2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

Switching Dynamical Systems with Deep Neural Networks

César Ojeda*, Bogdan Georgiev‡, Kostadin Cvejoski‡§,
Jannis Schücker‡, Christian Bauckhage‡ and Ramsés J. Sánchez†§
*Berlin Center for Machine Learning and TU Berlin, 10587 Berlin, Germany
†B-IT, University of Bonn, Bonn, Germany
‡Fraunhofer Center for Machine Learning and Fraunhofer IAIS, 53757 Sankt Augustin, Germany
§Competence Center Machine Learning Rhine-Ruhr
ojedamarin@tu-berlin.de, sanchez@bit.uni-bonn.de
{kostadin.cvejoski, bogdan.georgiev, jannis.schueker christian.bauckhage}@iais.fraunhofer.de

***Abstract*—The problem of uncovering different dynamical regimes is of pivotal importance in time series analysis. Switching dynamical systems provide a solution for modeling physical phenomena whose time series data exhibit different dynamical modes. In this work we propose a novel variational RNN model for switching dynamics allowing for both non-Markovian and non-linear dynamical behavior between and within dynamic modes. Attention mechanisms are provided to inform the switching distribution. We evaluate our model on synthetic and empirical datasets of diverse nature and successfully uncover different dynamical regimes and predict the switching dynamics.**

I. Introduction

Many complex dynamical signals naturally feature an inherent compositional form, in the sense that their data generating process can be decomposed into different dynamical modes. For example, an NBA basketball player rapidly changes between moves as the game evolves, thereby composing complex two-dimensional trajectories; a handwriting signal is made up of successive strokes which differ in length, velocity, curvature and hence in their dynamics; multi-stable dynamical systems exhibit trajectories that can naturally be labeled according to their behavior around different attractors. Such an inherent compositionality suggests that the holistic dynamics of these systems decomposes into a switching process determining the sequence of modes, the structure of the modes themselves, and the interplay between those two.

Switching dynamical systems (SDS) provide a natural way to perform unsupervised learning in time series. Akin to clustering methods which index data of similar structure in feature space, the switching mechanism in SDS indexes time series according to similar evolution patterns. Beyond indexing different dynamical regimes, however, the switching dynamics itself is of major interest if one wants to provide a complete dynamical picture of complex time series. In recent years recurrent SDS have provided tools to incorporate dynamical information into the switching process (see Section II). These methodologies are, however, hindered by the inherent limitations of linear dynamical systems — linear Gaussian transition functions ignore the pervasive non-linear and non-Markovian behavior common to real dynamical phenomena.

In this work we overcome these limitations by exploiting the advantages of recurrent neural networks (RNNs) within a framework for SDS. This approach allows for an explicit description of non-linear and non-Markovian transition functions for the dynamics of both modes and switching. Indeed, within our model the modes are learned through independent RNNs whereas, similar to (mixture of) expert systems, the selection of modes is handled via a categorical distribution. Furthermore, the class probabilities of the categorical variables change dynamically. On one hand, the class probabilities are conditioned on the hidden states of the (also independent) switching history, which we modeled via yet another RNN. On the other hand, the class probabilities are also informed on both hidden states and predictions of the dynamic modes through an attention mechanism. Finally, we enforce a finite entropy among the modes to balance them throughout the entire signal. We learn the model via the introduction of an approximate posterior distribution to perform inference over the dynamic categorical variables. The parameter optimization is then performed through maximum likelihood estimation of the corresponding bound.

II. Related Work

The study of different time regimes in time series has a venerable history in the Bayesian network community [1], [2], [3]. The modeling can be understood as an extension of the hidden Markov chain formalism where each latent state has an associated dynamical mode. Two main methodologies are applied: on one hand, change point models infer change point locations which differentiate the dynamical regimes [4]. Within each new regime, parameters are learned for new dynamical models. On the other hand, one can incorporate the knowledge of past regimes and then switch among the dynamical modes [5], [6]. Furthermore, non-parametric approaches allow these models to sample from an unknown number of dynamical modes [7]. Such methods have also been exploited in the context of stochastic processes [8] wherein dynamical modes correspond to different parameter values for drift and diffusion operators in stochastic differential equations. In contrast to these works, the use of RNNs allows our model to describe

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

2020 25th International Conference on Pattern Recognition (ICPR) | 978-1-7281-8808-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPR48806.2021.9412566

non-linear dynamics with non-Markovian characteristics while still being able to capture the stochastic nature of the change point dynamics.

