimate’s

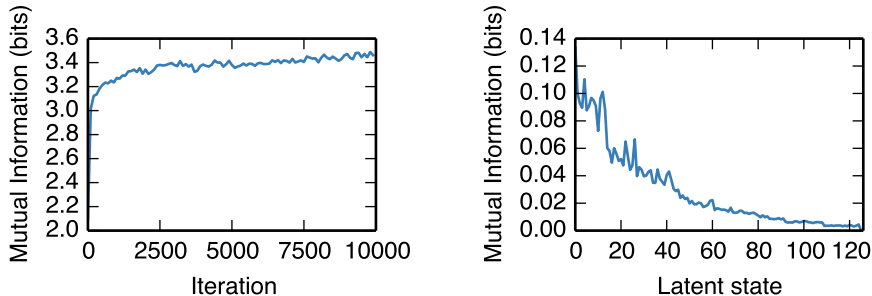


Figure 9: Mutual information of the inferred states and the rat’s position. On the left, we show the mutual information of the state sequence and the rat’s position (discretized into 121 equal-sized bins). On the right, we show how much information each state conveys about the location, ordered by the occupancy of the states. As expected, low-indexed states, i.e. those states which are most occupied, carry the most information.

variance can be inaccurate.

4.3 Latent State Dimensionality

In experimental data analysis, the number of identified states from HDP-HMM depends on the data. Given the same size of environment, different numbers of cells or different lengths of duration may yield different estimation results, since the nonparametric prior allocates states in accordance with the complexity of the data. We found that the weak priors over the concentration parameters have a minimal effect on the number of inferred states. The choice of firing rate hyperparameters does, however, have a strong effect. We also found that manually setting the firing rate hyperparameters to seemingly reasonable values could yield far fewer states. This is most likely because the unused states are assigned firing rates from the prior, and if the prior does not match the data, the sampled firing rates may not be able to explain the data and hence are unlikely to be chosen in favor of an existing state, whose firing rates are influenced by the data as well as the prior. However, it is also worth emphasizing that although the inferred state dimensionality may vary depending on different hyperparameters, the derived decoding error is insensitive to the exact number of state dimensionality.

4.4 Robustness of the Population Firing Model

A key assumption in our probabilistic model is the Poisson likelihood. Although this assumption may not be true in experimental data, our results have showed excellent performance. To further assess the robustness of HDP-HMM-Poisson model in experimental data analysis, at every temporal bin we further add additional homogeneous non-Poissonian noise to the observed population spike counts by drawing from a NB distribution (with varying levels of mean 0.25-1.0 and variance 0.5-2.0), and repeat the decoding error analysis. We have found that, as a general trend, the median decoding error gradually grows as increasing noise mean or variance; yet the decoding performance is still quite satisfactory (results not shown).

4.5 Use of Soft-labeled Spikes

Thus far, we have assumed that all recorded ensemble spikes are sorted and clustered into single units. Nevertheless, it is known that spike sorting is complex, time-consuming and error-prone (Wood and Black, 2008; Shalchyan and Farina, 2014). On the one hand, sorting error is inevitable when there is strong overlapping features (such as spike waveforms or principal components). On the other hand, traditional spike-sorting procedures often throw away considerable non-clusterable “noisy” spikes, which might contain informative tuning information (Kloosterman et al., 2014; Chen et al., 2012b). How to use these noisy spikes and maximize the information efficiency remains an open question. In other words, can we conduct the ensemble spike analysis using unsorted spikes?

Motivated from a sorting-free ensemble decoding analysis (Kloosterman et al., 2014; Chen et al., 2012b), we may use a soft-clustering method based on a Gaussian mixtures model (parameterized by an augmented vector $\xi = \{\ell_c, \mu_c, \Sigma_c\}_{c=1}^K$ that characterizes the weights, mean, and covariance parameters of the Gaussian mixtures). By clustering the spike waveform feature space, we assign each spike with a “soft” class label (about the unit identity)

according to the posterior probability within the K -mixtures. In the feature space, the points close to (far away from) the c -th cluster center are associated with a probability assignment value close to (smaller than) 1 in the c -th class. Because of the soft membership of individual spikes, the spike count $y_{c,t}$ ($c = 1, \dots, K$) within a time interval can be a non-integer value. Consequently, we replace the variable C with K to indicate that the number of neurons is unknown, and rewrite the log likelihood as follows

$$\log p(\mathbf{y}_{1:T}|\mathcal{S}, \boldsymbol{\theta}, \boldsymbol{\xi}) = \sum_{t=1}^T \sum_{c=1}^K \log p(y_{c,t}|S_t, \boldsymbol{\theta}, \boldsymbol{\xi}) \quad (19)$$

