TREE-STRUCTURED RECURRENT SWITCHING LINEAR DYNAMICAL SYSTEMS FOR MULTI-SCALE MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Many real-world systems studied are governed by complex, nonlinear dynamics. By modeling these dynamics, we can gain insight into how these systems work, make predictions about how they will behave, and develop strategies for controlling them. While there are many methods for modeling nonlinear dynamical systems, existing techniques face a trade off between offering interpretable descriptions and making accurate predictions. Here, we develop a class of models that aims to achieve both simultaneously, smoothly interpolating between simple descriptions and more complex, yet also more accurate models. Our probabilistic model achieves this multi-scale property through of a hierarchy of locally linear dynamics that jointly approximate global nonlinear dynamics. We call it the tree-structured recurrent switching linear dynamical system. To fit this model, we present a fully-Bayesian sampling procedure using Pólya-Gamma data augmentation to allow for fast and conjugate Gibbs sampling. Through a variety of synthetic and real examples, we show how these models outperform existing methods in both interpretability and predictive capability.

1 INTRODUCTION

Complex systems can often be described at multiple levels of abstraction. A computer program can be characterized by the list of functions it calls, the sequence of statements it executes, or the assembly instructions it sends to the microprocessor. As we zoom in, we gain an increasingly nuanced view of the system and its dynamics. The same is true of many natural systems. For example, brain activity can be described in terms of high-level psychological states or via detailed ion channel activations; different tasks demand different levels of granularity. One of our principal aims as scientists is to identify appropriate levels of abstraction for complex natural phenomena and to discover the dynamics that govern how these systems behave at each level of resolution.

Modern machine learning offers a powerful toolkit to aid in modeling the dynamics of complex systems. Bayesian state space models and inference algorithms enable posterior inference of the latent states of a system and the parameters that govern their dynamics (Särkkä, 2013; Barber et al., 2011; Doucet et al., 2001). In recent years, this toolkit has been expanded to incorporate increasingly flexible components like Gaussian processes (Frigola et al., 2014) and neural networks (Chung et al., 2015; Johnson et al., 2016; Gao et al., 2016; Krishnan et al., 2017) into probabilistic time series models. In neuroscience, sequential autoencoders offer highly accurate models of brain activity (Pandarinath et al., 2018). However, while these methods offer state of the art predictive models, their dynamics are specified at only the most granular resolution, leaving the practitioner to tease out higher level structure post hoc.

Here we propose a probabilistic generative model that provides a multi-scale view of the dynamics through a hierarchical architecture. We call it the *tree-structured recurrent switching linear dynamical system*, or TrSLDS. The model builds on the recurrent SLDS (Linderman et al., 2017) to approximate latent nonlinear dynamics through a hierarchy of locally linear dynamics. Once fit, the TrSLDS can be queried at different levels of the hierarchy to obtain dynamical descriptions at multiple levels of resolution. As we proceed down the tree, we obtain higher fidelity, yet increasingly complex, descriptions. Thus, depth offers a simple knob for trading off interpretability and flexibility. The key contributions are two-fold: first, we introduce a new form of tree-structured stick breaking for multinomial models that strictly generalizes the sequential stick breaking of the original rSLDS,

while still permitting Pólya-gamma data augmentation (Polson et al., 2013) for efficient posterior inference; second, we develop a hierarchical prior that links dynamics parameters across levels of the tree, thereby providing descriptions that vary smoothly with depth.

The paper is organized as follows. Section 2 provides background material on switching linear dynamical systems and their recurrent variants. Section 3 presents our tree-structured model and Section 4 derives an efficient fully-Bayesian inference algorithm for the latent states and dynamics parameters. Finally, in Section 5 we show how our model yields multi-scale dynamics descriptions for synthetic data from two standard nonlinear dynamical systems—the Lorenz attractor and the FitzHugh-Nagumo model of neural action potentials—as well as for a real dataset of neural responses to visual stimuli in a macaque monkey.

2 Background

Let $x_t \in \mathbb{R}^{d_x}$ and $y_t \in \mathbb{R}^{d_y}$ denote the latent state and the observation of the system at time t respectively. The system can be described using a state-space model:

$$x_t = f(x_{t-1}, w_t; \Theta), \quad w_t \sim \mathrm{F}_w \quad \textit{(state dynamics)} \tag{1}$$

$$y_t = g(x_t, v_t; \Psi), \quad v_t \sim \mathrm{F}_v \quad \textit{(observation)} \tag{2}$$

where Θ are the dynamics parameters, Ψ are the emission (observation) parameters, and w_t and v_t are the state and observation noise respectively. For simplicity, we restrict ourselves to systems of the form:

$$x_t = f(x_{t-1}; \Theta) + w_t, \quad w_t \sim \mathcal{N}(0, Q), \tag{3}$$

$$y_t = g(x_t; \Psi) + v_t, \quad v_t \sim \mathcal{N}(0, S). \tag{4}$$

If the state space model is completely specified then recursive Bayesian inference can be applied to obtain an estimate of the latent states using the posterior $p\left(x_{0:T} | y_{1:T}\right)$ (Doucet et al., 2001). However in many applications, the parametric form of the state space model is unknown. While there exist methods that perform smoothing to obtain an estimate of $x_{0:T}$ (Barber, 2006; Fox et al., 2009; Djuric & Bugallo, 2006), we are often interested in not only obtaining an estimate of the continuous latent sates but also in learning the dynamics $f(\cdot; \Theta)$ that govern the dynamics of the system. This is known as the dual estimation problem Haykin (2001).

In the simplest case, we can take a parametric approach to solving this dual estimation problems. When $f(\cdot; \Theta)$ and $g(\cdot; \Psi)$ are assumed to be linear functions, the posterior distribution over latent states is available in closed form and the parameters can be learned via expectation maximization. On the other hand, we have nonparametric methods that use Gaussian processes and neural networks to learn highly nonlinear dynamics and observations (Zhao & Park, 2016; 2017a; Frigola et al., 2014; Sussillo et al., 2016). Switching linear dynamical systems (SLDS) (Ackerson & Fu, 1970; Chang & Athans, 1978; Hamilton, 1990; Ghahramani & Hinton, 1996; Murphy, 1998) balance between these two extremes, approximating the dynamics by stochastically transitioning between a small number of linear regimes.