Motivated by the Bayesian dynamical network formalism we allow for stochastic dynamics along the switching. Variational auto-encoders (VAE) [9] were first used in [10], [11] to introduce variability into the transition functions of RNNs. These models are trained by maximizing a variational lower bound defined with respect to a set of approximating distributions. Estimating these lower bounds requires the calculation of averages over the approximating distributions, and the high variance of the derivatives of the such bounds is resolved via the reparametrization trick. Different from these approaches, our work introduces the stochastic nature of the transition *through the dynamics of the categorical variables* which characterize the switch. We do not provide a latent code for the transition operator but instead, a categorical variable to index the dynamic modes. More recently, variational auto-encoders and switching linear dynamical systems were used for Bayesian filtering in [12]. In contrast to this work, our switching mechanism takes place in data space directly and allows for non-Markovian transitions within each dynamic mode.

III. Background

In this work we combine ideas from Switching Linear Dynamical Systems models (SLDS) with those from sequence modeling with RNNs in order to incorporate richer dynamical representations. Inference is accomplished within the Bayesian formalism via variational approximate inference. In this section we briefly review the different models components, as well as the inference framework, thereby laying the foundations of our model.

A. Switching Linear Dynamical System models

Switching linear dynamical system models aim to capture complex (non-linear) time series behaviour via a collection of so-called dynamical modes, each of which is approximated by a linear model — i.e. the complex signal is broken into a collection of simpler (linear) mappings. These models inherit the methodology of hidden Markov models and linear dynamical systems. We refer to [5], [13] for further background.

In short, one assumes that at each time step t there is a corresponding categorical latent state z_t taking one of K different values and following the Markovian transitions

$$z_{t+1} \,|\, z_t \sim \pi_{z_t}, \tag{1}$$

where $\pi_{z_t} \in [0,1]^K$ gives the usual Markov transition probabilities. The classical approach [5], [13] also introduces the continuous latent states $\mathbf{h}_t \in \mathbb{R}^p$ — these follow affine dynamics, with the different modes being indexed by z_t,

$$\mathbf{h}_{t+1} = \mathbf{A}_{z_{t+1}}\mathbf{h}_t + \mathbf{b}_{z_{t+1}} + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(0, \mathbf{Q}_{z_{t+1}}), \tag{2}$$

where $\mathbf{A}_k, \mathbf{Q}_k$ are matrices of the form $\mathbb{R}^{p\times p}$ whereas $\mathbf{b}_k \in \mathbb{R}^p$ and $k \in (1, ..., K)$. At last, the observed data points $\mathbf{x}_t \in \mathbb{R}^d$ are obtained via

$$\mathbf{x}_t = \mathbf{C}_{z_t}\mathbf{h}_t + \mathbf{d}_{z_t} + \mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(0, \mathbf{S}_{z_{t+1}}), \tag{3}$$

with $\mathbf{C}_k \in \mathbb{R}^{d\times p}$, $\mathbf{S}_k \in \mathbb{R}^{d\times d}$ and the drift terms $\mathbf{d}_k \in \mathbb{R}^d$.

An important remark is that each categorical state z_{t+1} depends only on the previous one z_t. This seems to limit the influence of the continuous latent variable $\mathbf{h}_t$ on the discrete switch — for instance, suppose that a certain critical subset $U \subset \mathbb{R}^p$ is such that, if $\mathbf{h}_t \in U$, then a discrete switch of the system (i.e. a particular change of z_t) is to take place. One may imagine that in such a scenario the switching would be difficult to capture unless the z_t are informed about the $\mathbf{h}_t$. To overcome this disadvantage Linderman et al. proposed an augmented model (recurrent SLDS or **rSLDS**) [5] which makes use of the following generation scheme for z_t

$$z_{t+1} \,|\, z_t, \mathbf{h}_t, \{\mathbf{R}_k, \mathbf{r}_k\} \sim \pi_{SB}(\nu_{t+1}), \tag{4}$$

with $\nu_{t+1} = \mathbf{R}_{z_t}\mathbf{h}_t + \mathbf{r}_{z_t}$. Here π_{SB} is a certain stick-breaking distribution, $\mathbf{R}_k \in \mathbb{R}^{K-1\times p}$ captures the recurrent dependencies between z_t and $\mathbf{h}_t$, and $\mathbf{r}_k \in \mathbb{R}^{K-1}$ models the Markovian transitions between consecutive states z_{t+1} and z_t.

