RECURRENT SWITCHING LINEAR DYNAMICAL SYSTEMS

CHARBEL ABI YOUNES, MARVYN BAILLY, BART BOOM, ROHIN GILMAN, AND DARAN XU

Abstract. We investigate the modeling of complex dynamical systems using switching linear dynamical systems and recurrent switching linear dynamical systems. The literature shows that this method is able to estimate complex nonlinear dynamics, with an increase in performance by extending the SLDS to recurrent SLDS. The goal of this paper is to investigate if this model is able to capture the behavior of bifurcations effectively. To investigate this examine various systems where we know the dynamics.

1. Introduction

Consider a basketball player coming the court. They arrive at the three point line and have a decision to make: should they cut to the basket or run to the left corner. Depending on the positions of other players, the state in the game, or whether they are playing at home or away, the player might make a different decision. Scenarios like these produce data generated by complex, nonlinear dynamical systems. Instead of trying to produce the exact equations that lead to dynamics like those described above, we can instead understand these systems by decomposing them into multiple, simpler dynamical systems.

1.1. Switching Linear Dynamical Systems.

Switching linear dynamical systems (SLDSs) are used to split up complex, nonlinear dynamical system data into a set of linear models. By fitting the data to an SLDS, a nonlinear representation is learned. Given sufficient data and an adequate number of linear models, it can learn the global dynamics of the system.

An SLDS has the following components for each time step $t = 1, 2, \ldots, T$: a discrete latent state z_t, a continuous latent state x_t, and an observation y_t. The underlying state z_t is in the set $\{1, 2, \ldots, K\}$ and is driven by a Markov process:

$$\mathbb{P}(z_{t+1} = j \mid z_t = i) = \pi_{ij},$$

where $[\pi_{ij}]_{K\times K}$ is the transition matrix of the process. The underlying continuous state x_t is in $\mathbb{R}^M$ and is driven by conditionally linear dynamics, dependent on the discrete latent state z_{t+1}:

$$x_{t+1} = A_{z_{t+1}} x_t + b_{z_{t+1}} + v_t,$$

where $A_k \in \mathbb{R}^{M\times M}$, $b_k \in \mathbb{R}^M$, and $v_t \overset{\text{iid}}{\sim} \mathcal{N}(0, Q_{z_{t+1}})$, where $Q_k \in \mathbb{R}^{M\times M}$ is the covariance matrix of the normal distribution. Finally, the observation y_t is generated from the corresponding discrete continuous and

discrete latent states:

$$y_t = C_{z_t} x_t + d_{z_t} + w_t,$$

where $C_k \in \mathbb{R}^{N \times M}$, $d_k \in \mathbb{R}^N$, and $w_t \overset{\text{iid}}{\sim} \mathcal{N}(0, S_{z_t})$, where $S_k \in \mathbb{R}^{N \times N}$ is the covariance matrix of the normal distribution.

The library of the linear systems involved in the SLDS is denoted by

$$\theta = \{\pi_k, A_k, b_k, Q_k, C_k, d_k, S_k \mid k = 1, \ldots, K\}.$$

1.2. **Fitting SLDSs.** The most common way to perform Bayesian inference is to use Gibbs sampling, which is an iterative method that updates our belief in the parameters θ, continuous latent variables $\{x_t\}_{t=1:T}$, and the discrete latent variable $\{z_t\}_{t=1:T}$. At each iteration, we draw a sample from the conditional distribution of each variable based on the current samples of the other two variables. After convergence, we end up with estimations of θ, $\{x_t\}_{t=1:T}$ and $\{z_t\}_{t=1:T}$:

(1) Sample θ from $\mathbb{P}(\theta \mid \{z_t\}_{t=1:T}, \{x_t\}_{t=1:T})$
(2) Sample $\{x_t\}$ from $\mathbb{P}(\{x_t\}_{t=1:T} \mid \{z_t\}_{t=1:T}, \theta)$
(3) Sample $\{z_t\}$ from $\mathbb{P}(\{z_t\}_{t=1:T} \mid \{x_t\}_{t=1:T}, \theta)$
(4) Repeat steps (1-3)

The parameters sampling step can be efficiently performed by choosing appropriate prior distrbutions, to ensure the posterior has a closed-form expression, forming a conjugate pair between the prior and the posterior.

For simplicity, C, S, and d are assumed to be same for all discrete states. To perform Bayesian inference efficiently, we assume conjugate Dirichlet priors for each row of the transition matrix π_k, and conjugate matrix normal inverse Wishart (MNIW) priors for the dynamical system parameters $(A_k, b_k), Q_k$:

$$\pi_k \,\Big|\, \alpha \overset{\text{iid}}{\sim} \text{Dir}(\alpha), \quad (A_k, b_k), Q_k \,\Big|\, \lambda \overset{\text{iid}}{\sim} \text{MNIW}(\lambda),$$
$$(C, d), S \mid \eta \overset{\text{iid}}{\sim} \text{MNIW}(\eta),$$

where α, λ, and η denote hyperparameters. Given the samples latent states and observations, the posterior $\mathbb{P}(\theta \mid \{z_t\}_{t=1:T}, \{x_t\}_{t=1:T})$ benefit from simple conjugate updates.

Meanwhile, to obtain conditional samples of $\{x_t\}$ from $\mathbb{P}(\{x_t\}_{t=1:T} \mid \{z_t\}_{t=1:T}, \theta)$, and $\{z_t\}$ from $\mathbb{P}(\{z_t\}_{t=1:T} \mid \{x_t\}_{t=1:T}, \theta)$, one can use the message passing algorithm.

We start with considering the conditional density of the latent continuous state sequence $x_{1:T}$ given all other variables. Here we introduce the notation of potential: ψ maps two states or three to a nonnegative value ψ, such that $\psi(x_t, y_t) \propto \mathbb{P}(y_t \mid x_t)$ and $\psi(x_t, x_{t+1}, z_{t+1}) \propto \mathbb{P}(x_{t+1} \mid x_t, z_{t+1})$. This notation allows us to

manipulate the conditional probability and the joint probability without worrying about the normalization terms. Following this notation and the graphical model of SLDS, the conditional density of $x_{1:T}$ can be expressed as $\prod_{t=1}^{T-1} \psi\left(x_t, x_{t+1}, z_{t+1}\right) \psi\left(x_t, z_{t+1}\right) \prod_{t=1}^{T} \psi\left(x_t, y_t\right)$,

For $t \geq 2$,the message from time t to time $t+1$, denoted as $m_{t\to t+1}(x_{t+1})$, is computed by $m_{t\to t+1}(x_{t+1}) = \int \psi(x_t, y_t)\psi(x_t, x_{t+1}, z_{t+1})m_{t-1\to t}(x_t)dx_t$. The paper does not clarify how to define the initial message $m_{0\to 1}(x_1)$. We propose defining it as $m_{0\to 1}(x_1) = \psi\left(x_1, z_1\right) \psi\left(x_1, y_1\right)$. Given that $x_{1:T}$ is conditionally Gaussian given $z_{1:T}$ and $y_{1:T}$, these message passing integrals will have closed forms, allowing us to perform forward propagation to evaluate $m_{0\to 1}(x_1), m_{1\to 2}(x_2), \ldots, m_{T-1\to T}(x_T)$. After that, one can perform the sampling of $\{x_t\}_{t=1:T}$ in a backward manner:

- Sample x_T from $m_{T-1\to T}(x_T)$, $z_{1:T}$ and $y_{1:T}$
- Sample x_{T-1} from $m_{T-2\to T-1}(x_{T-1})$ and the sampled x_T, $z_{1:T}$ and $y_{1:T}$
- Sample x_{T-2} from $m_{T-3\to T-2}(x_{T-2})$ and the sampled x_{T-1}, $z_{1:T}$ and $y_{1:T}$
- continue until getting the sample of x_1

After completing this forward-backward procedure, we obtain samples of $\{x_t\}$ conditioning on $\{z_t\}_{t=1:T}$, parameters θ, and the observation $\{y_t\}_{t=1:T}$. Addtionally, we can generate samples for z_t using a discrete version of this message passing algrithm. These complete one iteration of the Gibbs sampling process.