2.1 Switching Linear Dynamical Systems

SLDS approximate nonlinear dynamics by switching between a discrete set of linear regimes. A discrete latent state $z_t \in \{1, \ldots, K\}$ determines the linear dynamics at time t,

$$x_t = x_{t-1} + A_{z_t} x_{t-1} + b_{z_t} + w_t, \quad w_t \sim \mathcal{N}(0, Q_{z_t}) \tag{5}$$

where $A_k, Q_k \in \mathbb{R}^{d_x \times d_x}$ and $b_k \in \mathbb{R}^{d_x}$ for $k = 1, \ldots, K$. Typically, z_t is endowed with Markovian dynamics, $\Pr(z_t | z_{t-1} = k) = \boldsymbol{\pi_k}$. The conditionally linear dynamics allow for fast and efficient learning of the model and can utilize the learning tools developed for linear systems (Haykin, 2001). While SLDS can estimate the continuous latent states $x_{0:T}$, the assumption of Markovian dynamics for the discrete latent states severely limits their generative capacity.

2.2 Recurrent Switching Linear Dynamical Systems

Recurrent switching linear dynamical systems (rSLDS) (Linderman et al., 2017), also known as augmented SLDS (Barber, 2006), are an extension of SLDS where the transition density of the

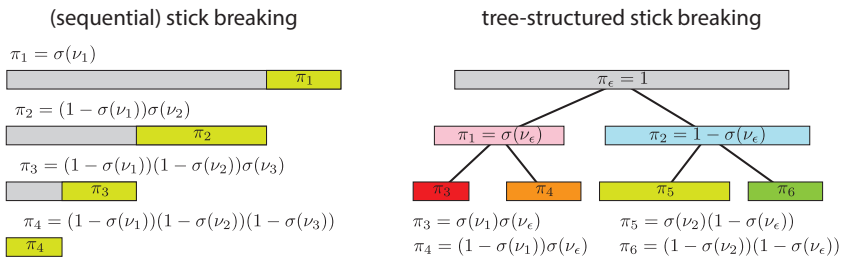

Figure 1: State probability allocation through stick-breaking in standard rSLDS and the TrSLDS.

discrete latent state depends on the previous location in latent space

$$z_t | x_{t-1}, \{R, r\} \sim \pi_{SB}\left(\nu_t\right), \tag{6}$$

$$\nu_t = R x_{t-1} + r, \tag{7}$$

where $R \in \mathbb{R}^{K-1 \times d_x}$ is a matrix of hyper-planes and $r \in \mathbb{R}^{K-1}$ is a bias vector. $\pi_{SB} : \mathbb{R}^{K-1} \to [0,1]^K$ maps from the reals to the simplex via stick-breaking:

$$\pi_{SB}(\nu) = \left(\pi_{SB}^{(1)}(\nu), \cdots, \pi_{SB}^{(K)}(\nu)\right), \quad \pi_{SB}^{(k)} = \sigma(\nu_k) \prod_{j<k} \sigma\left(-\nu_j\right), \tag{8}$$

for $k = 1, \ldots, K-1$ and $\pi_{SB}^{(K)} = \prod_{k=1}^{K-1} \sigma\left(-\nu_k\right)$ where ν_k is the kth component of of ν; Fig. 1 illustrates the stick-breaking procedure . By including this recurrence in the transition density of z_t, the rSLDS partition the latent space into K pieces, where each piece follows its own linear dynamics. It is through this combination of locally linear dynamical systems that the rSLDS approximates equation 3; the partitioning of the space allows for a more interpretable visualization of the underlying dynamics.

Recurrent SLDS can be learned efficiently and in a fully Bayesian manner, and experiments empirically show that they are adept in modeling the underlying generative process in many cases. However, the stick breaking process used to partition the space poses problems for inference due to its dependence on the permutation of the discrete states $\{1, \cdots, K\}$ (Linderman et al., 2017).

3 Tree-Strucutred Recurrent Switching Linear Dynamical Systems

Building upon the rSLDS, we propose the tree-structured recurrent switching linear dynamical system (TrSLDS). Rather than sequentially partitioning the latent space using stick breaking, we use a tree-structured stick breaking (Adams et al., 2010) procedure to partition the space.

Let $\mathcal{T}$ denote a tree structure with a finite set of nodes $\{\epsilon, 1, \cdots, N\}$. Each node n has a parent node denoted by $\text{par}(n)$ with the exception of the root node, ϵ, which has no parent. For simplicity, we restrict our scope to balanced binary trees where every internal node n is the parent of two children, $\text{left}(n)$ and $\text{right}(n)$. Let $\text{child}(n) = \{\text{left}(n), \text{right}(n)\}$ denote the set of children for internal node n. Let $\mathcal{Z} \subseteq \mathcal{T}$ denote the set of leaf nodes, which have no children. Let $\text{depth}(n)$ denote the depth of a node n in the tree, with $\text{depth}(\epsilon) = 0$.