B. Modeling Time Series with Recurrent Neural Networks

Given a sequence $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$, RNNs process each $\mathbf{x}_t$ through the update of a hidden state $\mathbf{h}_t$ at each time step $t \in (1, \ldots, T)$. The update is implemented via a deterministic non-linear transition function f_θ thus

$$\mathbf{h}_t = f_\theta\left(\mathbf{h}_{t-1}, \mathbf{x}_t\right), \tag{5}$$

where $\mathbf{h}_t \in \mathbb{R}^p$, $\mathbf{x}_t \in \mathbb{R}^d$ and θ is the parameter set of f. Given the set of hidden states $\mathbf{h}_t$ one can model the observed sequence by approximating its joint probability distribution function as

$$p(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T) = \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{<t}), \quad p(\mathbf{x}_t|\mathbf{x}_{<t}) = g_\varphi(\mathbf{h}_{t-1}), \tag{6}$$

where g with parameter set φ maps $\mathbf{h}_t$ to a probability distribution over outputs, and where $\mathbf{x}_{<t}$ denotes the dependence on the history.

C. Variational Inference

For inference, we follow the Bayesian inference scheme. Let $\mathbf{x}$ be the observed data and $\mathbf{z}$ a set of latent variables. In this formalism, we specify the model through the forms of the likelihood $p(\mathbf{x}|\mathbf{z})$ and a prior $p(\mathbf{z})$. The goal of inference is to obtain the posterior distribution over $\mathbf{z}$ following Bayes rule. One is usually confronted with intractable forms of the model's evidence $p(\mathbf{x})$, and in consequence the posterior distribution $p(\mathbf{z}|\mathbf{x})$ is not directly accessible. One must then resort to approximating methods for posterior distribution estimation [14]. Within the variational inference approach one is required to define an approximate posterior distribution $q(\mathbf{z})$ which is tractable. This distribution is chosen to approximate the unknown true posterior distribution — by minimizing e.g. the Kullback-Leibler divergence KL $[q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})]$. Since the posterior is not explicitly available, variational approximate

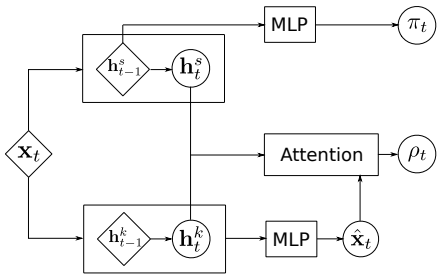

Fig. 1: Architecture of the NVSDS model. For a given sequence $\mathbf{x}_t$ the data is fed into the recurrent network modeling the modes dynamics (lower part) as well as the switching dynamics (upper part). The representations $\mathbf{h}$ obtained by the experts' dynamics are fed into an MLPs parametrizing a Gaussian distribution for the outputs $\hat{x}_t$. The experts' representations, their prediction as well as the hidden state of the switching are fed into the attention mechanism which finally parametrizes the categorical distribution.

inference achieve the minimization of this divergence via the maximization of a lower bound to the model's evidence

$$\mathcal{L}[q] = \mathbb{E}_{q(\mathbf{z})}\left[\log p(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}\left[q(\mathbf{z})||p(\mathbf{z})\right], \tag{7}$$

where the first term is the averaged log likelihood of the model over the approximate posterior distribution and drives the learning of the data, whereas the second term plays the role of a regularizer.

IV. Neural Variational Switching Dynamical Systems (NVSDS)

In this section we introduce the Neural Variational Switching Dynamical System model. Our main goal is to include the notions of non-Markovianity and non-linearity in the dynamical modes. In general terms, we proceed by substituting the transitions function of rSLDS with RNNs as well as MLP-parametrizations of categorical distributions for the switching distributions. We tackle inference via a variational approach, defining an approximate posterior over the categorical index variables. However, the RNN and MLP parameters are obtained as maximum likelihood with a direct optimization of the bound. An overview of our model is given in Fig. 1.