1.3. **Recurrent Switching Linear Dynamical Systems (rSLDSs).** If a switch in the discrete state of a system should occur when the trajectory enters (or leaves) a particular region, the model will be unable to capture this behavior since the discrete state is a function only of the previous discrete state. To rectify this, we instead consider a recurrent switching linear dynamical system (rSLDS).

The key difference between an SLDS and an rSLDS is in the update of the discrete latent state, z_t. Instead of depending only on the previous discrete state, the update also depends on the continuous latent variable x_t. We can see this difference in dependencies Figure 1.

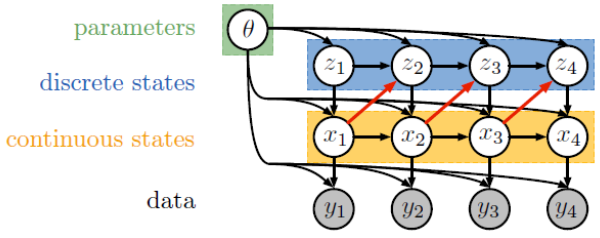


FIGURE 1. The black arrows represent the dependencies of a tradidional SLDS, and the red arrows represent the added dependencies of an rSLDS.

In the proposed model in the paper, the generation of discrete states z_t follows:

$$z_{t+1} \mid z_t, x_t, \{R_k, r_k\} \sim \pi_{\mathrm{SB}}\left(\nu_{t+1}\right),$$

$$\nu_{t+1} = R_{z_t} x_t + r_{z_t},$$

where $R_k \in \mathbb{R}^{K-1 \times M}$ represents 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 function $\pi_{\mathrm{SB}} : \mathbb{R}^{K-1} \to [0,1]^K$ maps a real vector to a normalized probability vector using the stick-breaking process, defined as follows:

$$\pi_{\mathrm{SB}}(\nu) = \begin{pmatrix} \pi_{\mathrm{SB}}^{(1)}(\nu) & \cdots & \pi_{\mathrm{SB}}^{(K)}(\nu) \end{pmatrix},$$

$$\pi_{\mathrm{SB}}^{(k)}(\nu) = \sigma\left(\nu_k\right) \prod_{j<k} \left(1 - \sigma\left(\nu_j\right)\right) = \sigma\left(\nu_k\right) \prod_{j<k} \sigma\left(-\nu_j\right),$$

for $k = 1, 2, \ldots, K-1$ and $\pi_{\mathrm{SB}}^{(K)}(\nu) = \prod_{k=1}^{K} \sigma\left(-\nu_k\right)$, where $\sigma(x) = e^x / \left(1 + e^x\right)$ denotes the logistic function. The rest of the generative process of the rSLDS follows that of the SLDS.

1.4. **Fitting rSLDSs.** Similar to SLDS, the fitting process of rSLDS is based on the Gibbs sampler. However, during the message passing stage, the message from time t to time $t+1$, denoted $m_{t \to t+1}\left(x_{t+1}\right)$, is computed via

$$\int \psi\left(x_t, y_t\right) \psi\left(x_t, z_{t+1}\right) \psi\left(x_t, x_{t+1}, z_{t+1}\right) m_{t-1 \to t}\left(x_t\right) \mathrm{d}x_t,$$

where the inclusion of $\psi\left(x_t, z_{t+1}\right)$ accounts for the introduced dependence of z_{t+1} on x_t.

With the previously defined stick-breaking function, one can derive the probability mass function $p(z \mid x) = \prod_{k=1}^{K} \sigma\left(\nu_k\right)^{\mathbb{I}[z=k]} \sigma\left(-\nu_k\right)^{[[z>k]}$, where $\mathbb{I}[\cdot]$ denotes an indicator function that takes value 1 when its argument is true and 0 otherwise. Therefore, the conditional distribution is non-Gaussian and the potential $\psi\left(x_t, z_{t+1}\right)$ is not a linear Gaussian factor. Thus, the integral in the message computation is not available in closed form, which makes the sampling of the latent continuous states challenging. In this paper, the Polya-gamma augmentation [3] is used, which introduces an auxiliary variable.

According to the stick breaking mapping, the non-Gaussian factor is

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

where ν_{t+1} is linear in x_t, and $\nu_{t+1,k}$ is the k-th dimension of ν_{t+1}. Expanding the definition of the logistic function, we have

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

An integral identity exists for polya-gamma distribution, given by:

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

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

Notice that the right hand side of $(**)$ exhibits the same form of the left hand side of $(*)$, while the right hand side of $(*)$ can be viewed as an integral of a joint density $e^{-\omega\nu^{2}/2} p_{\mathrm{PG}}(\omega \mid b, 0)$. Moreover, if we can represent this joint density as a factorized expression with $p_{\mathrm{PG}}(\omega \mid b, 0)$, the resulting factor will adopt the form $e^{-\omega\nu^{2}/2}$, enabling manipulation into a Gaussian distribution. This observation inspires us to express $\psi(x_t, z_{t+1})$ as a marginal of $\psi(x_t, z_{t+1}, \omega)$, which is a factor defined 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. In particular, we choose the auxiliary variables as $\mathbb{P}(\omega_{t,k} \mid x_t, z_{t+1}) \sim \mathrm{PG}(\mathbb{I}[z_{t+1} \geq k], \nu_{t+1,k})$. Then, we have $\psi(x_t, z_{t+1}, \omega_t) \propto \prod_{k=1}^{K-1} \exp\left\{\kappa_{t+1,k}\nu_{t+1,k} - \frac{1}{2}\omega_{t,k}\nu_{t+1,k}^{2}\right\}$, where $\kappa_{t+1,k} = \mathbb{I}[z_{t+1} = k] - \frac{1}{2}\mathbb{I}[z_{t+1} \geq k]$. Recalling that ν_{t+1} is a linear function of x_t,

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

with $\Omega_t = \mathrm{diag}(\omega_t)$ and $\kappa_{t+1} = [\kappa_{t+1,1} \ldots, \kappa_{t+1,K-1}]$.

Thus, after augmentation, the potentials on x_t becomes Gaussian, allowing the integrals required for message passing to be written analytically. As a result, the Gibbs sampling of $\{x_t\}_{t=1:T}$ can be performed efficiently. Meanwhile, in each iteration of Gibbs sampling, an additional step is needed to sample the auxiliary variables by $\omega_{t,k} \mid x_t, z_{t+1} \sim \mathrm{PG}(\mathbb{I}[z_{t+1} \geq k], \nu_{t+1,k})$. Finally, the recurrence weights, which are introduced to model the dependence of z on x, are also conjugate under a MNIW prior, given the auxiliary variables $\omega_{1:T}$. This conjugate pair ensures the efficient sampling of the paramters θ.

1.5. **Conclusions from the Paper.** In the paper [2], the authors use three examples to demonstrate the capabilities (and limitations) of SLDSs and rSLDSs. These examples are data produced by an actual rSLDS (Synthetic NASCAR), data simulated from a well-studied classical dynamical system (Lorenz Attractor), and real data (Basketball Player Trajectories).

The models performed reasonably well on the synthetic data. Figure 5 shows the true latent dynamics of the synthetic NASCAR simulation.