At time instant t, the discrete latent state z_t is chosen by starting at the root node and traversing down the tree until one of the K leaf nodes are reached. The traversal is done through a sequence of left/right choices by the internal nodes. Unlike in standard regression trees where the choices are deterministic Lakshminarayanan (2016), we model the choices as random variables. We can think of this as a stick breaking process. We start at the root node with a unit-length stick $\pi_\epsilon = 1$, which we divide between its two children. The left child receives a fraction $\pi_{\text{left}(\epsilon)} = \sigma(\nu_\epsilon)$ and the right child receives the remainder $\pi_{\text{right}(\epsilon)} = 1 - \sigma(\nu_\epsilon)$, where $\sigma(\nu) = (1 + e^{-\nu})^{-1}$ is the logistic function. The parameter $\nu_\epsilon \in \mathbb{R}$ specifies the left/right balance. This process is repeated recursively, subdividing π_n into two pieces at each internal node until we reach the leaves of the tree. The stick

assigned to each leaf node is then,

$$\pi_n = \begin{cases} \sigma(\nu_{\text{par}(n)})^{\mathbb{I}[n=\text{left}(\text{par}(n))]} \left(1 - \sigma(\nu_{\text{par}(n)})\right)^{\mathbb{I}[n=\text{right}(\text{par}(n))]} \pi_{\text{par}(n)} & n \neq \epsilon, \\ 1 & n = \epsilon. \end{cases} \tag{9}$$

We incorporate this into the TrSLDS by allowing ν_n to be a function of the continuous latent state,

$$\nu_n(x_{t-1}, R_n, r_n) = R_n x_{t-1} + r_n. \tag{10}$$

The parameters R_n and r_n specify a linear hyperplane in the continuous latent state space. As the continuous latent state x_{t-1} evolves, the left/right choices become more or less probable. This in turn changes the probability distribution $\pi_k(x_{t-1}, \Gamma, \mathcal{T})$ over the K leaf nodes, where $\Gamma = \{R_n, r_n\}_{n \in \mathcal{T}}$. In the TrSLDS, these leaf nodes correspond to the discrete latent states of the model, so that

$$p\left(z_t = k \mid x_{t-1}, \Gamma, \mathcal{T}\right) = \pi_k(x_{t-1}, \Gamma, \mathcal{T}). \tag{11}$$

Fig. 1 illustrates the tree-structured stick breaking procedure.

3.1 A Hierarchical Dynamics Prior that Respects the Tree Structure

Similar to standard rSLDS, the dynamics are conditionally linear given a leaf node z_t. It is intuitive to expect that nearby regions in latent space have similar dynamics. In the context of the tree-structured stick breaking partitions that share a common parent should have similar dynamics. We explicitly model this by enforcing a hierarchical tree-structured prior on the dynamics.

Let $\{A_n, b_n\}$ be the dynamics parameters associated with node n. Even though only the discrete states are associated with the leaf nodes, we will introduce dynamics at the internal nodes as well. These internal dynamics serve as a link between the leaf node dynamics via a hierarchical prior,

$$\text{vec}([A_n, b_n]) | \, \text{vec}([A_{\text{par}(n)}, b_{\text{par}(n)}]) \sim \mathcal{N}(\text{vec}([A_{\text{par}(n)}, b_{\text{par}(n)}]), \Sigma_n), \tag{12}$$

where $\text{vec}(\cdot)$ is the vectorization operator. The prior on the root node is

$$\text{vec}\left([A_\epsilon, b_\epsilon]\right) \sim \mathcal{N}\left(0, \Sigma_\epsilon\right). \tag{13}$$

We impose the following constraint on the covariance matrix of the prior

$$\Sigma_n = \lambda^{\text{depth}(n)} \Sigma_\epsilon, \tag{14}$$

where $\lambda \in (0, 1)$ is a hyper parameter that dictates how "close" a parent and child are to one another. The prior over the parameters can be written as, where the affine term and the $\text{vec}(\cdot)$ operator are dropped for compactness,

$$p(\{A_n\}_{n \in \mathcal{T}}) = p(A_\epsilon) \prod_{i \in \text{child}(\epsilon)} p(A_i | A_\epsilon) \prod_{j \in \text{child}(i)} p(A_j | A_i) \; \ldots \prod_{z \in \mathcal{Z}} p(A_z | A_{\text{par}(z)}). \tag{15}$$

It is through this hierarchical tree structured prior that allows TrSLDS to obtain a multi-scale view of the system. Parents are given the task of learning a higher level description of the dynamics while children are tasked with learning the nuances of the dynamics. The use of hierarchical priors also allows for neighboring sections of latent space to share common underlying dynamics inherited from their parent. TrSLDS can be queried at different levels, where levels deeper in the tree provide more resolution.

TrSLDS shares some features with regression trees (Lakshminarayanan, 2016), even though regression trees are primarily used for standard, static regression problems. The biggest difference is that our tree-structured model has stochastic choices. Moreover, the internal nodes of regression trees have no influence on equation 5; the hierarchical structure is only used for partitioning the latent space.

In the next section we show an alternate view of TrSLDS which we will refer to as the residual model in which internal nodes do contribute to the dynamics. Nevertheless, this “residual model” will turn out to be equivalent to the TrSLDS.

3.2 RESIDUAL MODEL

Let $\{\tilde{A}_n, \tilde{b}_n\}$ be the linear dynamics of node n. and let $\text{path}(n) = (\epsilon, \ldots, n)$ be the sequence of nodes visited to arrive at node n. In contrast to TrSLDS, the dynamics for a leaf node are now determined by all the nodes in the tree

$$p(x_t|x_{t-1}, \tilde{\Theta}, z_t) = \mathcal{N}(x_t|x_{t-1} + \bar{A}_{z_t} x_{t-1} + \bar{b}_{z_t}, \tilde{Q}_{z_t}), \tag{16}$$

$$\bar{A}_{z_t} = \sum_{j \in \text{path}(z_t)} \tilde{A}_j, \quad \bar{b}_{z_t} = \sum_{j \in \text{path}(z_t)} \tilde{b}_j, \tag{17}$$

We model the dynamics to be independent a priori, where once again the $\text{vec}(\cdot)$ operator and the affine term aren't shown for compactness,

$$p(\{\tilde{A}_n\}_{n \in \mathcal{T}}) = \prod_{n \in \mathcal{T}} p(\tilde{A}_n), \quad p(\tilde{A}_n) = \mathcal{N}(0, \tilde{\Sigma}_n), \tag{18}$$

and $\tilde{\Sigma}_n = \tilde{\lambda}^{\text{depth}(n)} \tilde{\Sigma}_\epsilon$ where $\tilde{\lambda} \in (0, 1)$.