Suppose we have a complex signal $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ composed of K dynamical modes. Following the traditional mixture model approach we implement the dynamic switching between modes in terms of discrete latent variables $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_T)$. Specifically, at each time step t we have a K-dimensional latent variable $\mathbf{z}_t$ having a $1-$of$-K$ representation — i.e. z_t^k is equal to 1 if $\mathbf{x}_t$ was generated from the kth dynamical mode and it is 0 otherwise.

Generative model — The prior distribution over $\mathbf{z}_t$ is specified in terms of a set of class probabilities $\pi_t^k = p(z_t^k = 1)$, which are required (by the switching dynamics) to have an intrinsic dynamic behaviour. Akin to Eq. (1) we introduce a RNN whose sole role is to govern the switching dynamics

$$\mathbf{h}_t^s = f_{\theta_s}\left(\mathbf{x}_t, \mathbf{h}_{t-1}^s\right), \tag{8}$$

with $\mathbf{h}_t^s \in \mathbb{R}^s$ encoding the independent switching dynamics and θ_s the parameter set of the transition function, which we implement using a long-short-term memory (LSTM) network [15]. We then define the set of dynamic class probabilities as

$$\pi_t^k = \mathrm{softmax}\left[g_{\theta_k}\left(\mathbf{h}_{t-1}^s\right)\right], \tag{9}$$

where the softmax function is computed over the K modes and g_{θ_k} denotes a multilayer perceptron (MLP) with parameter θ_k. We can now write the prior distribution over $\mathbf{z}_t$ as

$$p(\mathbf{z}_t) = \prod_{k=1}^{K}\left(\pi_t^k\right)^{z_t^k}. \tag{10}$$

On the other hand, we model each dynamic mode $\mathbf{x}_t^k$ as a normal distribution parametrized by MLPs. The latter are conditioned on the hidden state of a RNN which approximates the mode's transition function, as discussed in Section III-B. Specifically, we write

$$p(\mathbf{x}_t^k \,|\, \mathbf{x}_{<t}^k) = \mathcal{N}(\boldsymbol{\mu}_t^k, \mathrm{diag}\left[(\boldsymbol{\sigma}_t^k)^2\right]), \tag{11}$$

where both $\boldsymbol{\mu}_t^k, \boldsymbol{\sigma}_t^k \in \mathbb{R}^d$ are given by $\left[\boldsymbol{\mu}_t^k, \boldsymbol{\sigma}_t^k\right] = g_{\varphi_k}\left(\mathbf{h}_{t-1}^k\right)$, and g_{φ_k} is again an MLP. Let us note here that Eq. (11) corresponds to our neural network generalization of Eq. (3). The mode's hidden state $\mathbf{h}_t^k \in \mathbb{R}^p$ follows from the state transition function

$$\mathbf{h}_t^k = f_{\varphi_k}\left(\mathbf{x}_t, \mathbf{h}_{t-1}^k\right), \tag{12}$$

with f implemented by a LSTM and φ_k the parameter set for the kth mode. Eq. (12) should be compared to Eq. (2). Different from rSLDS the RNN provides a deterministic transition function. The stochastic aspect of the dynamical modes is included through the output distributions.

The full generative model has the following joint probability distribution

$$p(\boldsymbol{z}_{\leq \boldsymbol{T}}, \boldsymbol{x}_{\leq \boldsymbol{T}}) = \prod_{k=1}^{K}\prod_{t=1}^{T}\left(\pi_t^k\, p\left(\mathbf{x}_t^k \middle| \mathbf{x}_{<t}^k\right)\right)^{z_t^k}. \tag{13}$$

Inference — Although one can readily integrate out $\mathbf{z}_t$ from Eq. (13) and find an optimal posterior distribution, with respect to Eq. (7), via expectation-maximization (we do exactly this at the end of this section), the solution does not explicitly uses the hidden representations $\mathbf{h}_t^k$ and yields a distribution which is only optimal locally in time. Following our discussion in Section III-C, we introduce instead the approximate posterior distribution

$$q(\mathbf{z}_t|\mathbf{x}_{\leq t}) = \prod_{k=1}^{M}\left(\rho_t^k\right)^{z_t^k}, \tag{14}$$

where the dynamic posterior class probabilities ρ_t^k are defined through an *attention mechanism*, which we now define.