Figure 3 shows the states generated by both the SLDS and rSLDS fitting of the data. We can see that the rSLDS far outperformed the SLDS. This makes sense because the data was actually generated from an rSLDS, which the SLDS model will not be able to capture. Although this is just simulated data, this proof

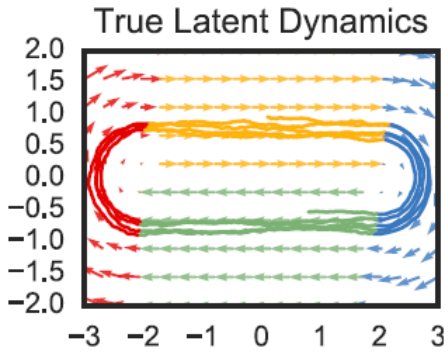
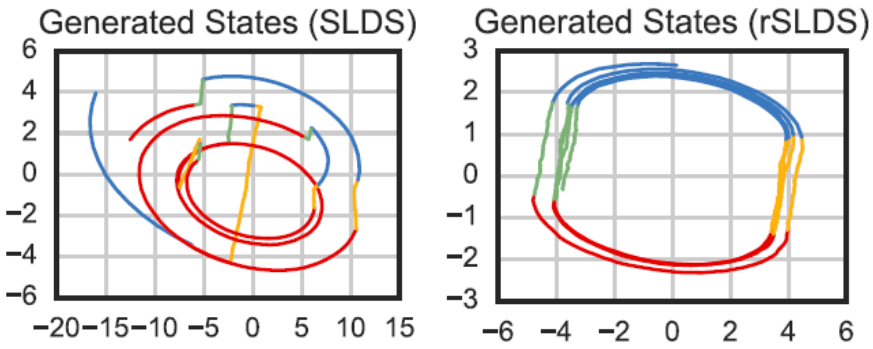


FIGURE 2. True dynamics of synthetic NASCAR.

of concept demonstrates that if the data does indeed come from an SLDS or rSLDS, the approach from the paper will generate a reasonable interpretation of that data.



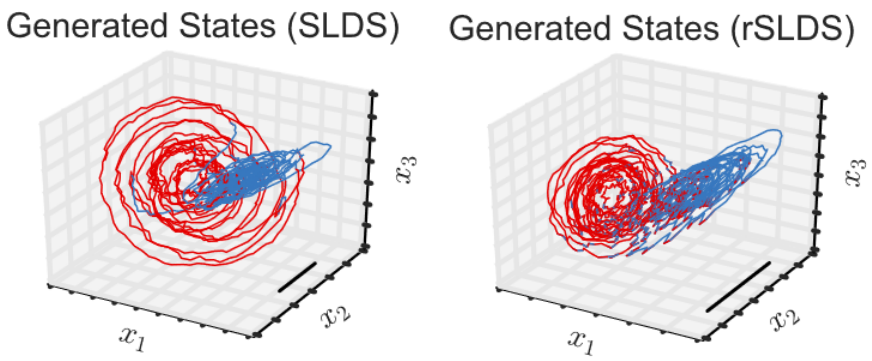
(A) SLDS (B) rSLDS

FIGURE 3. Generated states of fit data for synthetic NASCAR.

The next example the authors considered was a system generated by the Lorenz system, a classical nonlinear dynamical system that exhibits chaotic dynamics, switching between two distinct states. However, in order to demonstrate the robustness of their method in handling missing data, rather than using Gaussian observations as in a traditional rSLDS, the authors of the paper apply a generalized linear model to the simulated time series data generated by the differential equations and then apply the logistic function. Finally, they collect 100 Bernoulli observations to use as the data to fit the model on.

We can see in Figure 4 that both the SLDS model and the rSLDS model perform reasonably well. The authors found that the rSLDS did perform slightly better than the standard SLDS.

Finally, the authors analyzed data from real basketball players. The data used to fit the model were trajectories from five players in an NBA game between the Miami Heat and the Brooklyn Nets. Rather than an rSLDS, the authors fit the data to a recurrent autoregressive HMM (rAR-HMM), another generalization of the SLDS, where instead of passing through observations of the underlying continuous states, the continuous



(A) SLDS (B) rSLDS

FIGURE 4. Generated states of fit data for the Lorenz data.

states are observed directly. These states are assumed to be generated by K latent discrete states. The authors present a subset of these states, and labels them with names of basketball plays that they appear to represent. When comparing these states against those created by a random walk, the predicted states from the recurrent model perform far better; however, there is not evidence that the predicted states capture meaningful patterns in the movements of the players better than the random walk.

1.6. **New Methods.** To reproduce the results of the paper and to apply the models to more systems, we used a more up-to-date approach, as presented in [1] and [5]. These approaches are Black Box Variational Inference (BBVI) and Laplace Variational EM. These models are called state space models (SSMs).

The first approach is based on variational inference, a method from machine learning that approximates probability densities through optimization. The general idea of this approach is to first propose a family of densities that the target data may arise from and find one that generates data close to the target. The "black box" component of this approach is in reference to a class of these models that avoids any model-specific derivations, so that they can be used for a wide variety of problems. The latter approach combines variational and Laplace approximations over the underlying discrete and continuous variables. The methods for fitting an rSLDS in [2] using Polya-gamma augmentation are no longer the most up to date. The approaches presented above are more effective at fitting data to an rSLDS.

2. Reproduction of Results in the Paper

Prior to applying the recent methods to new systems, we first explore their potential as inference algorithm for the rSLDS by reproducing the results of [2]. We compare the Polyagamma approach to the variational inference algorithms by using data from an rSLDS that models the Synthetic NASCAR and solutions simulated from the chaotic Lorenz system.

2.1. **Synthetic NASCAR.** We generate a simulated dataset to represent a non-linear system that aims to imitate cars moving around a track. It consists of four states in total: two for driving along each straight section and two for navigating the semicircular turns at each end of the track. When constructing the rSLDS that models this dataset, we establish transition probabilities based solely on the preceding states. In other words, there is no reliance on the previous state z_t. Instead, each state is influenced by a constant bias r_i, which steers the transitions toward state i. This model is inherently less adaptable compared to the complete rSLDS formulation. Once we have created the rSLDS and generated a trajectory by sampling, we visualize the actual trajectory on a plot.

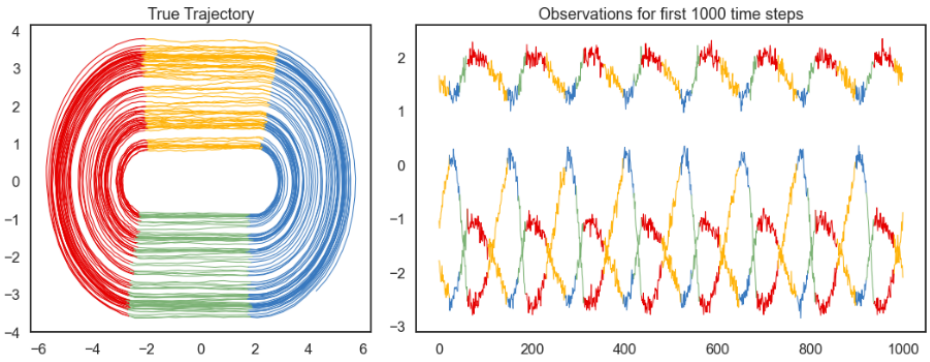


FIGURE 5. Exact trajectories of the Nascar model and the evolution of observations during the first 1000 time steps. The true dynamics switch between four states which are indicated by different colors.