The residual model offers a different perspective of TrSLDS. The covariance matrix can be seen as representing how much of the dynamics a node is tasked with learning. The root node is given the broadest prior because it is present in equation 17 for all leaf nodes; thus it is given the task of learning the global dynamics. Nodes deeper in the tree become more associated with certain regions of the space, so they are tasked with learning more localized dynamics which is represented by the prior being more sharply centered on 0. The model ultimately learns a multi-scale view of the dynamics where the root node captures a coarse estimate of the system while lower nodes learn a much finer grained picture.

3.3 EQUIVALENCE OF TRSLDS AND RESIDUAL MODEL

We show the equivalence of TrSLDS and residual model yield the same joint distribution.

Theorem 1. *TrSLDS and the residual model are equivalent if the following conditions are true:* $A_\epsilon = \tilde{A}_\epsilon$, $A_n = \sum_{j \in \text{path}(n)} \tilde{A}_j$, $Q_z = \tilde{Q}_z \; \forall z \in \text{leaves}(\mathcal{T})$, $\Sigma_\epsilon = \tilde{\Sigma}_\epsilon$ *and* $\lambda = \tilde{\lambda}$

Proof. Let $\mathcal{T}$ be a balanced binary tree with K leaf nodes. To show that the models are equal, it suffices to show the equivalence of the likelihood and the prior between models. For compactness, we drop the affine term and the $\text{vec}(\cdot)$ operator. The likelihood of TrSLDS is

$$p(x_{1:T}|z_{1:T}, \Theta) = \prod_{t=1}^{T} \mathcal{N}(x_t|x_{t-1} + A_{z_t} x_{t-1}, Q_{z_t}), \tag{19}$$

and the likelihood of the residual model is

$$p(x_{1:T}|z_{1:T}, \tilde{\Theta}) = \prod_{t=1}^{T} \mathcal{N}\left(x_t|x_{t-1} + \bar{A}_{z_t} x_{t-1}, \tilde{Q}_{z_t}\right). \tag{20}$$

where $\bar{A}_{z_t}$ is defined in equation 17. Substituting $A_{z_t} = \sum_{j \in \text{path}(z_t)} \tilde{A}_j$ into equation 20 equates the likelihoods. All that is left to do is to show the equality of the priors.

We can express $A_n = \sum_{j \in \text{path}(n)} \tilde{A}_j$ recursively

$$A_n = \tilde{A}_n + A_{\text{par}(n)}. \tag{21}$$

Plugging equation 21 into $\ln p(A_n|A_{\text{par}(n)})$

$$\ln p(A_n|A_{\text{par}(n)}) = -\frac{1}{2}\left(A_n - A_{\text{par}(n)}\right)^T \Sigma_n^{-1} \left(A_n - A_{\text{par}(n)}\right) + \text{C} \tag{22}$$

$$= -\frac{1}{2}\left(\tilde{A}_n + A_{\text{par}(n)} - A_{\text{par}(n)}\right)^T \Sigma_n^{-1} \left(\tilde{A}_n + A_{\text{par}(n)} - A_{\text{par}(n)}\right) + \text{C} \tag{23}$$

$$= -\frac{1}{2} \tilde{A}_n^T \Sigma_n^{-1} \tilde{A}_n + \text{C} \tag{24}$$

$$= -\frac{1}{2} \tilde{A}_n^T \left(\lambda^{\text{depth}(n)} \Sigma_\epsilon\right)^{-1} \tilde{A}_n + \text{C} \tag{25}$$

because $\Sigma_\epsilon = \tilde{\Sigma}_\epsilon$ and $\lambda = \tilde{\lambda}$, equation 25 is equivalent to the kernel of $p(\tilde{A}_n)$ implying that the priors are equal. Since this is true $\forall n \in \mathcal{T}$, the joint distributions of the two models are the same. □

4 BAYESIAN INFERENCE

The linear dynamic matrices Θ, the hyper-planes $\Gamma = \{R_n, r_n\}_{n \in \mathcal{T} \setminus \mathcal{Z}}$, the emission parameters Ψ , the continuous latent states $x_{0:T}$ and the discrete latent states $z_{1:T}$ must be inferred from the data. Under the Bayesian framework, this is achieved by computing the posterior

$$p\left(x_{0:T}, z_{0:T}, \Theta, \Psi, \Gamma | y_{1:T}\right) = \frac{p\left(x_{0:T}, z_{1:T}, \Theta, \Psi, \Gamma, y_{1:T}\right)}{p\left(y_{1:T}\right)}, \tag{26}$$

where $Z = p(y_{1:T})$, the marginal likelihood, is

$$p(y_{1:T}) = \sum_{z_{1:T}} \int p\left(x_{0:T}, z_{1:T}, \Theta, \Psi, \Gamma, y_{1:T}\right) \partial x_{0:T} \, \partial\Theta \, \partial\Psi \, \partial\Gamma \, . \tag{27}$$

Obtaining the posterior (26) requires the evaluation of (27) which is usually intractable in practice. We perform fully Bayesian inference via Gibbs sampling (Brooks et al., 2011) the sample from the posterior distribution equation 26. The structure of the model allows for closed form conditional posterior distributions that are easy to sample from. For clarity, the conditional distributions for the TrSLDS are given below:

1. The linear dynamic parameters (A_k, b_k) and state variance Q_k of a leaf node k are conjugate with a Matrix Normal Inverse Wishart (MNIW) prior

$$p((A_k, b_k), Q_k | x_{0:T}, z_{1:T}) \propto p((A_z, b_z), Q_z) \prod_{t=1}^{T} \mathcal{N}(x_t | x_{t-1} + A_{z_t} x_{t-1} + b_{z_t}, Q_{z_t})^{\mathbb{1}[z_t = k]}.$$

2. The linear dynamic parameters of an internal node n are conditionally Gaussian

$$p((A_n, b_n) | \Theta_{-n}) \propto p((A_n, b_n) | (A_{\text{par}(n)}, b_{\text{par}(n)})) \prod_{j \in \text{child}(n)} p((A_j, b_j) | (A_n, b_n)).$$

3. If we assume the observation model is linear and with Gaussian noise then emission parameters $\Psi = \{(C, d), S\}$ are also conjugate with a MNIW prior

$$p((C, d), S | x_{1:T}, y_{1:T}) \propto p((C, d), S) \prod_{t=1}^{T} \mathcal{N}(y_t | C x_t + d, S).$$