In this case, the inference procedure consists of two steps: At the first stage, the d -dimensional spike waveform features are clustered using a “constrained” Gaussian mixture model (Zou and Adams, 2012), which can be either finite or infinite. In the case of infinite Gaussian mixtures, we can also resort to the nonparametric Bayesian approach (Wood and Black, 2008; Rasmussen, 2000; Görür and Rasmussen, 2010). Upon completing the inference, each spike will be given a posterior probability of being assigned to each cluster. At the second stage, we sum the soft-labeled spikes to obtain the probabilistic spike count $y_{c,t}$ for all K -clusters, and the remaining nonparametric Bayesian (MCMC or VB) inference procedure remains unchanged. A detailed investigation of this idea will be pursued in future work.

5 Conclusion

In this paper, we have explored the use of HDP-HMMs with Poisson likelihoods to analyze rat hippocampal ensemble spike data during spatial navigation. Compared to the parametric finite-state HMM, the HDP-HMM allows more flexibility to model the experimental data (without relying on time-consuming cross-validation in model selection). We evaluate two nonparametric Bayesian inference algorithms for HDP-HMM, one based on VB and the other based on MCMC. Furthermore, we consider two approaches for hyperparameter selection, an issue that is particularly important for the real-life application. It is found that the MCMC algorithm with HMC updates for the hyperparameters is robust and achieves the best performance in all simulated and experimental data. Our investigation shows a promising direction in applying nonparametric Bayesian methods for ensemble neuronal spike data analysis.

Acknowledgements

We thank the members of the Wilson laboratory for sharing the experimental data and valuable discussions. S.W.L. was supported by a National Defense Science and Engineering Graduate (NDSEG) Fellowship and by the Center for Brains, Minds and Machines (CBMM). M.J.J. was supported by the Harvard/MIT Joint Research Grants Program. M.A.W. was supported by the NIH Grant R01-MH06197, TR01-GM10498, ONR-MURI N00014-10-1-0936 grant. This material is also based upon work supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216. Z.C. was supported by an NSF-CRCNS award (No. IIS-1307645) from the National Science Foundation.

References

- John O’Keefe and Lynn Nadel. *The Hippocampus as a Cognitive Map*, volume 3. Clarendon Press, 1978.
- Carina Curto and Vladimir Itskov. Cell groups reveal structure of stimulus space. *PLoS Computational Biology*, 4(10): e1000205, 2008.
- Yu Dabaghian, Facundo Memoli, L Frank, and Gunnar Carlsson. A topological paradigm for hippocampal spatial map formation using persistent homology. *PLoS Computational Biology*, 8(8):e1002581, 2012.
- Steven L Scott. Bayesian methods for hidden Markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, 97(457):337–351, 2002.
- Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. CRC press, 3rd edition, 2013.
- Yuriy Mishchenko, Joshua T Vogelstein, and Liam Paninski. A bayesian approach for inferring neuronal connectivity from calcium fluorescent imaging data. *Annals of Applied Statistics*, 5:1229–1261, 2011.