The left panel displays the continuous state trajectories, while the right one presents three observation traces from the first 1000 time steps. Although our observations have ten dimensions, we have plotted only three traces to maintain clarity and reduce visual clutter. Afterward, we proceed to create a new rSLDS object and train it using the data generated previously. It is important to note that this new rSLDS model will only have access to the observations, and it will not be aware of the true states or any other underlying information. By fitting and training the new model, we can recover the behavior of the system up to an affine transformation. The nonlinear behavior is learned by the model however one might still need to apply a rotation to recover the original vector field. This clarifies why the colors of the distinct states may not consistently correspond to the original ones. The behavior of NASCAR was simulated using both BBVI and Laplace EM methods, and the outcomes are presented in Figure 6. It is evident that the estimated latent trajectories obtained through the Laplace EM method exhibit a closer match to the ground truth. Furthermore, the inferred trajectories by either of the variational algorithms demonstrate a closer alignment with the solution when compared to the Polyagamma approach.

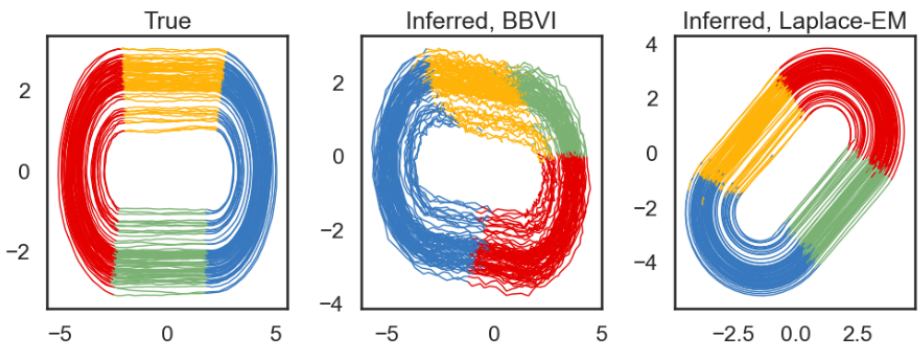


FIGURE 6. Generated trajectories of the Nascar model using rSLDS with both BBVI and Laplace EM.

2.2. **Lorenz system.** Rather than acquiring data from a linear model with Bernoulli observations, as stated in [2], we employ an ODE solver to generate data from a Lorenz system with various initial conditions. To enhance the realism of our data, we introduce perturbations by adding Gaussian noise with a mean of 0 and a variance of 1. We consider the resulting data as observations that would be utilized as input for our rSLDS model. Given the presence of two prominent states within the system, we construct our rSLDS model using 2 discrete states and a continuous latent variable of 3 dimensions. Subsequently, we fit the model to the 4-dimensional observations, and the outcomes for both the BBVI and Laplace EM approaches are condensed in Figure 7. Regarding this system, both variational algorithms performed exceptionally well in inferring the qualitative behavior, and both yielded more accurate results compared to those discussed in [2].

3. Novel Results

We continue to extend the rSLDS model using BBVI and Laplace EM inference to dynamical systems exhibiting bifurcations. We wish to see how well the model can recapture the changing dynamics of a fixed point near the bifurcation. First, we examine the Van der Pol oscillator which has a Hopf bifurcation at the origin. Next, we explore the normal form of the Bautin bifurcation which has a Bautin bifurcation at the origin.

3.1. **Van der Pol Oscillator.** The Van der Pol oscillator is a second-order differential equation of the form

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} - \mu(1-x^2)\frac{\mathrm{d}x}{\mathrm{d}t} + x = 0,$$

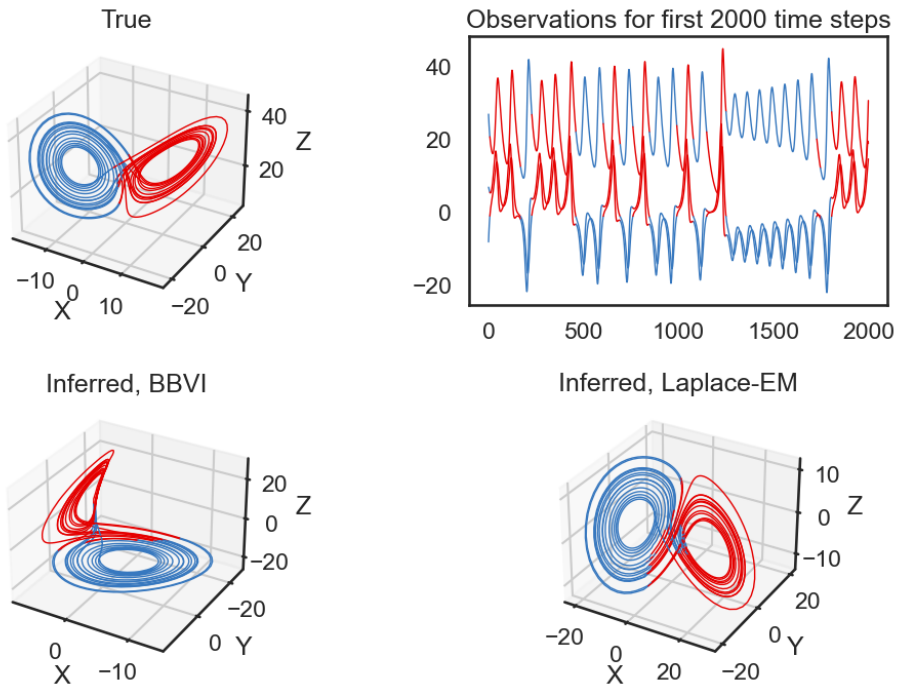


FIGURE 7. Qualitative behavior of the Lorenz system and comparison between true and generated states.

where x is the spatial coordinates and μ is a scalar parameter. The equation can be rewritten into its two-dimensional form using the transformation $y = x_t$ to get

$$\begin{cases} x_t = y, \\ y_t = \mu(1 - x^2)y - x. \end{cases}$$

Setting x_t and y_t equal to zero, we find that the Van der Pol oscillator has a fixed point at the origin $(x, y) = (0, 0)$ which is known to be a Hopf bifurcation. This means that the stability of the fixed point is stable for $\mu < 0$ and there are no limit cycles near the fixed point. When $\mu = 0$, the Van der Pol system becomes linear and the origin is a circular node. Lastly, when $\mu > 0$, the fixed point becomes unstable and a stable limit cycle around the origin appears. We demonstrate the different behaviors in Figure 8. On the left $\mu > 0$, the stable limit cycle around the fixed point can be seen. In the middle $\mu = 0$, the trajectory forms a circle around the center node. On the right $\mu < 0$, the trajectory spirals into the stable fixed point.

To fit the Van der Pol oscillator to the rSLDS model, we use an ODE solver to generate data from the Van der Pol at a fixed initial condition $(x, y) = (0, 2)$ and $t \in [0, 1000]$ using 10000 observations. To study how the model handles bifurcations, we sample the trajectory when $\mu = -1, 0, 1$ which are around and on the bifurcation. The true trajectories that are used to train the model can be seen in Figure 8. Once again, we enhance the realism of our data by adding Gaussian noise. In the $\mu = 1$ case, we expect a stable limit

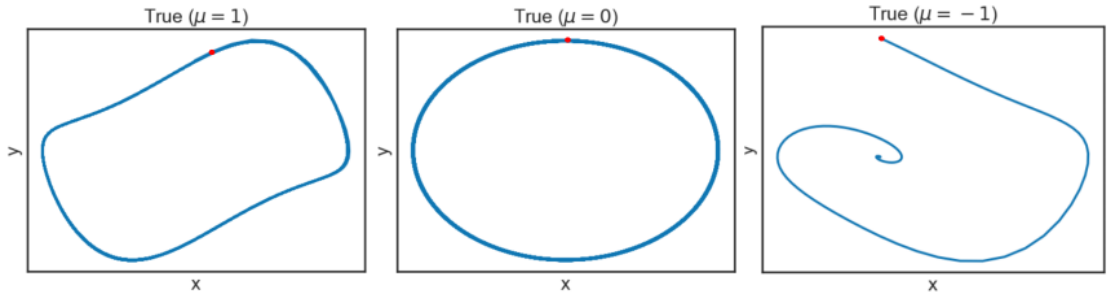


FIGURE 8. True trajectories of the Van der Pol oscillator used to fit rSLDS model over $T \in [0, 1000]$ with 1000 observations. The initial condition of $(0, 2.2)$ can be seen in red. On the left, $\mu = 1$ and the stable limit cycle can be seen. In the middle, $\mu = 0$ and the fixed point is a center. On the right, $\mu = -1$ and the fixed point is stable.

cycle that has two prominent states and thus we construct the model using two discrete states. For the sake of consistency, we set the model to have two discrete states for each μ value.