4. The choice parameters are logistic regressions which follow from the conditional

$$\begin{aligned} p\left(\Gamma | x_{0:T}, z_{1:T}\right) &\propto p\left(\Gamma\right) \prod_{t=1}^{T} p\left(z_t | x_{t-1}, \Gamma\right) \\ &= p\left(\Gamma\right) \prod_{t=1}^{T} \prod_{j \in \text{path}(z_t) \setminus \epsilon} \text{Bern}\left(j | \sigma\left(R_{\text{par}(j)}^T x_{t-1} + r_{\text{par}(j)}\right)\right). \end{aligned}$$

   Each of these Bernoulli probabilities is amenable to Pólya-gamma augmentation Linderman et al. (2015), Polson et al. (2013). Let ω_t^n be the auxiliary Pólya-gamma random variables introduced at time t for an internal node n. We can express the posterior over the hyper-planer for an internal node n as:

$$p((R_n, r_n) | x_{0:T}, z_{1:T}, \omega_{1:T}^n) \propto p((R_n, r_n)) \prod_{t=1}^{T} \mathcal{N}(\nu_t^n | \kappa_t^n / \omega_t^n, 1/\omega_t^n)^{\mathbb{1}(n \in \text{path}(z_t))} \tag{28}$$

   where $\nu_t^n = R_n^T x_{t-1} + r_n$ and $\kappa_t^n = \mathbb{1}[j = \text{left child}(n)] - \frac{1}{2}\mathbb{1}[j = \text{right child}(n)]$, $j \in \text{child}(n)$. Placing a Gaussian prior makes the posterior conditionally conjugate.

5. Conditioned on the discrete latent states, the continuous latent states are conditionally Gaussian. However, the presence of the tree-structured recurrence potentials $\psi(x_{t-1}, z_t)$ introduced through equation 11 destroys the Gaussinity of the conditional. When the model is augmented with PG random variables ω_t^n, the augmented recurrence potential ,$\psi(x_{t-1}, z_t, \omega_t^n)$, becomes effectively Gaussian, allowing for the use of message passing for efficient sampling. Linderman et al. (2017) show how to perform message-passing using the Pólya-gamma augmented recurrence potentials $\psi(x_t, z_t, w_t)$. In the interest of space, we show the details in the appendix.

6. The discrete latent variables $z_{1:T}$ are conditionally independent given $x_{1:T}$ thus

$$p\left(z_t = k | x_{1:T}, \Theta, \Gamma\right) = \frac{p\left(x_t | x_{t-1}, \theta_k\right) p\left(z_t = k | x_{t-1}, \Gamma\right)}{\sum_{l \in \text{leaves}(\mathcal{T})} p\left(x_t | x_{t-1}, \theta_l\right) p\left(z_t = l | x_{t-1}, \Gamma\right)}, \quad k \in \text{leaves}(\mathcal{T}).$$

7. The posteriors of the Pólya-Gamma random variables are also Pólya-Gamma: $\omega_t^n | z_t, \gamma_n, x_{t-1} \sim PG(1, \nu_t^n)$.

Due to the complexity of the model, good initialization is critical for the Gibbs sampler to converge in a reasonable amount of iterations. Details of the initialization procedure are contained in the appendix.

5 Experiments

We demonstrate the potential of the proposed model by testing them on a number of non-linear dynamical systems. The first, FitzHugh-Nagumo, is a common nonlinear system utilized throughout the neuroscience to describe an action potential. We show that the proposed method can offer different angles of the system. We also compare our model with other approaches and show that we can achieve state of the art performance. We then move on to a Lorenz attractor, a chaotic nonlinear dynamical system and show that the proposed model can once again break down the dynamics and offer an interesting perspective. Finally, we apply the proposed method on the data from [cite graf]

5.1 FitzHugh-Nagumo

The FitzHugh-Nagumo (FHN) model is a 2-dimensional reduction of the Hodgkin-Huxley model which is completely described by the following system of differential equations:

$$\dot{v} = v - \frac{v^3}{3} - w + I_{ext}, \tag{29}$$

$$\tau \dot{w} = v + a - bw. \tag{30}$$

We set the parameters to $a = 0.7$, $b = 0.8$, $\tau = 12.5$ and $I_{ext} \sim \mathcal{N}(0.7, 0.04)$. We trained our model with 100 trajectories where the starting points were sampled uniformly from a $(-3, 3) \times (-3, 3)$ cube. Each of the trajectories were of length of 430, where the last 30 data points of the trajectories were used for testingn. he results are displayed in figure 2.

