Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems

Scott W. Linderman* Columbia University

Matthew J. Johnson* Harvard and Google Brain

Andrew C. Miller Harvard University

Ryan P. Adams Harvard and Google Brain

David M. Blei Columbia University

Liam Paninski Columbia University

Abstract

Many natural systems, such as neurons firing in the brain or basketball teams traversing a court, give rise to time series data with complex, nonlinear dynamics. We can gain insight into these systems by decomposing the data into segments that are each explained by simpler dynamic units. Building on switching linear dynamical systems (SLDS), we develop a model class and Bayesian inference algorithms that not only discover these dynamical units but also, by learning how transition probabilities depend on observations or continuous latent states, explain their switching behavior. Our key innovation is to design these recurrent SLDS models to enable recent Pólya-gamma auxiliary variable techniques and thus make approximate Bayesian learning and inference in these models easy, fast, and scalable.

1 Introduction

Complex dynamical behaviors can often be broken down into simpler units. A basketball player finds the right court position and starts a pick and roll play. A mouse senses a predator and decides to dart away and hide. A neuron's voltage first fluctuates around a baseline until a threshold is exceeded; it spikes to peak depolarization, and then returns to baseline. In each of these cases, the switch to a new mode of behavior can depend on the continuous state of the system or on external factors. By discovering these behavioral units and their switching dependencies, we can gain insight into the rich processes generating complex natural phenomena.

This paper proposes a class of recurrent state space models that captures these intuitive dependencies, as well as corresponding Bayesian inference and learning algorithms that are computationally tractable and scalable to large datasets. We extend switching linear-Gaussian dynamical systems (SLDS) [Ackerson and Fu, 1970, Chang and Athans, 1978, Hamilton, 1990, Bar-Shalom and Li, 1993, Ghahramani and Hinton, 1996, Murphy, 1998, Fox et al., 2009] by allowing the discrete switches to depend on the continuous latent state and exogenous inputs through a logistic regression. This model falls into the general class of hybrid systems, but previously including this kind of dependence has destroyed the conditionally linear-Gaussian structure in the states and complicated inference, as in the augmented SLDS of Barber [2006]. To avoid these complications, we design our model to enable the use of recent auxiliary variable methods for Bayesian inference. In particular, our main technical contribution is an inference algorithm that leverages Pólya-gamma auxiliary variable methods [Polson, Scott, and Windle, 2013, Linderman, Johnson, and Adams, 2015] to make inference both fast and easy.

The class of models and the corresponding learning and inference algorithms we develop have several advantages for understanding rich time series data. First, these models decompose data into simple segments and attribute segment transitions to changes in latent state or environment; this provides interpretable representations of data dynamics. Second, we fit these models using fast, modular Bayesian inference algorithms; this makes it easy to handle Bayesian uncertainty, missing data, multiple observation modalities, and hierarchical extensions. Finally, these models are interpretable, readily able to incorporate prior information, and gen-

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

erative; this lets us take advantage of a variety of tools for model validation and checking.

In the following section we provide background on the key models and inference techniques on which our method builds. Next, we introduce the class of recurrent switching state space models, and then explain the main algorithmic contribution that enables fast learning and inference. Finally, we illustrate the method on a variety of synthetic data experiments and an application to real recordings of professional basketball players.

2 Background

Our model has two main components: switching linear dynamical systems and stick-breaking logistic regression. Here we review these components and fix the notation we will use throughout the paper.

2.1 Switching linear dynamical systems

Switching linear dynamical system models (SLDS) break down complex, nonlinear time series data into sequences of simpler, reused dynamical modes. By fitting an SLDS to data, we not only learn a flexible nonlinear generative model, but also learn to parse data sequences into coherent discrete units.

The generative model is as follows. At each time $t = 1, 2, \ldots, T$ there is a discrete latent state $z_t \in \{1, 2, \ldots, K\}$ that following Markovian dynamics,

$$z_{t+1} \,|\, z_t, \{\pi_k\}_{k=1}^K \sim \pi_{z_t} \tag{1}$$

where $\{\pi_k\}_{k=1}^K$ is the Markov transition matrix and $\pi_k \in [0,1]^K$ is its kth row. In addition, a continuous latent state $x_t \in \mathbb{R}^M$ follows conditionally linear (or affine) dynamics, where the discrete state z_t determines the linear dynamical system used at time t:

$$x_{t+1} = A_{z_{t+1}} x_t + b_{z_{t+1}} + v_t, \quad v_t \overset{\text{iid}}{\sim} \mathcal{N}(0, Q_{z_{t+1}}), \tag{2}$$

for matrices $A_k, Q_k \in \mathbb{R}^{M \times M}$ and vectors $b_k \in \mathbb{R}^M$ for $k = 1, 2, \ldots, K$. Finally, at each time t a linear Gaussian observation $y_t \in \mathbb{R}^N$ is generated from the corresponding latent continuous state,

$$y_t = C_{z_t} x_t + d_{z_t} + w_t, \qquad w_t \overset{\text{iid}}{\sim} \mathcal{N}(0, S_{z_t}), \tag{3}$$

for $C_k \in \mathbb{R}^{N \times M}$, $S_k \in \mathbb{R}^{N \times N}$, and $d_k \in \mathbb{R}^N$. The system parameters comprise the discrete Markov transition matrix and the library of linear dynamical system matrices, which we write as

$$\theta = \{(\pi_k, A_k, Q_k, b_k, C_k, S_k, d_k)\}_{k=1}^K.$$

For simplicity, we will require C, S, and d to be shared among all discrete states in our experiments.

To learn an SLDS using Bayesian inference, we place conjugate Dirichlet priors on each row of the transition matrix and conjugate matrix normal inverse Wishart (MNIW) priors on the linear dynamical system parameters, writing

$$\pi_k \,|\, \alpha \overset{\text{iid}}{\sim} \text{Dir}(\alpha), \quad (A_k, b_k), Q_k \,|\, \lambda \overset{\text{iid}}{\sim} \text{MNIW}(\lambda),$$
$$(C_k, d_k), S_k \,|\, \eta \overset{\text{iid}}{\sim} \text{MNIW}(\eta),$$

where α, λ, and η denote hyperparameters.

2.2 Stick-breaking logistic regression and Pólya-gamma augmentation

Another component of the recurrent SLDS is a stick-breaking logistic regression, and for efficient block inference updates we leverage a recent Pólya-gamma augmentation strategy [Linderman, Johnson, and Adams, 2015]. This augmentation allows certain logistic regression evidence potentials to appear as conditionally Gaussian potentials in an augmented distribution, which enables our fast inference algorithms.

Consider a logistic regression model from regressors $x \in \mathbb{R}^M$ to a categorical distribution on the discrete variable $z \in \{1, 2, \ldots, K\}$, written as

$$z \,|\, x \sim \pi_{\mathsf{SB}}(\nu), \quad \nu = Rx + r,$$

where $R \in \mathbb{R}^{K-1 \times M}$ is a weight matrix and $r \in \mathbb{R}^{K-1}$ is a bias vector. Unlike the standard multiclass logistic regression, which uses a softmax link function, we instead use a stick-breaking link function $\pi_{\mathsf{SB}} : \mathbb{R}^{K-1} \to [0,1]^K$, which maps a real vector to a normalized probability vector via the stick-breaking process

$$\pi_{\mathsf{SB}}(\nu) = \begin{pmatrix} \pi_{\mathsf{SB}}^{(1)}(\nu) & \cdots & \pi_{\mathsf{SB}}^{(K)}(\nu) \end{pmatrix},$$
$$\pi_{\mathsf{SB}}^{(k)}(\nu) = \sigma(\nu_k) \prod_{j<k} (1 - \sigma(\nu_j)) = \sigma(\nu_k) \prod_{j<k} \sigma(-\nu_j),$$

for $k = 1, 2, \ldots, K-1$ and $\pi_{\mathsf{SB}}^{(K)}(\nu) = \prod_{k=1}^K \sigma(-\nu_k)$, where $\sigma(x) = e^x/(1+e^x)$ denotes the logistic function. The probability mass function $p(z \,|\, x)$ is

$$p(z \,|\, x) = \prod_{k=1}^K \sigma(\nu_k)^{\mathbb{I}[z=k]} \sigma(-\nu_k)^{\mathbb{I}[z>k]}$$

where $\mathbb{I}[\cdot]$ denotes an indicator function that takes value 1 when its argument is true and 0 otherwise.