By fitting and training an rSLDS model for each μ value using the BBVI and Laplace EM methods, we recover the trajectories of the original system which are presented in Figure 9. While the rSLDS model was able to recover the trajectories for the $\mu = 0, 1$ case, the model failed to rediscover the behavior at $\mu = -1$. To handle this, we turn down the amount of Gaussian noise being added to the observations. Retraining the model we recover the dynamics of the $\mu = -1$ as seen in Figure 10. When comparing the results to the true trajectories, we see that for each μ value, the rSLDS models using Laplace EM and BBVI are both able to accurately recover the behavior of the Van der Pol system near the bifurcation. When $\mu = 1$, both models identify two latent states exceptionally well. In the $\mu = 0$ case, although the BBVI method struggles to identify the latent states by overlapping the latent states, the Laplace EM model identifies one latent state for the entire trajectory. On the other hand, when $\mu = -1$ both methods struggle to identify the latent states.

3.2. **Bautin Bifurcation.** We extend our examination of bifurcations to see how the models handle a generalized Hopf bifurcation. Consider the autonomous system of ordinary differential equations

$$\dot{y} = f(y, \beta),$$

where $y = (y_1, y_2)^T \in \mathbb{R}^2$ are the spatial coordinates and the system depends upon $\beta = (\beta_1, \beta_2) \in \mathbb{R}^2$. Supposing the system has a fixed point on the origin, the fixed point is said to experience a generalized Hopf bifurcation known as a Bautin bifurcation if specific conditions upon the first and second Lypanonuv coefficients are met. The details are omitted as they are not deemed insightful to the paper. When these

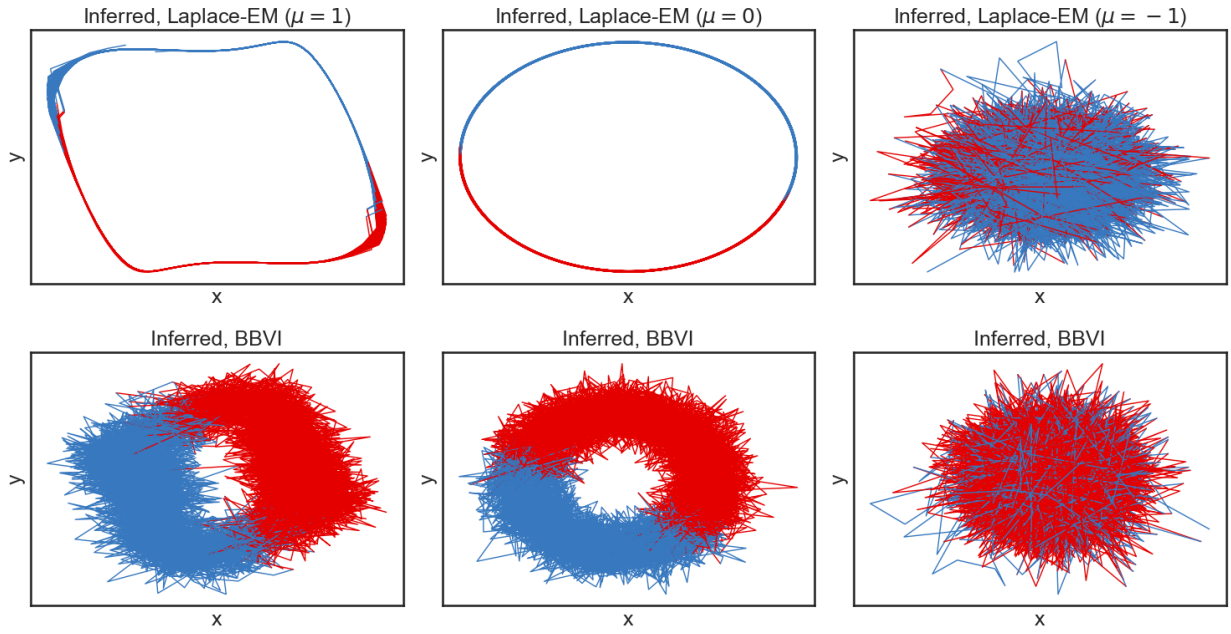


FIGURE 9. Recovered trajectories of the Van der Pol oscillator from rSLDS model using the Laplace-EM and BBVI methods with Gaussian noise present. In (A), $\mu = 1$ and the Laplace-EM recovers the system with two latent states, seen in blue and red, but BBVI has a high amount of noise. In (B), $\mu = 0$ and the same results occur as in (A). In (C), $\mu = -1$ and both methods fail to recover the true behavior.

conditions are met, $\dot{y}$ is locally equivalent near the origin to

$$\begin{cases} \dot{y_1} = \beta_1 y_1 - y + \beta_2 y_1 \left(y_1^2 + y^2\right) + \sigma y_1 \left(y_1^2 + y^2\right)^2, \\ \dot{y_2} = y_1 + \beta_1 y + \beta_2 y \left(y_1^2 + y^2\right) + \sigma y \left(y_1^2 + y^2\right)^2, \end{cases}$$

which is the normal form of the Bautin bifurcation. Here $\sigma = \pm 1$ is related to the sign of the first Lypanonuv coefficient.

For the work of the paper, we will consider $\dot{y}$ to be the normal form of the Bautin bifurcation and observe how the system handles the three following behaviors around the bifurcation point. When $\beta_1, \beta_2 < 0$, the origin is stable with an unstable limit cycle around it. When $\beta_1, \beta_2 > 0$, the fixed point becomes stable and the limit cycle around unstable. In the specific subregion of $-1/4\beta_2^2 < \beta_1 < 0$ and $\beta_2 > 0$, a second limit cycle emerges around the stable fixed point. The outer limit cycle is stable while the inner limit cycle is unstable. We demonstrate this behavior in Figure 11. On the left, $\beta_1 = \beta_2 < 0$ and the trajectory spirals into the stable fixed point and away from the unstable limit cycle. In the middle $\beta_1 = \beta_2 > 0$ and the