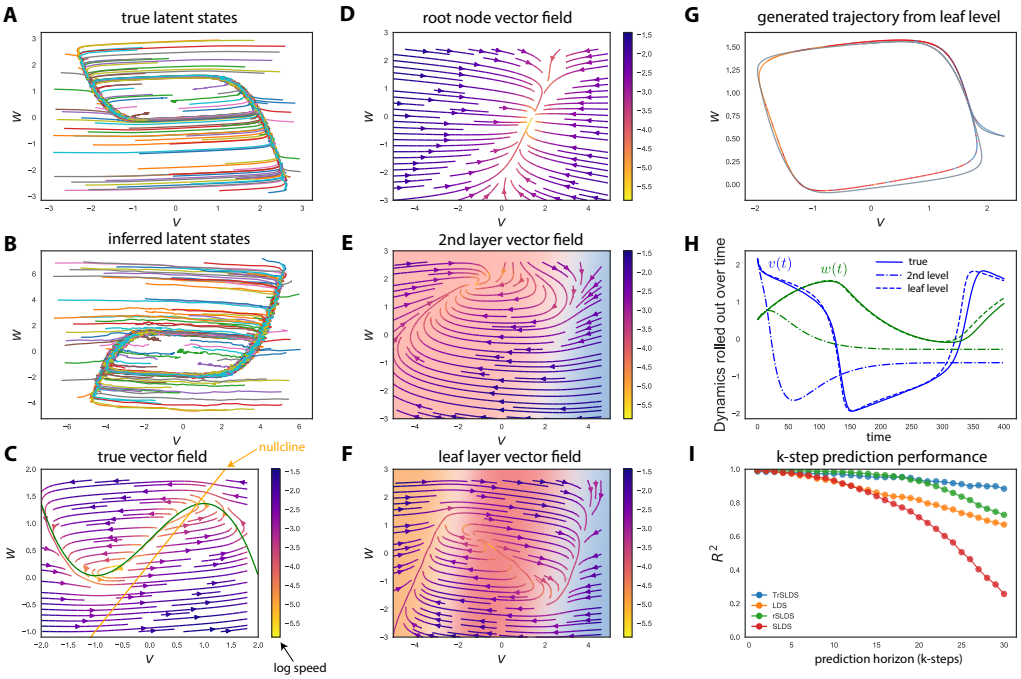

Figure 2: TrSLDS applied to model the FitzHugh-Nagumo nonlinear oscillator. (a) The model was trained with 100 trajectories with different starting points. (b)The model can infer the latent trajectories, up to an affine transformation. (c) The true vector field of FHN is shown where color represents log-speed. The two nullclines are plotted in yellow and green. (d) & (e) & (f) The vector fields display the multi-scale view learned from the model. As we go deeper in the tree, the resolution increases as well which is evident from the vector fields.. (g) A deterministic trajectory from the leaf nodes, projected onto a trajectory FHN for clarity. (h) Plotting w and v over time, we see that the second level captures some of the oscillations but ultimately converges to a fixed point. The model learned by the leaf nodes provides a much better approximation. (i) It is evident from the plot that TrSLDS can accurately perform multi-step prediction of the true trajectory from FHN.

To quantitatively measure the predictive power of TrSLDS, we compute the k-step mean squred error, MSE_k, and it's normalized version, R_k^2, on a test set where MSE_k and R_k^2 are defined as

$$\text{MSE}_k = \frac{1}{T-k}\sum_{t=0}^{T-k} \|y_{t+k} - \hat{y}_{t+k}\|_2^2\,, \qquad R_k^2 = 1 - \frac{(T-k)\text{MSE}_k}{\sum_{t=0}^{T-k}\|y_{t+k} - \bar{y}\|_2^2} \tag{31}$$

where $\bar{y}$ is the average of a trial and $\hat{y}_{t+k}$ is the prediction at time $t+k$ which is obtained by (i) using the the samples produced by the sampler to obtain an estimate of $\hat{x}_T$ (ii) propagate $\hat{x}_T$ for k time steps forward to obtain $\hat{x}_{t+k}$ and then (iii) obtain $\hat{y}_{t+k}$ according to equation 3. We compare the model to LDS. SLDS and rSLDS for $k = 1, \ldots 30$ over the last 30 time steps for all 100 trajectories. Figure 2i displays the comparisons.

5.2 Lorenz Attractor

Lorenz attractors are chaotic systems whose nonlinear dynamics are defined by the following differential equations

$$\begin{aligned} \dot{x_1} &= \sigma\left(x_2 - x_1\right), \\ \dot{x_2} &= x_1(\rho - x_3) - x_2, \\ \dot{x_3} &= x_1 x_2 - \beta x_3. \end{aligned}$$

The parameters of the Lorenz were set to $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$. The data consisted of 50 trajectories, each of length of 230 where the first 200 points are used for training and the last 30 are

used for testing. We fit the data using 4 states which corresponds to a tree of depth 3. The results are shown in Figure 3.

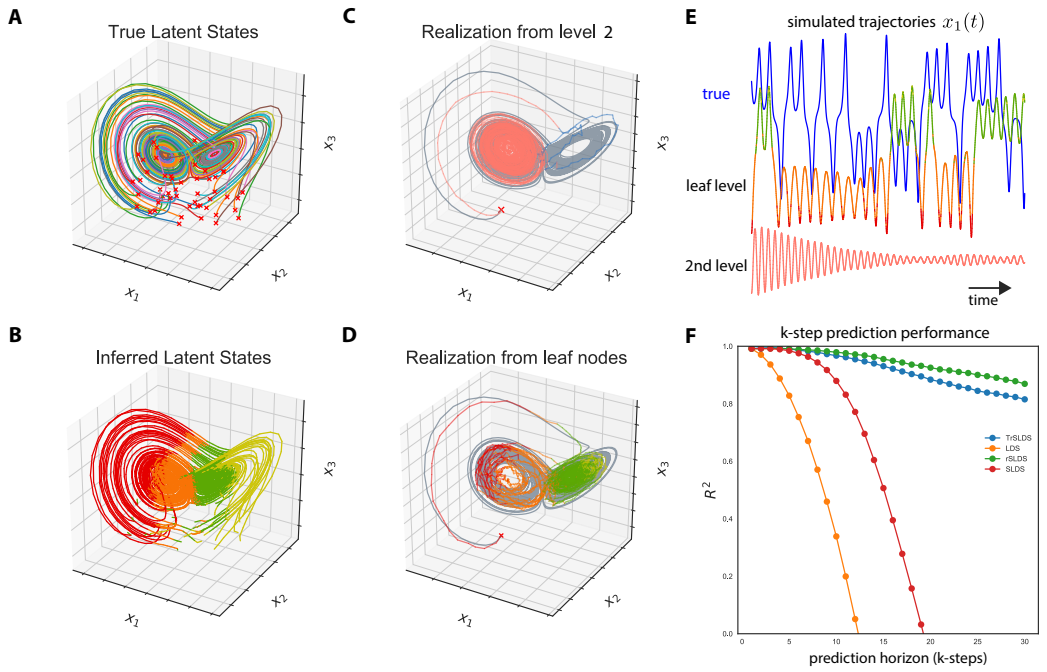

Figure 3: (a) The 50 trajectories used to train the model are plotted where the red "x" displays the starting point of the trajectory.(b) The inferred latent states are shown, colored by their discrete latent state. (c) We see that the second layer approximates the Lorenz with 2 ellipsoids. A Lorenz starting at the same initial point is shown for comparison. (d) Going one level lower in the tree, we see that in order to capture the nuances of the dynamics, each of the ellipsoids must be split in half. A trajectory from the Lorenz is shown for comparison. (e) Plotting the dynamics, it is evident that the leaf nodes improve on it's parent's approximation. (f) The R_k^2 shows the predictive power of TrSLDS.