If we use this regression model as a likelihood $p(z \,|\, x)$ with a Gaussian prior density $p(x)$, the posterior density $p(x \,|\, z)$ is non-Gaussian and does not admit easy Bayesian updating. However, Linderman, Johnson, and Adams [2015] show how to introduce Pólya-gamma auxiliary variables $\omega = \{\omega_k\}_{k=1}^K$ so that the

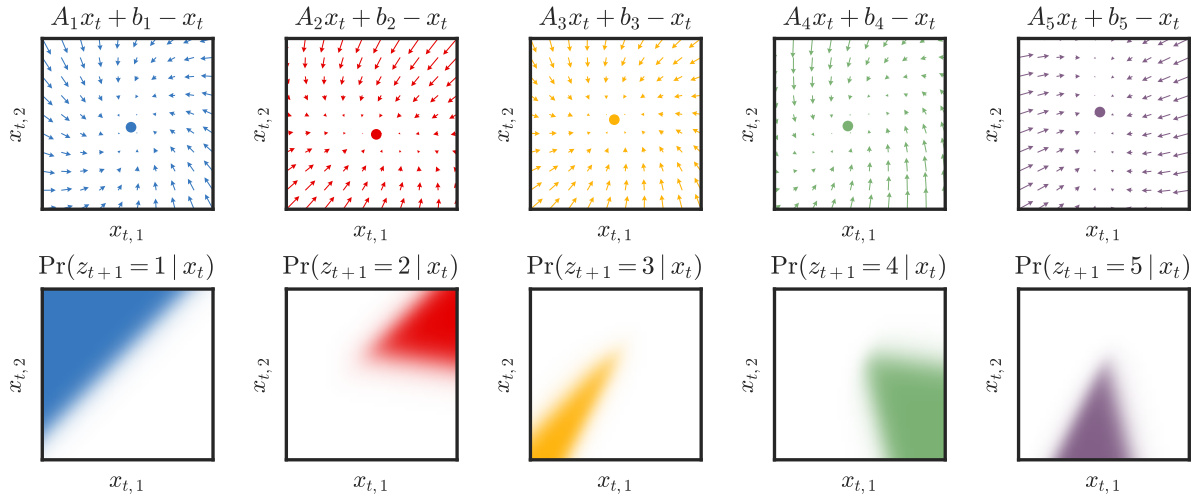


Figure 1: A draw from the prior over recurrent switching linear dynamical systems with $K = 5$ discrete states shown in different colors. **(Top)** The linear dynamics of each latent state. Dots show the fixed point $(I - A_k)^{-1}b_k$. **(Bottom)** The conditional $p(z_{t+1}|x_t)$ plotted as a function of x_t (white=0; color=1). Note that the stick breaking construction iteratively partitions the continuous space with linear hyperplanes. For simpler plotting, in this example we restrict $p(z_{t+1} \mid x_t, z_t) = p(z_{t+1}|x_t)$.

conditional density $p(x \mid z, \omega)$ becomes Gaussian. In particular, by choosing $\omega_k \mid x, z \sim \mathrm{PG}(\mathbb{I}[z \geq k], \nu_k)$, we have,

$$x \mid z, \omega \sim \mathcal{N}(\Omega^{-1}\kappa,\ \Omega^{-1}),$$

where the mean vector $\Omega^{-1}\kappa$ and covariance matrix Ω^{-1} are determined by

$$\Omega = \mathrm{diag}(\omega), \qquad \kappa_k = \mathbb{I}[z = k] - \frac{1}{2}\mathbb{I}[z \geq k].$$

Thus instantiating these auxiliary variables in a Gibbs sampler or variational mean field inference algorithm enables efficient block updates while preserving the same marginal posterior distribution $p(x \mid z)$.

3 Recurrent Switching State Space Models

The discrete states in the SLDS of Section 2.1 are generated via an *open loop*: the discrete state z_{t+1} is a function only of the preceding discrete state z_t, and $z_{t+1} \mid z_t$ is independent of the continuous state x_t. That is, if a discrete switch should occur whenever the continuous state enters a particular region of state space, the SLDS will be unable to learn this dependence.

We consider *recurrent* switching linear dynamical system (rSLDS), also called the augmented SLDS [Barber, 2006], an extension of the SLDS to model these dependencies directly. Rather than restricting the discrete states to open-loop Markovian dynamics as in Eq. (1), the rSLDS allows the discrete state transition probabilities to depend on additional covariates, and in particular on the preceding continuous latent state [Barber, 2006]. In our version of the model, the discrete states are generated as

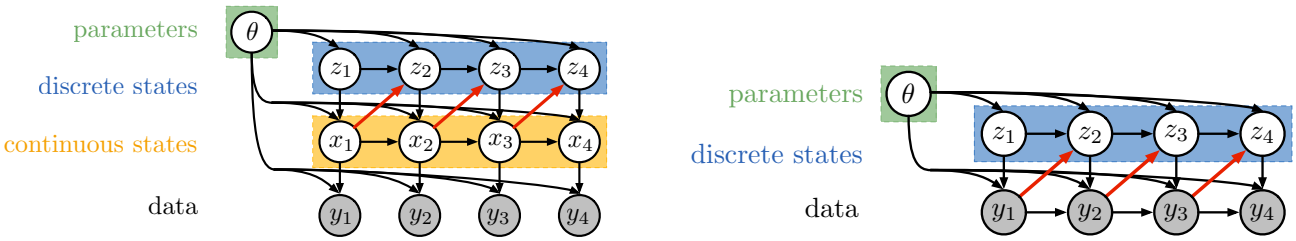
$$z_{t+1} \mid z_t, x_t, \{R_k, r_k\} \sim \pi_{\mathsf{SB}}(\nu_{t+1}),$$
$$\nu_{t+1} = R_{z_t}x_t + r_{z_t}, \tag{4}$$

where $R_k \in \mathbb{R}^{K-1 \times M}$ is a weight matrix that specifies the recurrent dependencies and $r_k \in \mathbb{R}^{K-1}$ is a bias that captures the Markov dependence of z_{t+1} on z_t. The remainder of the rSLDS generative process follows that of the SLDS from Eqs. (2)-(3). See Figure 2a for a graphical model, where the edges representing the new dependencies of the discrete states on the continuous latent states are highlighted in red.

Figure 1 illustrates an rSLDS with $K = 5$ discrete states and $M = 2$ dimensional continuous states. Each discrete state corresponds to a set of linear dynamics defined by A_k and b_k, shown in the top row. The transition probability, π_t, is a function of the previous states z_{t-1} and x_{t-1}. We show only the dependence on x_{t-1} in the bottom row. Each panel shows the conditional probability, $\Pr(z_{t+1} = k \mid x_t)$, as a colormap ranging from zero (white) to one (color). Due to the logistic stick breaking, the latent space is iteratively partitioned with linear hyperplanes.

There are several useful special cases of the rSLDS.

Recurrent ARHMM (rAR-HMM) Just as the autoregressive HMM (AR-HMM) is a special case of the SLDS in which we observe the states $x_{1:T}$ directly, we can define an analogous rAR-HMM. See Figure 2b for a graphical model, where the edges representing the dependence of the discrete states on the continuous observations are highlighted in red.



(a) rSLDS (b) rAR-HMM

Figure 2: Graphical models for the recurrent SLDS (rSLDS) and recurrent AR-HMM (rAR-HMM). Edges that represent recurrent dependencies of discrete states on continuous observations or continuous latent states are highlighted in red.

Shared rSLDS (rSLDS(s)) Rather than having separate recurrence weights (and hence a separate partition) for each value of z_t, we can share the recurrence weights as,

$$\nu_{t+1} = Rx_t + r_{z_t}.$$

Recurrence-Only (rSLDS(ro)) There is no dependence on z_t in this model. Instead,

$$\nu_{t+1} = Rx_t + r.$$

While less flexible, this model is eminently interpretable, easy to visualize, and, as we show in experiments, works well for many dynamical systems. The example in Figure 1 corresponds to this special case.

We can recover the standard SLDS by setting $\nu_{t+1} = r_{z_t}$.

4 Bayesian Inference

Adding the recurrent dependencies from the latent continuous states to the latent discrete states introduces new inference challenges. While block Gibbs sampling in the standard SLDS can be accomplished with message passing because $x_{1:T}$ is conditionally Gaussian given $z_{1:T}$ and $y_{1:T}$, the dependence of z_{t+1} on x_t renders the recurrent SLDS non-conjugate. To develop a message-passing procedure for the rSLDS, we first review standard SLDS message passing, then show how to leverage a Pólya-gamma augmentation along with message passing to perform efficient Gibbs sampling in the rSLDS. We discuss stochastic variational inference [Hoffman et al., 2013] in the supplementary material.