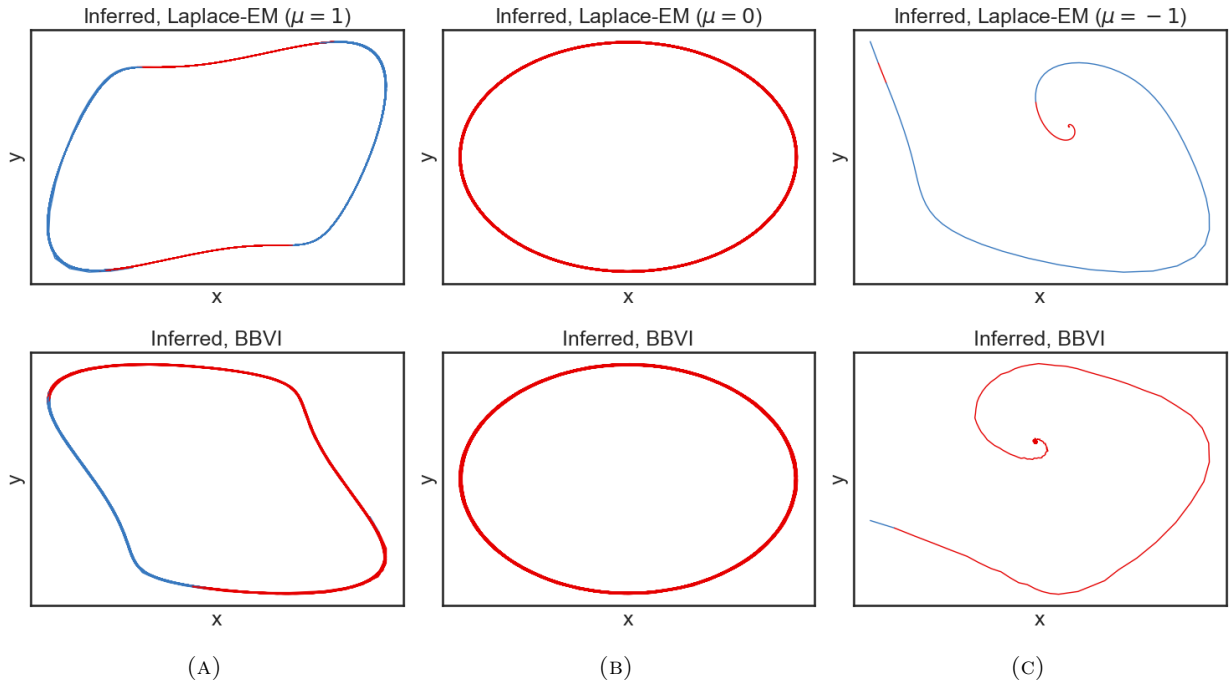


FIGURE 10. Recovered trajectories of the Van der Pol oscillator from rSLDS model using the Laplace-EM and BBVI methods with a low amount of Gaussian noise present. In (A), $\mu = 1$ and while both methods accurately recover the true trajectories, Laplace-EM places the latent states with higher accuracy. In (B), $\mu = 0$ and the same results occur as in (A). While from the graph it is not evident, BBVI fails to place the latent states accurately and rather overlaps them. In (C), $\mu = -1$ and both methods fail to recover the latent states that are insightful.

trajectory travels to the stable limit cycle away from the unstable fixed point. On the right, $\beta_1 = -1/4\beta_2^2$ and $\beta_2 > 0$ and we see the trajectory spiraling away from the unstable limit cycle into the stable limit cycle.

To fit the Bautin system to the rSLDS model, we follow the same procedure as outlined for the Van der Pol. We will once again use an ODE solver on $\dot{y}$ for a fixed initial condition $(y_1, y_2) = (0, 0.55)$ and $t \in [0, 1000]$ using 10000 observations to create the sample data for the model. We used parameter values $(\beta_1, \beta_2) \in \{(-0.1, -0.1), (0.2, 0.2), (-1/2, \sqrt{2})\}$ which are near the Bautin bifurcation to get the true trajectories seen in Figure 11. Note that once again, the amount of noise added to the system is reduced to help the rSLDS model rediscover the dynamics. Fitting the rSLDS model using BBVI and Laplace EM, both methods recover the dynamics near the bifurcation point up to an affine transformation as seen in Figure 12. When $(\beta_1, \beta_2) = (-0.1, -0.1)$, both methods capture the dynamics of the stable fixed point although the BBVI results are less smooth. When $(\beta_1, \beta_2) = (0.2, 0.2)$, both methods recover the stable limit cycle. Finally, when $(\beta_1, \beta_2) = (-1/2, \sqrt{2})$, both methods show the presence of an unstable limit cycle within a stable limit

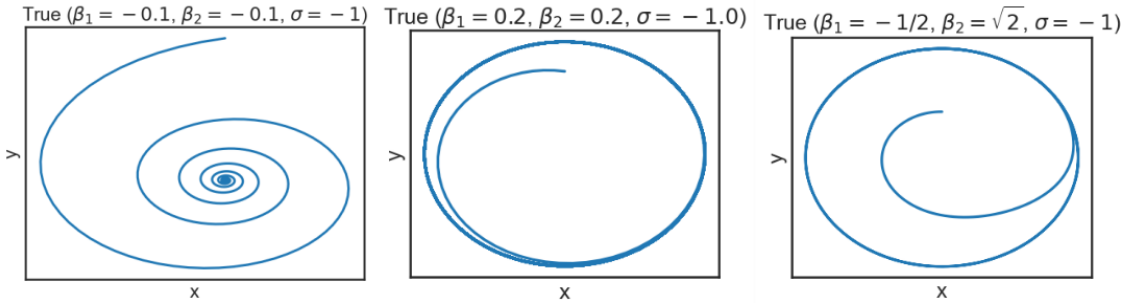


FIGURE 11. True trajectories of the Bautin system used to fit rSLDS model over $T \in [0, 1000]$ with 1000 observations. The initial condition of $(0, 0.55)$ and $\sigma = -1$ is used for each trajectory. On the left, $\beta_1 = \beta_2 = -0.1$ and the trajectory spirals into the stable fixed point away from the unstable limit cycle. In the middle, $\beta_1 = \beta_2 = 0.2$ and the trajectory moves away from the unstable fixed point and into the stable limit cycle. On the right $\beta_1 = -1/2$ and $\beta_2 = \sqrt{2}$ and the trajectory spirals away from the unstable limit cycle on the inside and towards the stable limit cycle on the outside.

cycle. In each of the cases, both the Laplace EM and BBVI methods struggled to place two latent spaces and so we omitted the colors of the different states from the figure.

3.3. **Remarks.** For both the methods, Laplace EM and BBVI, of the rSLDS model, the model accurately rediscovers the local dynamics near and on the bifurcation point but struggles to place two latent states in meaningful positions. More specially, fitting the model to the Van der Pol oscillator near the Hopf bifurcation highlights how the Laplace EM method outperforms the BBVI method at smoothly recreating the dynamics of the trajectory. Here the Laplace EM method is also able to place meaningful latent states for two of the cases near the bifurcation. Fitting the model to the Bautin system once again shows that the Laplace method outperforms the BBVI. Nevertheless, in both situations, the rSLDS model is only able to fit the dynamics of the model accurately when there is a low amount of noise present in the observation data.

4. CONCLUSIONS

The rSLDS model, in all the cases that we examined, was able to capture the dynamics of complex canonical dynamical systems, at least under certain conditions. When the dynamics of the underlying system feature a stable fixed point or an unstable orbit, the model is only able to reproduce the dynamics if the noise on the observations is sufficiently small. However, if the system exhibits a stable orbit or limit cycle, the rSLDS model is particularly effective at reproducing the dynamics of the underlying system.