The butterfly shape of the Lorenz lends itself to being roughly approximated by two ellipsoids as a rough estimate of the dynamics; this is exactly what TrSLDS learns in the second level of the tree. As is evident from Figure 2b, the two ellipsoids don't capture the nuances of the dynamics. Thus, the model partitions each of the ellipsoid to obtain a finer description. We can see that embedding the system with a hierarchical tree-structured prior allows for the children to build off it's parents approximations.

5.3 Neural Data

To validate the model and inference procedure, we used the neural spike train data recorded from the primary visual cortex of an anesthetized macaque monkey collected by Graf et al. (2011). The dataset is composed of short trials where the monkey viewed periodic temporal pattern of motions of 72 orientations, each repeated 50 times. Previous state space modeling of the dataset showed that for each orientation of the drifting grating stimulus, the neural response oscillates over time, but in a stimulus dependent geometry (Zhao & Park, 2017b). We used 25 trials each from a subset of 4 stimulus orientations grouped in two (140 and 150 degrees vs 230 and 240 degrees). Each trial contained 140 neurons, and their spike trains were binarized with a 10 ms window. We truncated the onset and offset neural responses, resulting in 111 time bins per trial.

The TrSLDS with 3-dimensional latent state and 4 leaf nodes converged to Fig. 4. We observe that the population-wide modulation is captured by temporal oscillations in the form of rings in the state space. Furthermore, the discrete states segment the oscillations into different phases. This is similar to the nonlinear limit cycle case of FHN, but in this case, the nonlinear dynamics is probably due to

the biased subsampling of the electrode array that preferentially responds to certain phases of the oscillation.

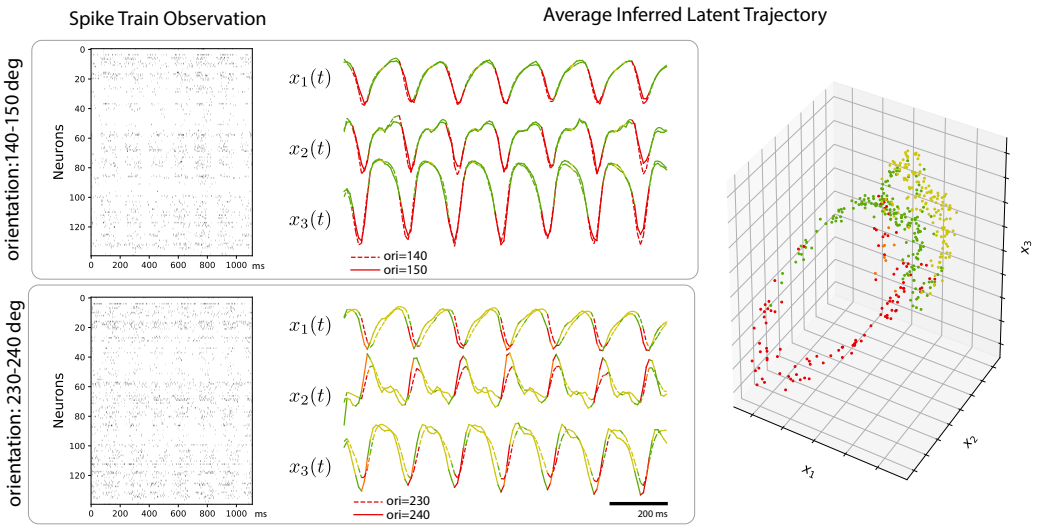

Figure 4: Inference results from neural data. (left) Spike raster plot showing two single trials in response to drifting gratings of two different orientations. (middle) Inferred 3-dim latent trajectory averaged over the 25 repeated trials, displayed over time. Color corresponds to discrete states. Yellow and green discrete states share their parent node. (right) Same as the middle, except viewed in the state space. Note the two ring structures corresponding to the two orientation groups.

6 Conclusion and Future Work

In this paper, we propose tree-structured recurrent switching linear dynamical systems (TrSLDS) which is an extension of Linderman et al. (2017) rSLDS. The tree-structured stick breaking removes the dependence on the permutation of the discrete latent states. The tree-structured stick breaking paradigm naturally lends itself to imposing a tree-structured hierarchical prior on the dynamics. The structure of the prior allows for a multi-scale view of the system; one can query at different levels of the trees to see different scales of the resolution. We also developed a fully Bayesian approach to learning the parameters of the model. The analysis of the Graf data suggests that the method can also be used to analyze neural data.

References

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

Ryan P Adams, Zoubin Ghahramani, and Michael I Jordan. Tree-Structured Stick Breaking for Hierarchical Data. In J D Lafferty, C K I Williams, J Shawe-Taylor, R S Zemel, and A Culotta (eds.), *Advances in Neural Information Processing Systems 23*, pp. 19–27. Curran Associates, Inc., 2010.

David Barber. Expectation Correction for Smoothed Inference in Switching Linear Dynamical Systems. Technical report, 2006.

David Barber, A Taylan Cemgil, and Silvia Chiappa. *Bayesian time series models*. Cambridge University Press, 2011.

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

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.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pp. 2980–2988, 2015.

Petar Djuric and Mónica Bugallo. Cost-Reference Particle Filtering for Dynamic Systems with Nonlinear and Conditionally Linear States, 9 2006.

Arnaud Doucet, Nando Freitas, and Neil Gordon. An Introduction to Sequential Monte Carlo Methods. In *Sequential Monte Carlo Methods in Practice*, pp. 3–14. Springer New York, New York, NY, 2001. doi: 10.1007/978-1-4757-3437-9{_}1.