4.1 Message Passing

First, consider the conditional density of the latent continuous state sequence $x_{1:T}$ given all other variables, which is proportional to

$$\prod_{t=1}^{T-1} \psi(x_t, x_{t+1}, z_{t+1})\, \psi(x_t, z_{t+1}) \prod_{t=1}^{T} \psi(x_t, y_t),$$

where $\psi(x_t, x_{t+1}, z_{t+1})$ denotes the potential from the conditionally linear-Gaussian dynamics and $\psi(x_t, y_t)$ denotes the evidence potentials. The potentials $\psi(x_t, z_{t+1})$ arise from the new dependencies in the rSLDS and do not appear in the standard SLDS. This factorization corresponds to a chain-structured undirected graphical model with nodes for each time index.

We can sample from this conditional distribution using message passing. The message from time t to time $t' = t + 1$, denoted $m_{t\to t'}(x_{t'})$, is computed via

$$\int \psi(x_t, y_t)\psi(x_t, z_{t'})\psi(x_t, x_{t'}, z_{t'})m_{t''\to t}(x_t)\,\mathrm{d}x_t, \quad (5)$$

where t'' denotes $t - 1$. If the potentials were all Gaussian, as is the case without the rSLDS potential $\psi(x_t, z_{t+1})$, this integral could be computed analytically. We pass messages forward once, as in a Kalman filter, and then sample backward. This constructs a joint sample $\hat{x}_{1:T} \sim p(x_{1:T})$ in $O(T)$ time. A similar procedure can be used to jointly sample the discrete state sequence, $z_{1:T}$, given the continuous states and parameters. However, this computational strategy for sampling the latent continuous states breaks down when including the non-Gaussian rSLDS potential $\psi(x_t, z_{t+1})$.

Note that it is straightforward to handle missing data in this formulation; if the observation y_t is omitted, we simply have one fewer potential in our graph.

4.2 Augmentation for non-Gaussian Factors

The challenge presented by the recurrent SLDS is that $\psi(x_t, z_{t+1})$ is not a linear Gaussian factor; rather, it is a categorical distribution whose parameter depends nonlinearly on x_t. Thus, the integral in the message computation (5) is not available in closed form. There are a number of methods of approximating such integrals, like particle filtering [Doucet et al., 2000], Laplace approximations [Tierney and Kadane, 1986], and assumed density filtering as in Barber [2006], but here we take an alternative approach using the recently developed Pólya-gamma augmentation scheme [Polson

et al., 2013], which renders the model conjugate by introducing an auxiliary variable in such a way that the resulting marginal leaves the original model intact.

According to the stick breaking transformation described in Section 2.2, the non-Gaussian factor is

$$\psi(x_t, z_{t+1}) = \prod_{k=1}^{K} \sigma(\nu_{t+1,k})^{\mathbb{I}[z_{t+1}=k]}\, \sigma(-\nu_{t+1,k})^{\mathbb{I}[z_{t+1}>k]},$$

where $\nu_{t+1,k}$ is the k-th dimension of ν_{t+1}, as defined in (4). Recall that ν_{t+1} is linear in x_t. Expanding the definition of the logistic function, we have,

$$\psi(x_t, z_{t+1}) = \prod_{k=1}^{K-1} \frac{(e^{\nu_{t+1,k}})^{\mathbb{I}[z_{t+1}=k]}}{(1+e^{\nu_{t+1,k}})^{\mathbb{I}[z_{t+1}\geq k]}}. \quad (6)$$

The Pólya-gamma augmentation targets exactly such densities, leveraging the following integral identity:

$$\frac{(e^{\nu})^a}{(1+e^{\nu})^b} = 2^{-b} e^{\kappa\nu} \int_0^\infty e^{-\omega\nu^2/2} p_{\mathrm{PG}}(\omega \mid b, 0)\, \mathrm{d}\omega, \quad (7)$$

where $\kappa = a - b/2$ and $p_{\mathrm{PG}}(\omega \mid b, 0)$ is the density of the Pólya-gamma distribution, $\mathrm{PG}(b, 0)$, which does not depend on ν.

Combining (6) and (7), we see that $\psi(x_t, z_{t+1})$ can be written as a marginal of a factor on the augmented space, $\psi(x_t, z_{t+1}, \omega_t)$, where $\omega_t \in \mathbb{R}_+^{K-1}$ is a vector of auxiliary variables. As a function of ν_{t+1}, we have

$$\psi(x_t, z_{t+1}, \omega_t) \propto \prod_{k=1}^{K-1} \exp\left\{\kappa_{t+1,k}\, \nu_{t+1,k} - \tfrac{1}{2}\omega_{t,k}\, \nu_{t+1,k}^2\right\},$$

where $\kappa_{t+1,k} = \mathbb{I}[z_{t+1} = k] - \frac{1}{2}\mathbb{I}[z_{t+1} \geq k]$. Hence,

$$\psi(x_t, z_{t+1}, \omega_t) \propto \mathcal{N}(\nu_{t+1} \mid \Omega_t^{-1}\kappa_{t+1},\, \Omega_t^{-1}),$$

with $\Omega_t = \mathrm{diag}(\omega_t)$ and $\kappa_{t+1} = [\kappa_{t+1,1} \ldots, \kappa_{t+1,K-1}]$. Again, recall that ν_{t+1} is a linear function of x_t. Thus, after augmentation, the potential on x_t is effectively Gaussian and the integrals required for message passing can be written analytically. Finally, the auxiliary variables are easily updated as well, since $\omega_{t,k} \mid x_t, z_{t+1} \sim \mathrm{PG}(\mathbb{I}[z_{t+1} \geq k], \nu_{t+1,k})$.

4.3 Updating Model Parameters

Given the latent states and observations, the model parameters benefit from simple conjugate updates. The dynamics parameters have conjugate MNIW priors, as do the emission parameters. The recurrence weights are also conjugate under a MNIW prior, given the auxiliary variables $\omega_{1:T}$. We set the hyperparameters of these priors such that random draws of the dynamics are typically stable and have nearly unit spectral radius in expectation, and we set the mean of the recurrence bias such that states are equiprobable in expectation.

As with other many models, initialization is important. We propose a step-wise approach, starting with simple special cases of the rSLDS and building up. The supplement contains full details of this procedure.

5 Experiments

We demonstrate the potential of recurrent dynamics in a variety of settings. First, we consider a case in which the underlying dynamics truly follow an rSLDS, which illustrates some of the nuances involved in fitting these rich systems. With this experience, we then apply these models to simulated data from a canonical nonlinear dynamical system – the Lorenz attractor – and find that its dynamics are well-approximated by an rSLDS. Moreover, by leveraging the Pólya-gamma augmentation, these nonlinear dynamics can even be recovered from discrete time series with large swaths of missing data, as we show with a Bernoulli-Lorenz model. Finally, we apply these recurrent models to real trajectories on basketball players and discover interpretable, location-dependent behavioral states.

5.1 Synthetic NASCAR®

We begin with a toy example in which the true dynamics trace out ovals, like a stock car on a NASCAR® track.[1] There are four discrete states, $z_t \in \{1, \ldots, 4\}$, that govern the dynamics of a two dimensional continuous latent state, $x_t \in \mathbb{R}^2$. Fig. 3a shows the dynamics of the most likely state for each point in latent space, along with a sampled trajectory from this system. The observations, $y_t \in \mathbb{R}^{10}$ are a linear projection of the latent state with additive Gaussian noise. The 10 dimensions of y_t are superimposed in Fig. 3b. We simulated $T = 10^4$ time-steps of data and fit an rSLDS to these data with 10^3 iterations of Gibbs sampling.

Fig. 3c shows a sample of the inferred latent state and its dynamics. It recovers the four states and a rotated oval track, which is expected since the latent states are non-identifiable up to invertible transformation. Fig. 3d plots the samples of $z_{1:1000}$ as a function of Gibbs iteration, illustrating the uncertainty near the change-points.

From a decoding perspective, both the SLDS and the rSLDS are capable of discovering the discrete latent states; however, the rSLDS is a much more effective generative model. Whereas the standard SLDS has

[1] Unlike real NASCAR drivers, these states turn right.