Another potential downside of this model is that it is only able to reproduce the dynamics associated with a single trajectory, at least with the current fitting methods. Since the model depends on time series data, we cannot append multiple trajectories in the same system to fit the model for broader initial conditions. This

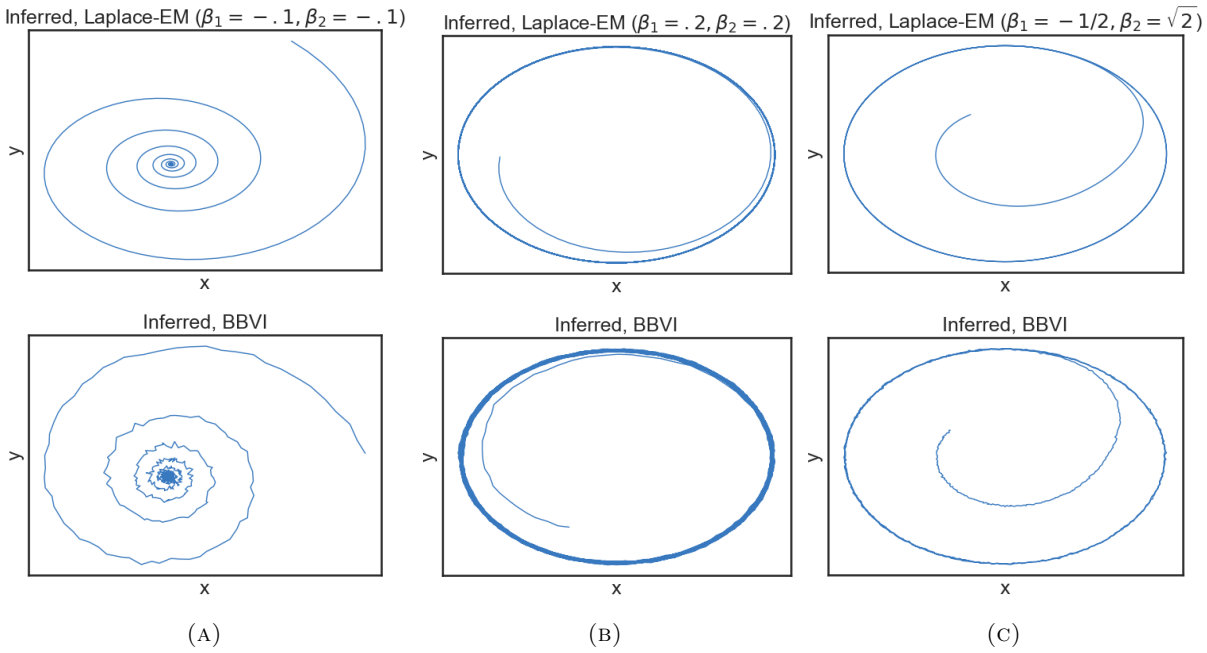


FIGURE 12. Recovered trajectories of the Bautin system from the rSLDS model using the Laplace-EM and BBVI methods with a low amount of Gaussian noise present. In (A), $\beta_1 = \beta_2 = -0.1$ and the Laplace-EM method smoothly recovers the true trajectory while the BBVI recovers a trajectory with noise. In (B), $\beta_1 = \beta_2 = 0.2$ and both methods perform similarly to (A). Note that once again it can be seen that the BBVI has more noise due to the variance (thickness of the line) of the limit cycle. In (C) $\beta_1 = -1/2$ and $\beta_2 = \sqrt{2}$ and the same results from (A) and (B) occur.

makes the method limited in learning the global dynamics and showing where the boundaries of stability are especially the position of unstable fixed points and periodic orbits.

REFERENCES

[1] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877, April 2017. arXiv:1601.00670 [cs, stat].

[2] Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 914–922. PMLR, April 2017. ISSN: 2640-3498.

[3] Scott Linderman, Matthew J Johnson, and Ryan P Adams. Dependent Multinomial Models Made Easy: Stick-Breaking with the Polya-gamma Augmentation. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[4] J. H. MacKe, L. Buesing, and M. Sahani. Estimating state and parameters in state space models of spike trains. pages 137–159. Cambridge University Press, September 2015. Book Title: Advanced State Space Methods for Neural and Clinical Data Edition: 1.

[5] David M. Zoltowski, Jonathan W. Pillow, and Scott W. Linderman. Unifying and generalizing models of neural dynamics during decision-making, January 2020. arXiv:2001.04571 [q-bio, stat].

5. Appendix

All the code used to generate the figures in this report can be found here:

https://github.com/rohingilman/AMATH575-rSLDSs

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. Copyright 2017 by the author(s).

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

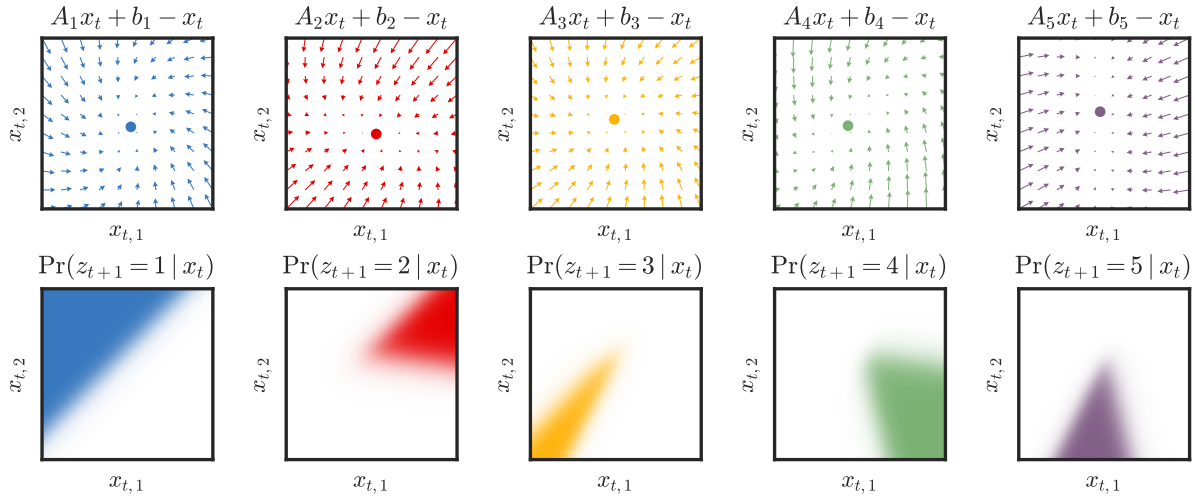


Figure 1: A draw from the prior over recurrent switching linear dynamical systems with $K = 5$ discrete states shown in different colors. **(Top)** The linear dynamics of each latent state. Dots show the fixed point $(I - A_k)^{-1}b_k$. **(Bottom)** The conditional $p(z_{t+1}|x_t)$ plotted as a function 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

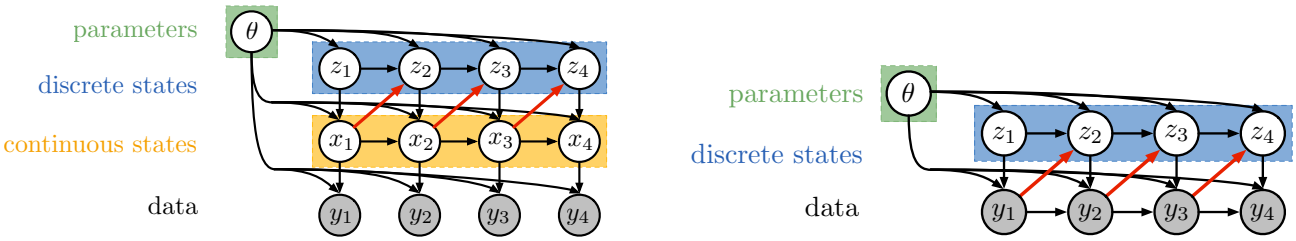
$$z_{t+1} \mid z_t, x_t, \{R_k, r_k\} \sim \pi_{\mathsf{SB}}(\nu_{t+1}),$$
$$\nu_{t+1} = R_{z_t}x_t + r_{z_t}, \tag{4}$$

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.