Attention mechanism — We take advantage of the freedom we have in defining ρ_t^k to enrich the switching process by

incorporating contextual comparison of the modes' encoding. We do this through an attention mechanism — that is, we dynamically adapt the modes' selection given their current representation. Consider a per-mode hidden state representation

$$\mathbf{u}_k = \sigma\left(\mathbf{W}_k\,\mathbf{H}_t + \mathbf{V}_k\,\mathbf{h}_t^s + \mathbf{b}_k\right), \tag{15}$$

where $\mathbf{H}_t = \left([\hat{\mathbf{x}}_t^1, \mathbf{h}_t^1], [\hat{\mathbf{x}}_t^2, \mathbf{h}_t^2], \ldots, [\hat{\mathbf{x}}_t^K, \mathbf{h}_t^K]\right) \in \mathbb{R}^{K*(d+p)}$ contains both the prediction $\hat{\mathbf{x}}_t^k$ and hidden state $\mathbf{h}_t^k$ of each mode; $\mathbf{h}_t^s$ is the hidden state encoding the switching dynamics, Eq. (8); and $\mathbf{W}_k \in \mathbb{R}^{a\times K*(d+p)}$, $\mathbf{V}_k \in \mathbb{R}^{a\times s}$ and $\mathbf{b}_k \in \mathbb{R}^a$ are trainable variables. The function $\sigma(\cdot)$ denotes the hyperbolic tangent.

Now consider a per-mode context vector, or per-mode fixed query, $\mathbf{c}_k \in \mathbb{R}^a$ which is also to be trained. We define the dynamic posterior class probabilities as

$$\rho_k^t = \text{softmax}\left[\mathbf{u}_k \cdot \mathbf{c}_k\right], \tag{16}$$

where the softmax function is taken over the modes. Different from common forms of additive attention [16] [17], within our approach each mode performs its own *translation* of the context, similar to what was proposed in [18]. As an illustrative example, one can think of a crowd of experts of different nationalities trying to understand a text in a foreign language. The common text (here $\mathbf{u}_k$) is the context known to all experts, each expert must in turn translate the context to its own native language prior (here $\mathbf{c}_k$) to decide whether the text in question corresponds to her area of expertise.

Learning — Inserting Eq. (13) and Eq. (14) into Eq. (7) we write the variational lower bound

$$\mathcal{L}[q] = \mathbb{E}_{p_D(\mathbf{x})}\left[\sum_{k=1}^{K}\sum_{t=1}^{T}\left\{\rho_t^k\,\log p(\mathbf{x}_t^k|\mathbf{x}_{<t}^k) + \rho_t^k\,\log\left[\frac{\pi_t^k}{\rho_t^k}\right]\right\}\right], \tag{17}$$

where we have performed the averages over the categorical distributions, using the fact that $\mathbb{E}_{q(\mathbf{z})} z_t^k = \rho_t^k$, and where $p_D(\mathbf{x})$ denotes the empirical data distribution.

Mode regularization — One common drawback of expert-like systems occurs during training. Due to inhomogenities in the initial conditions of the parameter space, one mode (expert) may happen to have a subtle advantage over the others in predicting the dynamics. Such initial conditions will compound over time as the categorical switch prioritizes this mode thus increasing its advantage over the others. In other words, the mode will be preferred not as a consequence of its knowledge of the dynamical regime, but as a consequence of training imbalances. One must therefore enforce dynamical diversity by imposing cost functions to be trained along side the maximization of the lower bound Eq. (17). This issue has been encountered and addressed in various (static) settings [19], [20], [18]. We introduce the mode entropy

$$\mathcal{H}[\rho] = -\mathbb{E}_{p_D(\mathbf{x})}\sum_{k=1}^{K}\tilde{\rho}_k\,\log\tilde{\rho}_k, \tag{18}$$

where $\tilde{\rho}_k$ is the time-average posterior class probabilities. Below we demonstrate empirically our intuition that maximizing this entropy helps avoiding the "elimination" of the expert modes during training. The regularized model then seeks to maximize $\mathcal{L}'[q] = \mathcal{L}[q] + \lambda_e\,\mathcal{H}[\rho]$, where λ_e is a hyperparameter.