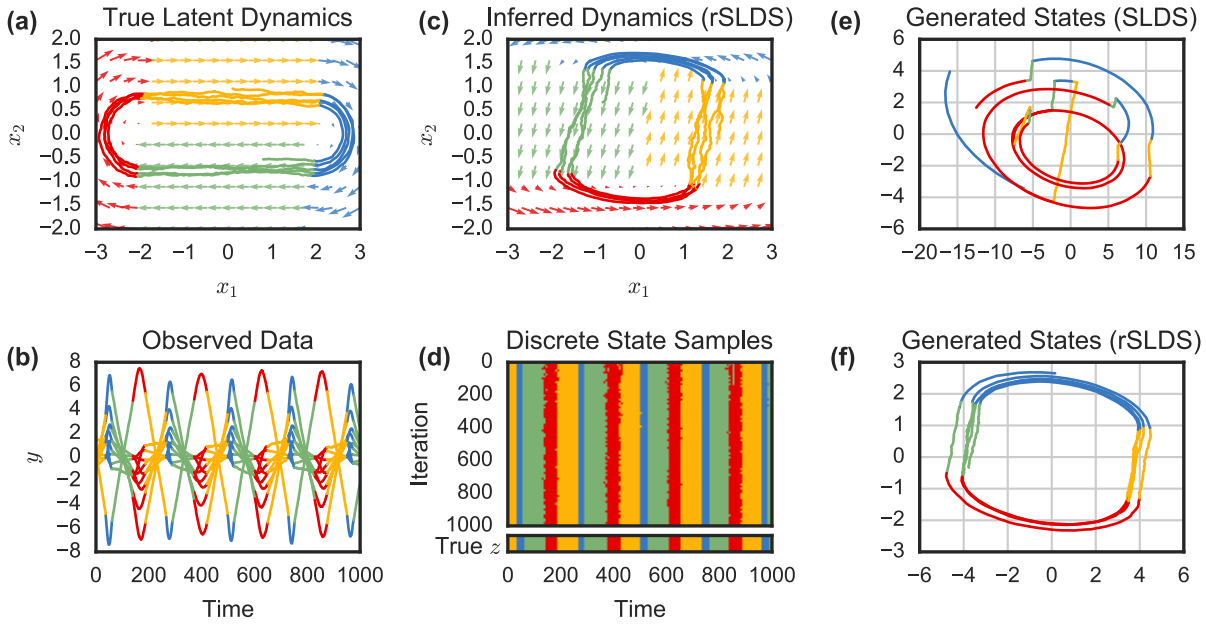


Figure 3: Synthetic NASCAR®, an example of Bayesian inference in a recurrent switching linear dynamical system (rSLDS). **(a)** In this case, the true dynamics switch between four states, causing the continuous latent state, $x_t \in \mathbb{R}^2$, to trace ovals like a car on a NASCAR® track. The dynamics of the most likely discrete state at a particular location are shown with arrows. **(b)** The observations, $y_t \in \mathbb{R}^{10}$, are a linear projection with additive Gaussian noise (colors not given; for visualization only). **(c)** Our rSLDS correctly infers the continuous state trajectory, up to affine transformation. It also learns to partition the continuous space into discrete regions with different dynamics. **(d)** Posterior samples of the discrete state sequence match the true discrete states, and show uncertainty near the change points. **(e)** Generative samples from a standard SLDS differ dramatically from the true latent states in (a), since the run lengths in the SLDS are simple geometric random variables that are independent of the continuous state. **(f)** In contrast, the rSLDS learns to generate states that shares the same periodic nature of the true model.

only a Markov model for the discrete states, and hence generates the geometrically distributed state durations in Fig 3e, the rSLDS leverages the location of the latent state to govern the discrete dynamics and generates the much more realistic, periodic data in Fig. 3f.

5.2 Lorenz Attractor

Switching linear dynamical systems offer a tractable approximation to complicated nonlinear dynamical systems. Indeed, one of the principal motivations for these models is that once they have been fit, we can leverage decades of research on optimal filtering, smoothing, and control for linear systems. However, as we show in the case of the Lorenz attractor, the standard SLDS is often a poor generative model, and hence has difficulty interpolating over missing data. The recurrent SLDS remedies this by connecting discrete and continuous states.

Fig. 4a shows the states of a Lorenz attractor whose nonlinear dynamics are given by,

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \begin{bmatrix} \alpha(x_2 - x_1) \\ x_1(\beta - x_3) - x_2 \\ x_1 x_2 - \gamma x_3 \end{bmatrix}.$$

Though nonlinear and chaotic, we see that the Lorenz attractor roughly traces out ellipses in two opposing planes. Fig. 4c unrolls these dynamics over time, where the periodic nature and the discrete jumps become clear.

Rather than directly observing the states of the Lorenz attractor, $x_{1:T}$, we simulate $N = 100$ dimensional discrete observations from a generalized linear model, $\rho_{t,n} = \sigma(c_n^{\mathsf{T}} x_t + d_n)$, where $\sigma(\cdot)$ is the logistic function, and $y_{t,n} \sim \mathrm{Bern}(\rho_{t,n})$. A window of observations is shown in Fig. 4d. Just as we leveraged the Pólya-gamma augmentation to render the continuous latent states conjugate with the multinomial discrete state samples, we again leverage the augmentation scheme to render them conjugate with Bernoulli observations. As a further challenge, we also hold out a slice of data for $t \in [700, 900)$, identified by a gray mask in the center panels. We provide more details in the supplementary material.

Fitting an rSLDS via the same procedure described above, we find that the model separates these two planes into two distinct states, each with linear, rotational dynamics shown in Fig. 4b. Note that the latent states are only identifiable up to invertible trans-

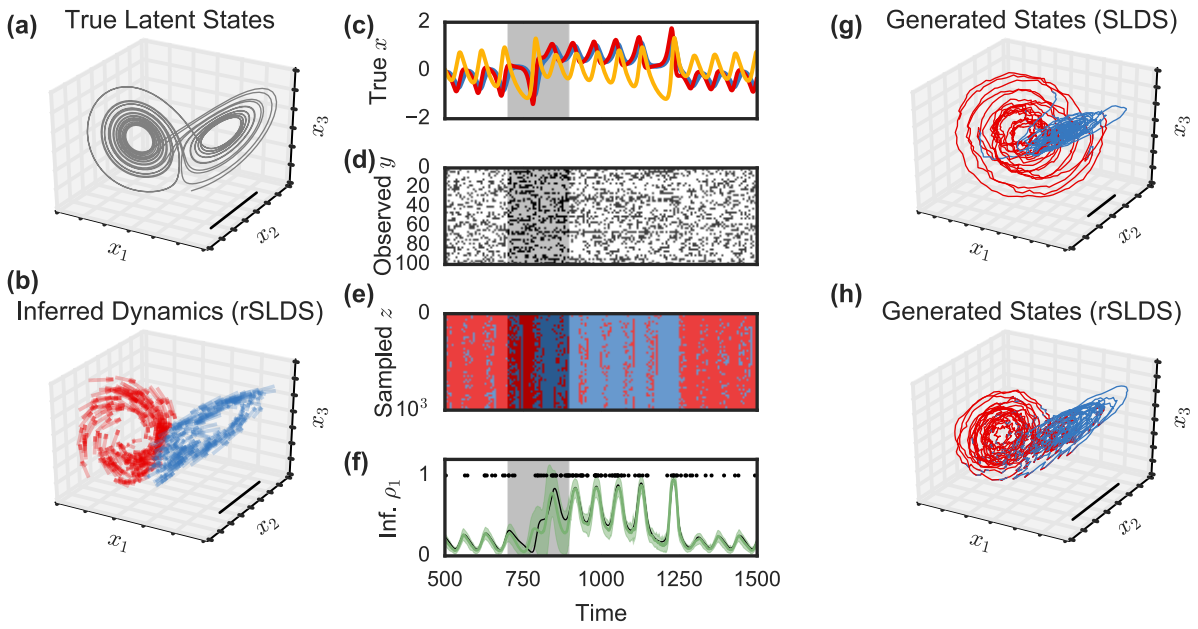


Figure 4: A recurrent switching linear dynamical system (rSLDS) applied to simulated data from a Lorenz attractor — a canonical nonlinear dynamical system. **(a)** The Lorenz attractor chaotically oscillates between two planes. Scale bar shared between (a), (b), (g) and (h). **(b)** Our rSLDS, with $x_t \in \mathbb{R}^3$, identifies these two modes and their approximately linear dynamics, up to an invertible transformation. It divides the space in half with a linear hyperplane. **(c)** Unrolled over time, we see the points at which the Lorenz system switches from one plane to the other. Gray window denotes masked region of the data. **(d)** The observations come from a generalized linear model with Bernoulli observations and a logistic link function. **(e)** Samples of the discrete state show that the rSLDS correctly identifies the switching time even in the missing data. **(f)** The inferred probabilities (green) for the first output dimension along with the true event times (black dots) and the true probabilities (black line). Error bars denote ± 3 standard deviations under posterior. **(g)** Generative samples from a standard SLDS differ substantially from the true states in (a) and are quite unstable. **(h)** In contrast, the rSLDS learns to generate state sequences that closely resemble those of the Lorenz attractor.

formation. Comparing Fig. 4e to 4c, we see that the rSLDS samples changes in discrete state at the points of large jumps in the data, but when the observations are masked, there is more uncertainty. This uncertainty in discrete state is propagated to uncertainty in the event probability, ρ, which is shown for the first output dimension in Fig. 4f. The times $\{t : y_{t,1} = 1\}$ are shown as dots, and the mean posterior probability $\mathbb{E}[\rho_{t,1}]$ is shown with ± 3 standard deviations.