Elena A. Erosheva and S. McKay Curtis. Dealing with Reflection Invariance in Bayesian Factor Analysis. *Psychometrika*, 82(2):295–307, 6 2017. ISSN 0033-3123. doi: 10.1007/s11336-017-9564-y.

Emily Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. Nonparametric Bayesian Learning of Switching Linear Dynamical Systems. In D Koller, D Schuurmans, Y Bengio, and L Bottou (eds.), *Advances in Neural Information Processing Systems 21*, pp. 457–464. Curran Associates, Inc., 2009.

Roger Frigola, Yutian Chen, and Carl Edward Rasmussen. Variational Gaussian Process State-Space Models. In Z Ghahramani, M Welling, C Cortes, N D Lawrence, and K Q Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 3680–3688. Curran Associates, Inc., 2014.

Yuanjun Gao, Evan W Archer, Liam Paninski, and John P Cunningham. Linear dynamical neural population models through nonlinear embeddings. In *Advances in neural information processing systems*, pp. 163–171, 2016.

John Geweke and Guofu Zhou. Measuring the Pricing Error of the Arbitrage Pricing Theory. *Review of Financial Studies*, 9(2):557–587, 4 1996. ISSN 0893-9454. doi: 10.1093/rfs/9.2.557.

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

Arnulf B. Graf, Adam Kohn, Mehrdad Jazayeri, and J. Anthony Movshon. Decoding the activity of neuronal populations in macaque primary visual cortex. *Nature neuroscience*, 14(2):239–245, February 2011. ISSN 1546-1726. doi: 10.1038/nn.2733.

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

Simon S Haykin. *Kalman Filtering and Neural Networks*. John Wiley & Sons, Inc., New York, NY, USA, 2001. ISBN 0471369985.

Matthew Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in neural information processing systems*, pp. 2946–2954, 2016.

Rahul G Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. 2017.

Balaji Lakshminarayanan. Decision Trees and Forests: A Probabilistic Perspective. Technical report, UCL (University College London), 2016.

Scott Linderman, Matthew Johnson, and Ryan P Adams. Dependent Multinomial Models Made Easy: Stick-Breaking with the Polya-gamma Augmentation. In C Cortes, N D Lawrence, D D Lee, M Sugiyama, and R Garnett (eds.), *Advances in Neural Information Processing Systems 28*, pp. 3456–3464. Curran Associates, Inc., 2015.

Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems. In Aarti Singh and Jerry Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 914–922, Fort Lauderdale, FL, USA, 9 2017. PMLR.

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

Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, pp. 1, 2018.

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. doi: 10.1080/01621459.2013.829001.

Simo Särkkä. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.

David Sussillo, Rafal Józefowicz, L. F Abbott, and Chethan Pandarinath. LFADS - Latent Factor Analysis via Dynamical Systems. *CoRR*, abs/1608.06315, 2016.

Yuan Zhao and Il Memming Park. Interpretable nonlinear dynamic modeling of neural trajectories. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Yuan Zhao and Il Memming Park. Variational recursive dual filtering. (under review), July 2017a.

Yuan Zhao and Il Memming Park. Variational Latent Gaussian Process for Recovering Single-Trial Dynamics from Population Spike Trains. *Neural Computation*, 29(5), May 2017b. doi: 10.1162/NECO_a_00953.

A INITIALIZATION

We initialized the Gibbs sampler using the following initialization procedure: (i) probabilistic PCA was performed on the data, $y_{1:T}$ to initialize the emission parameters, $\{C, d\}$ and the continuous latent states, $x_{1:T}$. (ii) To initialize the dynamics of the nodes ,Θ, and the hyperplanes, Γ, we propose greedily fitting the proposed model using MSE as the loss function. We first optimize over the root node

$$\underset{A_\epsilon, b_\epsilon}{\arg\min} \frac{1}{T} \sum_{t=0}^{T} \|x_{t+1} - x_t - A_\epsilon x_t - b_\epsilon\|_2^2 \,, \tag{32}$$

and obtain $A_\epsilon^*, b_\epsilon^*$ (Note that $A_\epsilon^*, b_\epsilon^*$ can obtained in closed form by computing their corresponding OLS estimates). Fixing A_ϵ^* and b_ϵ^*, we then optimize over the second level in the tree

$$\underset{A_1, b_1, A_2, b_2, R_\epsilon, r_\epsilon}{\arg\min} \frac{1}{T} \sum_{t=0}^{T} \|x_{t+1} - \sigma(v_\epsilon)\hat{x}_1 - \sigma(-v_\epsilon)\hat{x}_2\| \,, \tag{33}$$

$$\hat{x}_i = x_t + (A_\epsilon^* + A_i)\, x_t + (b_\epsilon^* + b_i) \,, \tag{34}$$

$$v_\epsilon = R_\epsilon^T x_t + r_\epsilon. \tag{35}$$

This procedure would continue until we reach the leaf nodes of the tree. Θ^* and Γ^* are then used to initialize the dynamics and the hyperplanes, respectively. In our simulations, we used stochastic gradient descent with momentum to perform the optimization. (iii) The discrete latent states, $z_{1:T}$, were initialized by performing hard classification using Γ^* and the initial estimate of $x_{0:T}$.

B DEALING WITH ROTATIONAL INVARIANCE

A well known problem with these types of model is it's susceptibility to rotational and scaling transformation, thus we can only learn the dynamics up to an affine transformation Erosheva &

Curtis (2017). During Gibbs sampling the parameters will continuously rotate and scale, which slows down the mixing of the chains. One possible solution to the issue is if we constrained C to have some special structure which would make the model identifiable; this would require sampling from manifolds which is usually inefficient. Similar to Geweke & Zhou (1996), we use the following procedure to prevent the samples from continuously rotating and scaling:

- Once we obtain a sample from the conditional posterior of the emission parameters $\{C, D\}$, we normalize the columns of C.
- RQ decomposition is performed on C to obtain U, O where $U \in \mathcal{R}^{d_y \times d_x}$ is an upper triangular matrix and $O \in \mathcal{R}^{d_x \times d_x}$ is an orthogonal matrix.
- We set $C = U$ and rotate all the parameters of the model using O.