An expectation-maximization solution to NVSDS (NVSDS-EM) — Instead of introducing the approximate posterior Eq. (14) one can directly optimize Eq. (7) by noticing $\mathcal{L}[q]$ is nothing but the negative Kullback-Leibler divergence between $q(\mathbf{z})$ and $p(\mathbf{x},\mathbf{z})$. An optimal lower bound is then found by minimizing this divergence. Such a minimum happens only for $\log q(\mathbf{z}) = \log p(\mathbf{x},\mathbf{z}) + \text{const}$. We can then simply write the approximate posterior as

$$q(\mathbf{z}) = \prod_{t=1}^{T}\prod_{k=1}^{K}\left(\frac{\rho_t^k}{\sum_k^K \rho_t^k}\right)^{z_t^k}, \quad \rho_t^k = \pi_t^k\,p(\mathbf{x}_t^k|\mathbf{x}_{<t}^k), \tag{19}$$

where π_t^k and $p(\mathbf{x}_t^k|\mathbf{x}_{<t}^k)$ are defined in Eq. (9) and Eq. (11), respectively. Note that in contrast to the NVSDS model, the computation of ρ_t^k does not exploit the $\mathbf{h}_t^k$ encoding of the modes through an attention mechanism.

V. Experiments

In this section we provide the experimental framework upon which we tested our model. We begin by briefly defining three baseline models against which we compared our results. We then specify both the architecture of the different neural networks composing the models, as well as the corresponding hyperparameters.

We evaluate our models in four datasets: (i) as a show-case or proof of concept we consider a dynamical system with a chaotic attractor — the Lorenz system (Lorenz); (ii) we execute a detailed model comparison on sinusoidal data with switching frequencies (Sine) and (iii) Handwriting data (HW); (iv) finally, we perform an exploratory analysis with the NVSDS model on a basketball dataset (Basket). Below we describe in detail each of these datasets and analyse our results.

A. Baseline Models

A simple approach to dissect a time-series into different regimes is to perform clustering on the hidden states of a RNN model trained on the time-series. Although the resulting dissection is static, it provides a useful baseline to compare the dynamic dissection of our model. We thus use k-means clustering together with an LSTM cell — We shall refer to this model as R-k-Means. We also consider the rSLDS model [5], which we briefly introduced in Section III-A. For background, framework and code utilities we refer to [21]. Finally we consider a standard mixture of experts model (see e.g. [14]) wherein each expert is modeled via an LSTM (Eq. (11)), and where the gating mechanism (π_k^t in our notation) is defined as

$$\pi_k^t = \text{softmax}\left[g_{\theta_k}(\mathbf{h}_t^k, \mathbf{x}_t^k)\right]. \tag{20}$$

Here $\mathbf{h}_t^k \in \mathbb{R}^p$ is the expert's hidden state, $\hat{\mathbf{x}}_t^k \in \mathbb{R}^d$ is the expert's prediction and g_{θ_k} is given by an MLP with parameters θ_k. We refer to this model as MoE.

TABLE I: Hyperparameter specification for all experiments: Number of modes (K), hidden size per mode (p), hidden size of switching LSTM (s), attention dimension (a), number of layers in the switching MLP (N_l^s), and learning rate (λ).

Dataset	Model	K	p	s, a	N_l^s	λ
Lorenz	NVSDS	2	16	16	2	0.005
Sine	MoE NVSDS-EM NVSDS	2	16	32	1	0.005
HW	MoE NVSDS-EM NVSDS	4	64	32	2	0.005
Basket-all players	NVSDS-EM	16	32	32	2	0.005
Basket-single player	NVSDS-EM	8	64	32	2	0.005

B. Training Details

We choose the latent dimension and the discrete hidden state dimension of the rSLDS model to be two (four) for the Sine (Handwritting) dataset. For the R-k-Means model we train an LSTM with hidden size 16 (64) for the Sine (Handwriting) data. We set the number of clusters equal to the number of modes of the NVSDS model in the respective dataset. The hyperparameters of our models are given in Table I. We also set λ_e, the hyperparameter controling the entropy regularizer in NVSDS and NSVDD-EM models, to one in all our experiments. Regarding the optimization of the neural networks parameters we use ADAM [22].