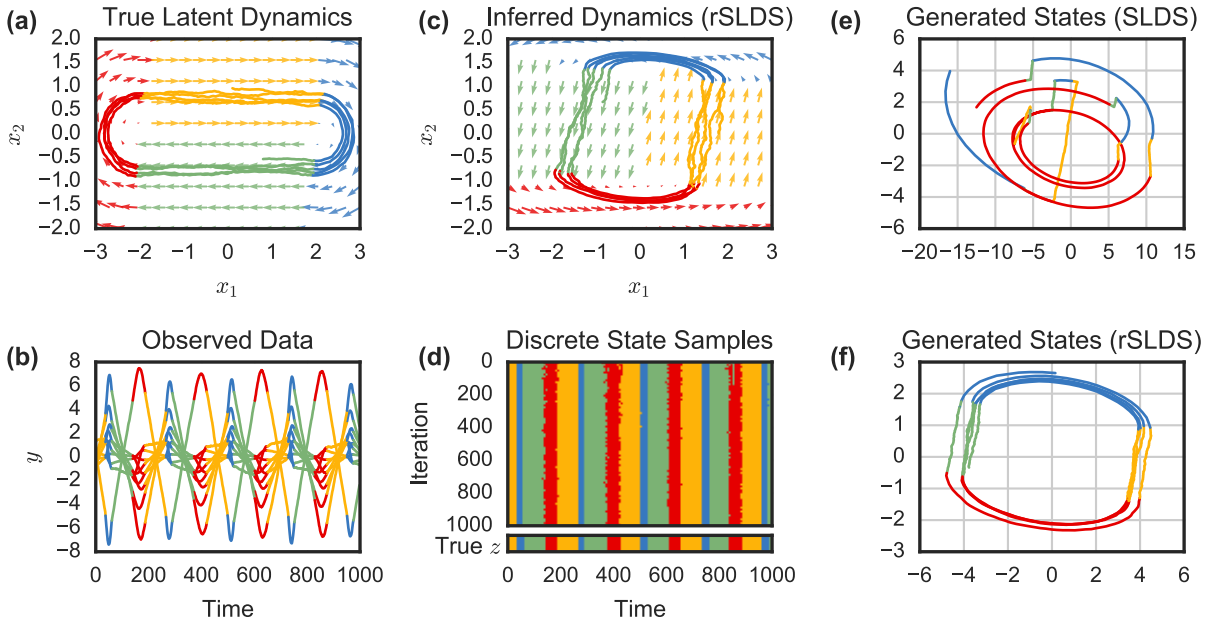


Figure 3: Synthetic NASCAR®, an example of Bayesian inference in a recurrent switching linear dynamical system (rSLDS). **(a)** In this case, the true dynamics switch between four states, causing the continuous latent state, $x_t \in \mathbb{R}^2$, to trace ovals like a car on a NASCAR® track. The dynamics 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-

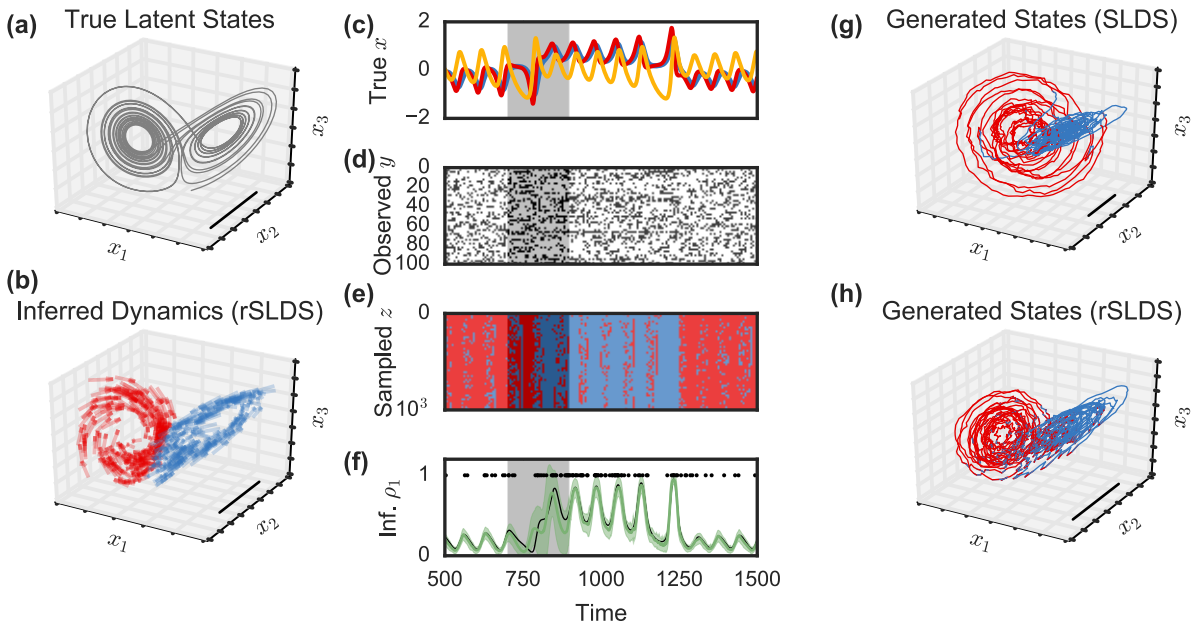


Figure 4: A recurrent switching linear dynamical system (rSLDS) applied to simulated data from a Lorenz attractor — a canonical nonlinear dynamical system. **(a)** The Lorenz attractor chaotically oscillates between two planes. Scale bar shared between (a), (b), (g) and (h). **(b)** Our rSLDS, with $x_t \in \mathbb{R}^3$, 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

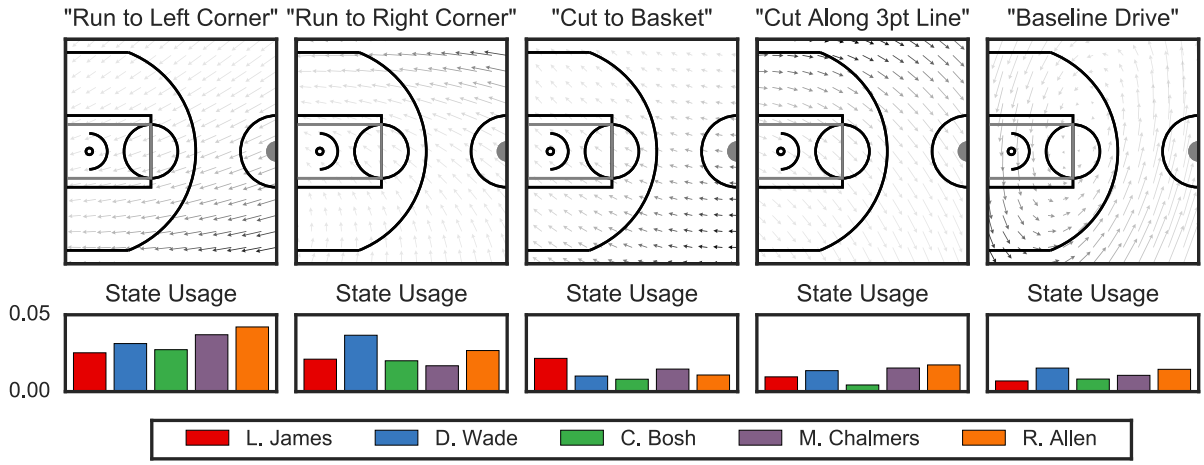


Figure 5: Exploratory analysis of NBA player trajectories from the Nov. 1, 2013 game between the Miami Heat and the Brooklyn Nets. **(Top)** When applied to trajectories of five Heat players, the recurrent AR-HMM (ro) discovers $K = 30$ discrete states with linear dynamics; five hand-picked states are 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.