The generated trajectories in Figures 4g and 4h provide a qualitative comparison of how well the SLDS and rSLDS can reproduce the dynamics of a nonlinear system. While the rSLDS is a better fit by eye, we have quantified this using posterior predictive checks (PPCs) [Gelman et al., 2013]. The SLDS, and rSLDS both capture low-order moments of the data, but one salient aspect of the Lorenz model is the switch between "sides" roughly every 200 time steps. This manifests in jumps between high probability ($\rho_1 > 0.4$) and low probability for the first output (c.f. Figure 4f). Thus, a natural test statistic, t, is the maximum duration of time spent in the high probability side. Samples from the SLDS show $t_{\mathsf{SLDS}} \sim 91 \pm 33$ time steps, dramatically underestimating the true value of $t_{\mathsf{true}} = 215$. The rSLDS samples are much more realistic, with $t_{\mathsf{rslds}} \sim 192 \pm 84$ time steps. While the rSLDS samples have high variance, it covers the true value of the statistic with its state-dependent model for discrete state transitions.

5.3 Basketball Player Trajectories

We further illustrate our recurrent models with an application to the trajectories run by five National Basketball Association (NBA) players from the Miami Heat in a game against the Brooklyn Nets on Nov. 1st, 2013. We are given trajectories, $y^{(p)}_{1:T_p} \in \mathbb{R}^{T_p \times 2}$, for each player p. We treat these trajectories as independent realizations of a "recurrence-only" AR-HMM with a shared set of $K = 30$ states. Positions are recorded every 40ms; combining the five players yields 256,103 time steps in total. We use our rAR-HMM to discover discrete dynamical states as well as the court locations in which those states are most likely to be deployed. We fit the model with 200 iteration of Gibbs sampling, initialized with a draw from the prior.

The dynamics of five of the discovered states are shown in Fig. 5 (top), along with the names we have assigned

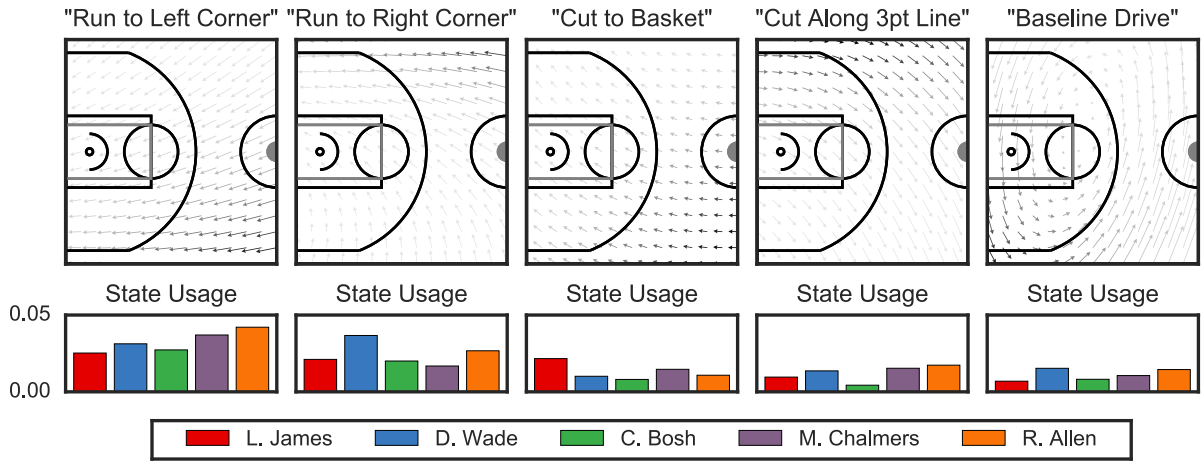


Figure 5: Exploratory analysis of NBA player trajectories from the Nov. 1, 2013 game between the Miami Heat and the Brooklyn Nets. **(Top)** When applied to trajectories of five Heat players, the recurrent AR-HMM (ro) discovers $K = 30$ discrete states with linear dynamics; five hand-picked states are shown here along with our names. Speed of motion is proportional to length of arrow. **(Bottom)** The probability with which players use the state under the posterior.

them. Below, we show the frequency with which each player uses the states under the posterior distribution. First, we notice lateral symmetry; some players drive to the left corner whereas others drive to the right. Anecdotally, Ray Allen is known to shoot more from the left corner, which agrees with the state usage here. Other states correspond to unique plays made by the players, like cuts along the three-point line and drives to the hoop or along the baseline. The complete set of states is shown in the supplementary material.

The recurrent AR-HMM strictly generalizes the standard AR-HMM, which in turn strictly generalizes AR models, and so on. Thus, barring overfitting or a inference pathologies, the recurrent model should perform at least as well as its special cases in likelihood comparisons. Here, the AR-HMM achieves a heldout log likelihood of 8.110 nats/time step, and the rAR-HMM achieves 8.124 nats/time step. Compared to a naive random walk baseline, which achieves 5.073 nats/time step, the recurrent model provides a small yet significant relative improvement (0.47%), but likelihood is only one aggregate measure of performance. It does not necessarily show that the model better captures specific salient features of the data (or that the model is more interpretable).

6 Discussion

This work is similar in spirit to the *piecewise affine* (PWA) framework in control systems [Sontag, 1981, Juloski et al., 2005, Paoletti et al., 2007]. The most relevant approximate inference work for these models is developed in Barber [2006], which uses variational approximations and assumed density filtering to perform inference in recurrent SLDS with softmax link functions. Here, because we design our models to use logistic stick-breaking, we are able to use Pólya-gamma augmentation to derive asymptotically unbiased MCMC algorithms for inferring both the latent states and the parameters.

Recurrent SLDS models strike a balance between flexibility and tractability. Composing linear systems through simple switching achieves globally nonlinear dynamics while admitting efficient Bayesian inference algorithms and easy interpretation. The Bernoulli-Lorenz example suggests that these methods may be applied to other discrete domains, like multi-neuronal spike trains [e.g. Sussillo et al., 2016]. Likewise, beyond the realm of basketball, these models may naturally apply to model social behavior in multiagent systems. These are exciting avenues for future work.

Acknowledgments

SWL is supported by the Simons Foundation SCGB-418011. ACM is supported by the Applied Mathematics Program within the Office of Science Advanced Scientific Computing Research of the U.S. Department of Energy under contract No. DE-AC02-05CH11231. RPA is supported by NSF IIS-1421780 and the Alfred P. Sloan Foundation. DMB is supported by NSF IIS-1247664, ONR N00014-11-1-0651, DARPA FA8750-14-2-0009, DARPA N66001-15-C-4032, Adobe, and the Sloan Foundation. LP is supported by the Simons Foundation SCGB-325171; DARPA N66001-15-C-4032; ONR N00014-16-1-2176; IARPA MICRONS D16PC00003.

References

Guy A Ackerson and King-Sun Fu. On state estimation in switching environments. *IEEE Transactions on Automatic Control*, 15(1):10–17, 1970.

Yaakov Bar-Shalom and Xiao-Rong Li. *Estimation and tracking*. Artech House, Boston, MA, 1993.

David Barber. Expectation correction for smoothed inference in switching linear dynamical systems. *Journal of Machine Learning Research*, 7(Nov):2515–2540, 2006.

Chaw-Bing Chang and Michael Athans. State estimation for discrete systems with switching parameters. *IEEE Transactions on Aerospace and Electronic Systems*, (3):418–425, 1978.

Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

Emily Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. Nonparametric Bayesian learning of switching linear dynamical systems. *Advances in Neural Information Processing Systems*, pages 457–464, 2009.

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.

Zoubin Ghahramani and Geoffrey E Hinton. Switching state-space models. Technical report, University of Toronto, 1996.

James D Hamilton. Analysis of time series subject to changes in regime. *Journal of econometrics*, 45(1): 39–70, 1990.

Matthew D Hoffman, David M Blei, Chong Wang, and John William Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1): 1303–1347, 2013.

Aleksandar Lj Juloski, Siep Weiland, and WPMH Heemels. A Bayesian approach to identification of hybrid systems. *IEEE Transactions on Automatic Control*, 50(10):1520–1533, 2005.

Scott W Linderman, Matthew J Johnson, and Ryan P Adams. Dependent multinomial models made easy: Stick-breaking with the Pólya-gamma augmentation. In *Advances in Neural Information Processing Systems*, pages 3438–3446, 2015.

Kevin P Murphy. Switching Kalman filters. Technical report, Compaq Cambridge Research, 1998.

Simone Paoletti, Aleksandar Lj Juloski, Giancarlo Ferrari-Trecate, and René Vidal. Identification of hybrid systems a tutorial. *European Journal of Control*, 13(2):242–260, 2007.

Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian inference for logistic models using Pólya-gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349, 2013.

Eduardo Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, 26(2):346–358, 1981.

David Sussillo, Rafal Jozefowicz, L. F. Abbott, and Chethan Pandarinath. LFADS: Latent factor analysis via dynamical systems. *arXiv preprint arXiv:1608.06315*, 2016.

Luke Tierney and Joseph B Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81(393):82–86, 1986.

Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems

Scott W. Linderman* Columbia University

Matthew J. Johnson* Harvard and Google Brain

Andrew C. Miller Harvard University

Ryan P. Adams Harvard and Google Brain

David M. Blei Columbia University

Liam Paninski Columbia University

1 Stochastic Variational Inference

The main paper introduces a Gibbs sampling algorithm for the recurrent SLDS and its siblings, but it is straightforward to derive a mean field variational inference algorithm as well. From this, we can immediately derive a stochastic variational inference (SVI) [Hoffman et al., 2013] algorithm for conditionally independent time series.

We use a structured mean field approximation on the augmented model,

$$p(z_{1:T}, x_{1:T}, \omega_{1:T}, \theta \,|\, y_{1:T}) \approx q(z_{1:T})\, q(x_{1:T})\, q(\omega_{1:T})\, q(\theta; \eta).$$

The first three factors will not be explicitly parameterized; rather, as with Gibbs sampling, we leverage standard message passing algorithms to compute the necessary expectations with respect to these factors. Moreover, $q(\omega_{1:T})$ further factorizes as,

$$q(\omega_{1:T}) = \prod_{t=1}^{T} \prod_{k=1}^{K-1} q(\omega_{t,k}).$$

To be concrete, we also expand the parameter factor,

$$q(\theta; \eta) = \prod_{k=1}^{K} q(R_k, r_k \,|\, \eta_{\mathsf{rec},k})\, q(A_k, b_k, B_k; \eta_{\mathsf{dyn},k}) \times q(C_k, d_k, D_k; \eta_{\mathsf{obs},k}).$$

The algorithm proceeds by alternating between optimizing $q(x_{1:T})$, $q(z_{1:T})$, $q(\omega_{1:T})$, and $q(\theta)$.

Updating $q(x_{1:T})$. Fixing the factor on the discrete states $q(z_{1:T})$, the optimal variational factor on the continuous states $q(x_{1:T})$ is determined by,

$$\ln q(x_{1:T}) = \psi(x_1) + \sum_{t=1}^{T-1} \psi(x_t, x_{t+1}) + \sum_{t=1}^{T-1} \psi(x_t, z_{t+1}, \omega_t) + \sum_{t=1}^{T} \psi(x_t; y_t) + c.$$

where

$$\psi(x_1) = \mathbb{E}_{q(\theta)q(z)} \ln p(x_1 \,|\, z_1, \theta) \quad (1)$$

$$\psi(x_t, x_{t+1}) = \mathbb{E}_{q(\theta)q(z)} \ln p(x_{t+1} \,|\, x_t, z_t, \theta), \quad (2)$$

$$\psi(x_t, z_{t+1}) = \mathbb{E}_{q(\theta)q(z)q(\omega)} \ln p(z_{t+1} \,|\, x_t, z_t, \omega_t, \theta), \quad (3)$$

Because the densities $p(x_1 \,|\, z_1, \theta)$ and $p(x_{t+1} \,|\, x_t, z_t, \theta)$ are Gaussian exponential families, the expectations in Eqs. (1)-(2) can be computed efficiently, yielding Gaussian potentials with natural parameters that depend on both $q(\theta)$ and $q(z_{1:T})$. Furthermore, each $\psi(x_t; y_t)$ is itself a Gaussian potential. As in the Gibbs sampler, the only non-Gaussian potential comes from the logistic stick breaking model, but once again, the Pólya-gamma augmentation scheme comes to the rescue. After augmentation, the potential as a function of x_t is,

$$\mathbb{E}_{q(\theta)q(z)q(\omega)} \ln p(z_{t+1} \,|\, x_t, z_t, \omega_t, \theta) = -\frac{1}{2} \nu_{t+1}^{\mathsf{T}} \Omega_t \nu_{t+1} + \nu_{t+1}^{\mathsf{T}} \kappa(z_{t+1}) + c.$$

Since $\nu_{t+1} = R_{z_t} x_t + r_{z_t}$ is linear in x_t, this is another Gaussian potential. As with the dynamics and observation potentials, the recurrence weights, (R_k, r_k), also have matrix normal factors, which are conjugate after augmentation. We also need access to $\mathbb{E}_q[\omega_{t,k}]$; we discuss this computation below.

After augmentation, the overall factor $q(x_{1:T})$ is a Gaussian linear dynamical system with natural parameters computed from the variational factor on the

dynamical parameters $q(\theta)$, the variational parameter on the discrete states $q(z_{1:T})$, the recurrence potentials $\{\psi(x_t, z_t, z_{t+1})\}_{t=1}^{T-1}$, and the observation model potentials $\{\psi(x_t; y_t)\}_{t=1}^{T}$.

Because the optimal factor $q(x_{1:T})$ is a Gaussian linear dynamical system, we can use message passing to perform efficient inference. In particular, the expected sufficient statistics of $q(x_{1:T})$ needed for updating $q(z_{1:T})$ can be computed efficiently.

Updating $q(\omega_{1:T})$. We have,

$$\begin{aligned}\ln q(\omega_{t,k}) &= \mathbb{E}_q \ln p(z_{t+1} \mid \omega_t, x_t) + c \\ &= -\frac{1}{2}\mathbb{E}_q[\nu_{t+1}^2]\omega_{t,k} \\ &\quad + \mathbb{E}_{q(z_{1:T})} \ln p_{\mathrm{PG}}(\omega_{t,k} \mid \mathbb{I}[z_{t+1} \geq k], 0) + c\end{aligned}$$

While the expectation with respect to $q(z_{1:T})$ makes this challenging, we can approximate it with a sample, $\hat{z}_{1:T} \sim q(z_{1:T})$. Given a fixed value $\hat{z}_{1:T}$ we have,

$$q(\omega_{t,k}) = p_{\mathrm{PG}}(\omega_{t,k} \mid \mathbb{I}[\hat{z}_{t+1} \geq k], \mathbb{E}_q[\nu_{t+1}^2]).$$

The expected value of the distribution is available in closed form:

$$\mathbb{E}_q[\omega_{t,k}] = \frac{\mathbb{I}[\hat{z}_{t+1} \geq k]}{2\mathbb{E}_q[\nu_{t+1}^2]} \tanh\left(\tfrac{1}{2}\mathbb{E}_q[\nu_{t+1}^2]\right).$$

Updating $q(z_{1:T})$. Similarly, fixing $q(x_{1:T})$ the optimal factor $q(z_{1:T})$ is proportional to

$$\exp\left\{\psi(z_1) + \sum_{t=1}^{T-1}\psi(z_t, x_t, z_{t+1}) + \sum_{t=1}^{T}\psi(z_t)\right\},$$

where

$$\begin{aligned}\psi(z_1) &= \mathbb{E}_{q(\theta)} \ln p(z_1 \mid \theta) + \mathbb{E}_{q(\theta)q(x)} \ln p(x_1 \mid z_1, \theta) \\ \psi(z_t, x_t, z_{t+1}) &= \mathbb{E}_{q(\theta)q(x_{1:T})} \ln p(z_{t+1} \mid z_t, x_t) \\ \psi(z_t) &= \mathbb{E}_{q(\theta)q(x)} \ln p(x_{t+1} \mid x_t, z_t, \theta)\end{aligned}$$

The first and third densities are exponential families; these expectations can be computed efficiently. The challenge is the recurrence potential,

$$\psi(z_t, x_t, z_{t+1}) = \mathbb{E}_{q(\theta),q(x)} \ln \pi_{\mathsf{SB}}(\nu_{t+1}).$$

Since this is not available in closed form, we approximate this expectation with Monte Carlo over x_t, R_k, and r_k. The resulting factor $q(z_{1:T})$ is an HMM with natural parameters that are functions of $q(\theta)$ and $q(x_{1:T})$, and the expected sufficient statistics required for updating $q(x_{1:T})$ can be computed efficiently by message passing in the same manner.