- Zhe Chen, David F Putrino, Soumya Ghosh, Riccardo Barbieri, and Emery N Brown. Statistical inference for assessing functional connectivity of neuronal ensembles with sparse spiking data. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(2):121–135, 2011.
- Zhe Chen. An overview of Bayesian methods for neural spike train analysis. *Computational Intelligence and Neuroscience*, 2013:251905, 2013.
- Thomas J Davidson, Fabian Kloosterman, and Matthew A Wilson. Hippocampal replay of extended experience. *Neuron*, 63(4):497–507, 2009.
- Fabian Kloosterman, Stuart P Layton, Zhe Chen, and Matthew A Wilson. Bayesian decoding using unsorted spikes in the rat hippocampus. *Journal of Neurophysiology*, 111(1):217–227, 2014.
- C Yau, Omiros Papaspiliopoulos, Gareth O Roberts, and Christopher Holmes. Bayesian non-parametric hidden Markov models with applications in genomics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(1):37–57, 2011.
- Zhe Chen, Fabian Kloosterman, Emery N Brown, and Matthew A Wilson. Uncovering spatial topology represented by rat hippocampal population neuronal codes. *Journal of Computational Neuroscience*, 33(2):227–255, 2012a.
- Zhe Chen, Stephen N Gomperts, Jun Yamamoto, and Matthew A Wilson. Neural representation of spatial topology in the rodent hippocampus. *Neural Computation*, 26(1):1–39, 2014.
- Yee Whye Teh and Michael I Jordan. Hierarchical Bayesian nonparametric models with applications. In N. L. Hjort, C. Holmes, P. Müller, and S. G. Walker, editors, *Bayesian Nonparametrics*, pages 158–207. Cambridge University Press, 2010.
- Frank Wood and Michael J Black. A nonparametric Bayesian alternative to spike sorting. *Journal of Neuroscience Methods*, 173(1):1–12, 2008.
- B. M. Yu, J. P. Cunningham, G. Santhanam, S. I. Ryu, K. V. Shenoy, and M. Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Journal of Neurophysiology*, 102:614–635, 2009.
- Vahid Shalchyan and Dario Farina. A non-parametric bayesian approach for clustering and tracking non-stationarities of neural spikes. *Journal of Neuroscience Methods*, 223:85–91, 2014.
- Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581, 2006.
- Matthew J Johnson and Alan S Willsky. Stochastic variational inference for Bayesian time series models. *JMLR W&CP*, 32:1854–1862, 2014.
- Matthew J Johnson. *Bayesian time series models and scalable inference*. PhD thesis, Massachusetts Institute of Technology, June 2014.
- Radford M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, pages 113–162, 2010.
- M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden Markov model. In *Advances in Neural Information Processing Systems 14*, pages 577–585, Cambridge, MA, 2002. MIT Press.
- Warren John Ewens. Population genetics theory—the past and the future. In S. Lessard, editor, *Mathematical and Statistical Developments of Evolutionary Theory*, pages 177–227. Springer, 1990.
- Robbe LT Goris, J Anthony Movshon, and Eero P Simoncelli. Partitioning neuronal variability. *Nature Neuroscience*, 17:858–865, 2014.
- Jurgen Van Gael, Yunus Saatci, Yee Whye Teh, and Zoubin Ghahramani. Beam sampling for the infinite hidden Markov model. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1088–1095, 2008.
- Hemant Ishwaran and Mahmoud Zarepour. Exact and approximate sum representations for the Dirichlet process. *Canadian Journal of Statistics*, 30(2):269–283, 2002.
- Michael Bryant and Erik B Sudderth. Truly nonparametric online variational inference for hierarchical dirichlet processes. In *Advances in Neural Information Processing Systems 25*, pages 2699–2707, 2012.
- Percy Liang, Slav Petrov, Michael I Jordan, and Dan Klein. The infinite PCFG using hierarchical Dirichlet processes. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 688–697, 2007.

- Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.
- Adrian E Raftery and Steven Lewis. How many iterations in the Gibbs sampler? In J. M. Bernardo, J. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics*, pages 763–773. Oxford University Press, 1992.
- Mary Kathryn Cowles and Bradley P Carlin. Markov chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91:883–904, 1996.
- Matthew J Johnson and Alan S Willsky. Bayesian nonparametric hidden semi-Markov models. *Journal of Machine Learning Research*, 14(1):673–701, 2013.
- Emily B Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. An HDP-HMM for systems with state persistence. In *Proceedings of the 25th International Conference on Machine learning*, pages 312–319, 2008.
- Zhe Chen, Fabian Kloosterman, Stuart Layton, and Matthew A Wilson. Transductive neural decoding for unsorted neuronal spikes of rat hippocampus. In *Proceedings of IEEE Engineering in Medicine and Biology Conference*, pages 1310–1313, 2012b.
- J. Y. Zou and R. P. Adams. Priors for diversity in generative latent variable models. In *Advances in Neural Information Processing Systems 24*, Cambridge, MA, 2012. MIT Press.
- Carl Edward Rasmussen. The infinite Gaussian mixture model. In *Advances in Neural Information Processing Systems 12*, pages 554–560, Cambridge, MA, 2000. MIT Press.
- Dilan Görür and Carl Edward Rasmussen. Dirichlet process gaussian mixture models: Choice of the base distribution. *Journal of Computer Science and Technology*, 25(4):653–664, 2010.