Updating $q(\theta)$. To compute the expected sufficient statistics for the mean field update on η, we can also use message passing, this time in both factors $q(x_{1:T})$ and $q(z_{1:T})$ separately. The required expected sufficient statistics are of the form

$$\begin{gathered}\mathbb{E}_{q(z)}\mathbb{I}[z_t = i, z_{t+1} = j], \quad \mathbb{E}_{q(z)}\mathbb{I}[z_t = i], \\ \mathbb{E}_{q(z)}\mathbb{I}[z_t = k]\mathbb{E}_{q(x)}\left[x_t x_{t+1}^{\mathsf{T}}\right], \\ \mathbb{E}_{q(z)}\mathbb{I}[z_t = k]\mathbb{E}_{q(x)}\left[x_t x_t^{\mathsf{T}}\right], \quad \mathbb{E}_{q(z)}\mathbb{I}[z_1 = k]\mathbb{E}_{q(x)}[x_1],\end{gathered} \tag{4}$$

where $\mathbb{I}[\,\cdot\,]$ denotes an indicator function. Each of these can be computed easily from the marginals $q(x_t, x_{t+1})$ and $q(z_t, z_{t+1})$ for $t = 1, 2, \ldots, T-1$, and these marginals can be computed in terms of the respective graphical model messages.

Given the conjugacy of the augmented model, the dynamics and observation factors will be MNIW distributions as well. These allow closed form expressions for the required expectations,

$$\begin{gathered}\mathbb{E}_q[A_k], \quad \mathbb{E}_q[b_k], \quad \mathbb{E}_q[A_k B_k^{-1}], \quad \mathbb{E}_q[b_k B_k^{-1}], \quad \mathbb{E}_q[B_k^{-1}], \\ \mathbb{E}_q[C_k], \quad \mathbb{E}_q[d_k], \quad \mathbb{E}_q[C_k D_k^{-1}], \quad \mathbb{E}_q[d_k D_k^{-1}], \quad \mathbb{E}_q[D_k^{-1}].\end{gathered}$$

Likewise, the conjugate matrix normal prior on (R_k, r_k) provides access to

$$\mathbb{E}_q[R_k], \qquad \mathbb{E}_q[R_k R_k^{\mathsf{T}}], \qquad \mathbb{E}_q[r_k].$$

Stochastic Variational Inference. Given multiple, conditionally independent observations of time series, $\{y_{1:T_p}^{(p)}\}_{p=1}^{P}$ (using the same notation as in the basketball experiment), it is straightforward to derive a stochastic variational inference (SVI) algorithm [Hoffman et al., 2013]. In each iteration, we sample a random time series; run message passing to compute the optimal local factors, $q(z_{1:T_p}^{(p)})$, $q(x_{1:T_p}^{(p)})$, and $q(\omega_{1:T_p}^{(p)})$; and then use expectations with respect to these local factors as unbiased estimates of expectations with respect to the complete dataset when updating the global parameter factor, $q(\theta)$. Given a single, long time series, we can still derive efficient SVI algorithms that use subsets of the data, as long as we are willing to accept minor, controllable bias [Johnson and Willsky, 2014, Foti et al., 2014].

2 Initialization

Given the complexity of these models, it is important to initialize the parameters and latent states with reasonable values. We used the following initialization procedure: (i) use probabilistic PCA or factor analysis to initialize the continuous latent states, $x_{1:T}$, and the observation, C, D, and d; (ii) fit an AR-HMM to $x_{1:T}$ in order to initialize the discrete latent states, $z_{1:T}$,

and the dynamics models, $\{A_k, Q_k, b_k\}$; and then (iii) greedily fit a decision list with logistic regressions at each node in order to determine a permutation of the latent states most amenable to stick breaking. In practice, the last step alleviates the undesirable dependence on ordering that arises from the stick breaking formulation.

As mentioned in Section 4, one of the less desirable features of the logistic stick breaking regression model is its dependence on the ordering of the output dimensions; in our case, on the permutation of the discrete states $\{1, 2, \ldots, K\}$. To alleviate this issue, we first do a greedy search over permutations by fitting a decision list to $(x_t, z_t), z_{t+1}$ pairs. A decision list is an iterative classifier of the form,

$$z_{t+1} = \begin{cases} o_1 & \text{if } \mathbb{I}[p_1] \\ o_2 & \text{if } \mathbb{I}[\neg p_1 \wedge p_2] \\ o_3 & \text{if } \mathbb{I}[\neg p_1 \wedge \neg p_2 \wedge p_3] \\ \vdots & \\ o_K & \text{o.w.,} \end{cases}$$

where $(o_1, \ldots, o_K)$ is a permutation of $(1, \ldots, K)$, and $p_1, \ldots, p_k$ are predicates that depend on (x_t, z_t) and evaluate to true or false. In our case, these predicates are given by logistic functions,

$$p_j = \sigma(r_j^\mathsf{T} x_t) > 0.$$

We fit the decision list using a greedy approach: to determine o_1 and r_1, we use maximum a posterior estimation to fit logistic regressions for each of the K possible output values. For the k-th logistic regression, the inputs are $x_{1:T}$ and the outputs are $y_t = \mathbb{I}[z_{t+1} = k]$. We choose the best logistic regression (measured by log likelihood) as the first output. Then we remove those time points for which $z_{t+1} = o_1$ from the dataset and repeat, fitting $K - 1$ logistic regressions in order to determine the second output, o_2, and so on.

After iterating through all K outputs, we have a permutation of the discrete states. Moreover, the predicates $\{r_k\}_{k=1}^{K-1}$ serve as an initialization for the recurrence weights, R, in our model.

3 Bernoulli-Lorenz Details

The Pólya-gamma augmentation makes it easy to handle discrete observations in the rSLDS, as illustrated in the Bernoulli-Lorenz experiment. Since the Bernoulli likelihood is given by,

$$\begin{aligned} p(y_t \mid z_t, \theta) &= \prod_{n=1}^{N} \text{Bern}(\sigma(c_n^\mathsf{T} x_t + d_n) \\ &= \prod_{n=1}^{N} \frac{(e^{c_n^\mathsf{T} x_t + d_n})^{y_{t,n}}}{1 + e^{c_n^\mathsf{T} x_t + d_n}}, \end{aligned}$$

we see that it matches the form of (7) with,

$$\nu_{t,n} = c_n^\mathsf{T} x_t + d_n, \quad b(y_{t,n}) = 1, \quad \kappa(y_{t,n}) = y_{t,n} - \frac{1}{2}.$$

Thus, we introduce an additional set of Pólya-gamma auxiliary variables,

$$\xi_{t,n} \sim \text{PG}(1, 0),$$

to render the model conjugate. Given these auxiliary variables, the observation potential is proportional to a Gaussian distribution on x_t,

$$\psi(x_t, y_t) \propto \mathcal{N}(C x_t + d \mid \Xi_t^{-1} \kappa(y_t), \Xi_t^{-1}),$$

with

$$\begin{aligned} \Xi_t &= \text{diag}\left([\xi_{t,1}, \ldots, \xi_{t,N}]\right), \\ \kappa(y_t) &= [\kappa(y_{t,1}), \ldots, \kappa(y_{t,N})]. \end{aligned}$$

Again, this admits efficient message passing inference for $x_{1:T}$. In order to update the auxiliary variables, we sample from their conditional distribution, $\xi_{t,n} \sim \text{PG}(1, \nu_{t,n})$.

This augmentation scheme also works for binomial, negative binomial, and multinomial obsevations as well [Polson et al., 2013].

4 Basketball Details

For completeness, Figures 1 and 2 show all $K = 30$ inferred states of the rAR-HMM (ro) for the basketball data.

References

Nicholas Foti, Jason Xu, Dillon Laird, and Emily Fox. Stochastic variational inference for hidden Markov models. In *Advances in Neural Information Processing Systems*, pages 3599–3607, 2014.

Matthew D Hoffman, David M Blei, Chong Wang, and John William Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1): 1303–1347, 2013.

Matthew J. Johnson and Alan S. Willsky. Stochastic variational inference for Bayesian time series models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1854–1862, 2014.

Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian inference for logistic models using Pólya-gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349, 2013.

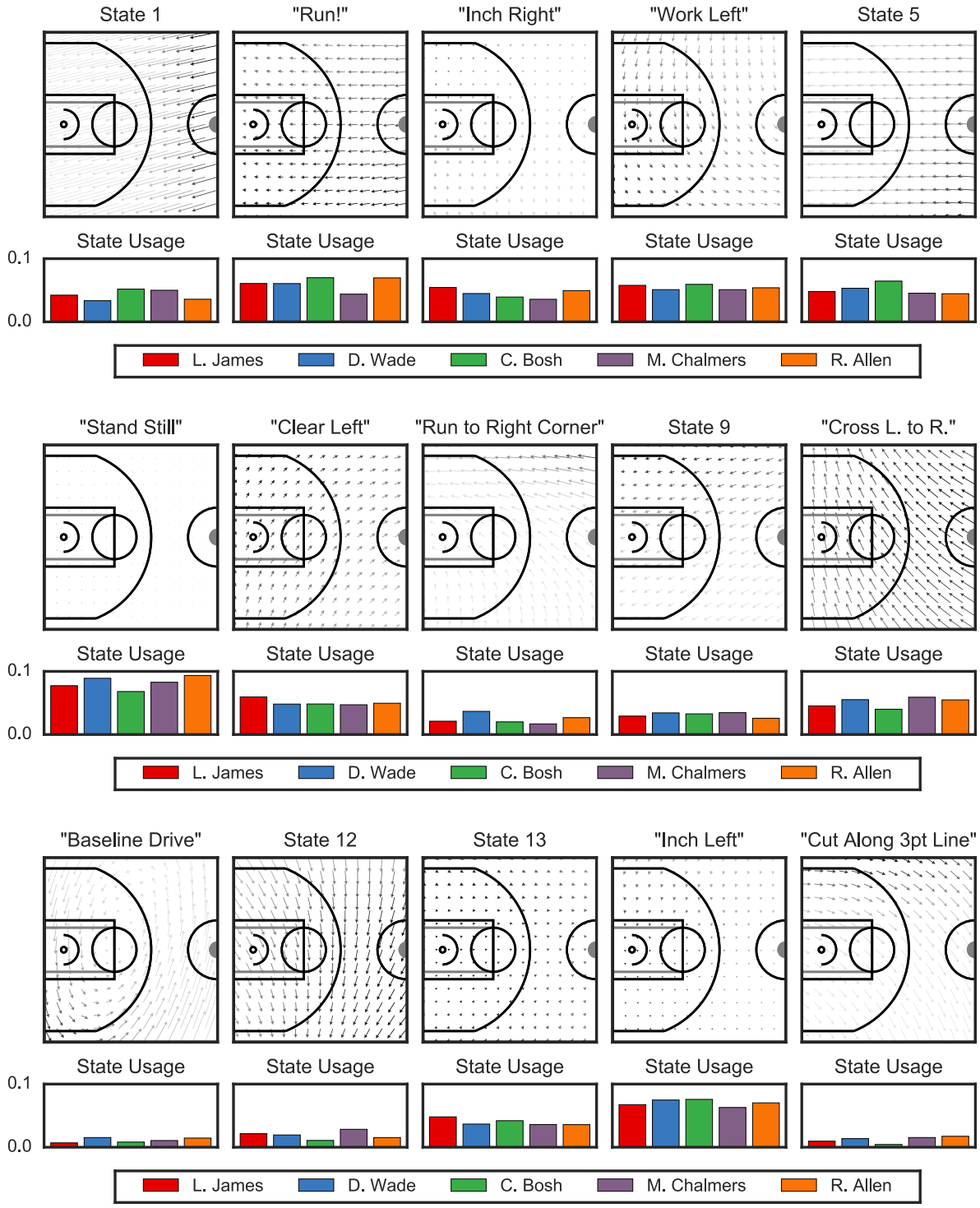


Figure 1: All of the inferred basketball states

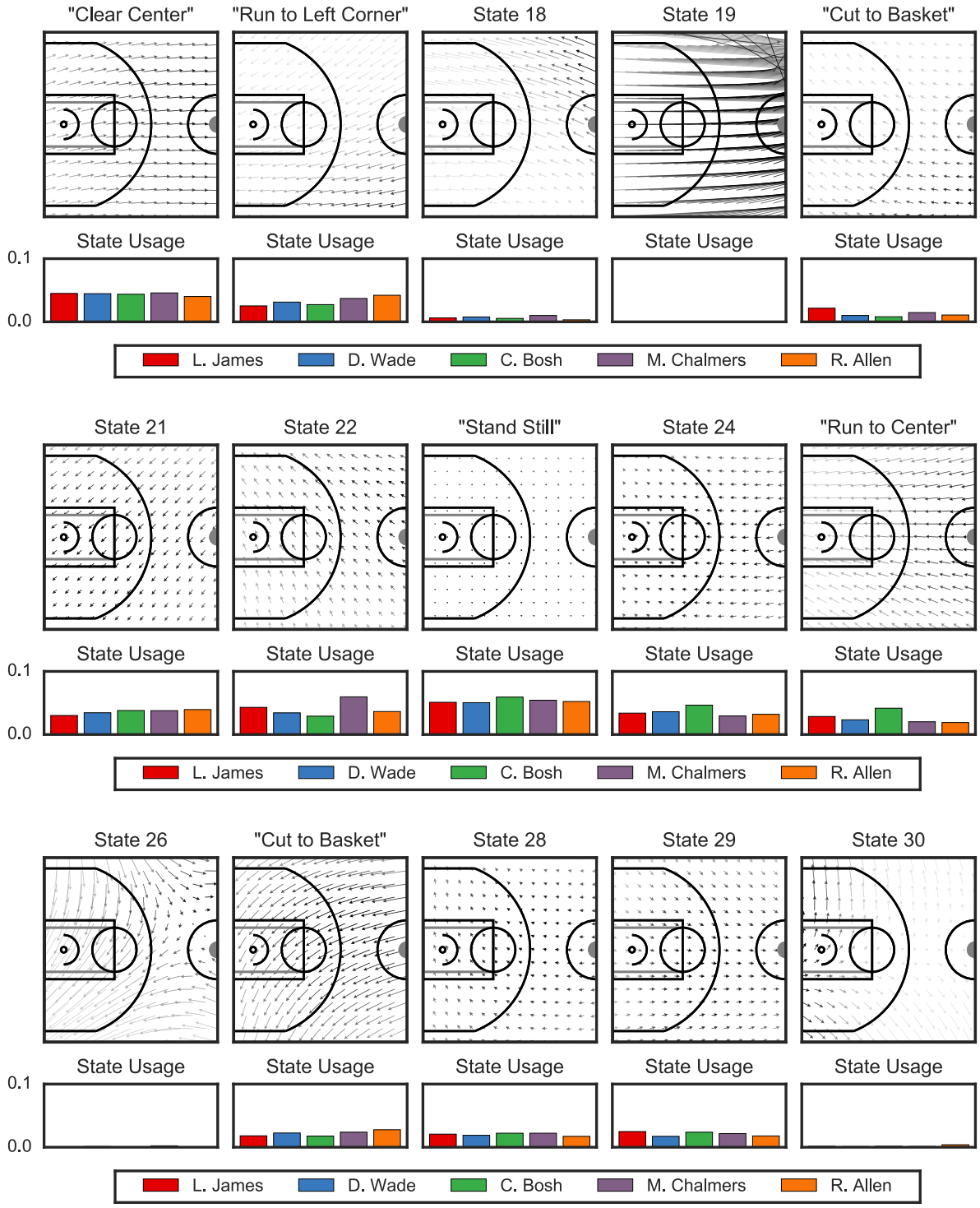


Figure 2: All of the inferred basketball states