C. Lorentz Attracttor

Being an epitome of a chaotic dynamical systems, the Lorentz system was introduced as a model of atmospheric convection [23]. The system is characterized by two chaotic attractors — that is, separated regions of phase space with different dynamic behaviors. We simulated the Lorenz system with different initial conditions and generated 100 trajectories which correspond to our training set. We set K, the number of modes (experts) to two.

Intuitively, one would expect each attractor to be identified as one of the modes (experts) of the model. In Fig. 2 we show the behavior of the NVSDS model on a test trajectory. As expected, each mode picks one attractor. We also generated trajectories in an open loop manner and observed an explicit separation between the two different attractors, forming the well known “butterfly" shape. These results, however, are meant to be a proof of concept. We do not show a comparison to the other models since all of them are able to correctly dissect the signal. Nevertheless our models (NVSDS, NVSDS-EM) are the first neural dynamical switching models successfully applied to this dataset.

D. Switching Oscillatory Dynamics

We now consider a synthetic dataset with two dynamical modes ($K = 2$): two sine functions with different frequencies, concatenated at regular time intervals. The switching behavior of the signal is therefore stationary. The goal is to dissect the signal into these modes, that is, uncover the two different frequencies. While this task is trivial for a human, and can

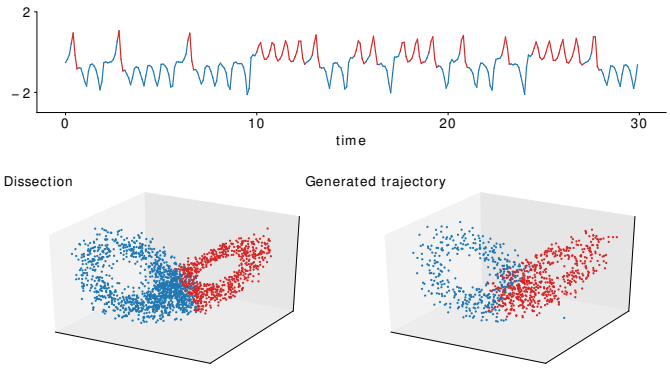

Fig. 2: Switching behavior in the Lorenz attractor. The upper panel corresponds to one dimension of the system over time. The solid line shows the one-step ahead prediction of the NVSDS model, the color corresponds to the dominating mode. Likewise the lower left panel shows one-step ahead prediction for the two-dimensional time series. The lower right panel corresponds to a generated trajectory.

easily be solved using signal processing methods, we have found it to be highly non-trivial for a SDS — which only sees one data point at a time. The guiding intuition is that, in a 2-expert model, each expert learns a different sine function. Fig. 3 shows our results. The R-k-Means, rSLDS and MoE models are not able to dissect the sine functions; they find instead a more local dissection into “up-down" states. The NVSDS and the NVSDS-EM models perfectly dissect the signal, showing the abstraction capabilities of these models. We asses these observations quantitatively by defining a binary target vector $\hat{\rho}$ indicating the dynamical mode which is present at a particular time-step. Evaluating the mean squared error between the predicted ρ and $\hat{\rho}$ yields a dissection error, which we average over multiple trials (different initial conditions). The NVSDS (0.09) and the NVSDS-EM (0.1) clearly outperform R-k-Means (0.26), rSLDS (0.4) and MoE (0.47).

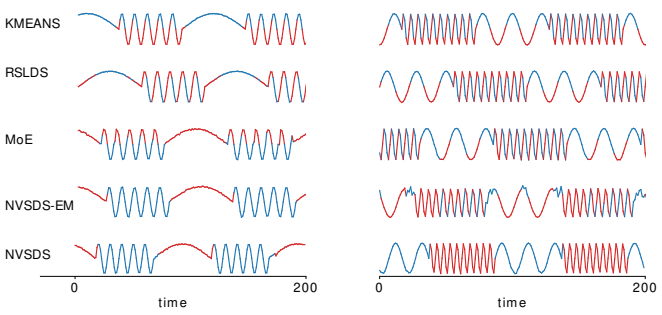

Fig. 3: Switching behavior in an oscillatory signal. The solid line shows the one-step ahead prediction of the different models. The color corresponds to the dominating expert (or chosen cluster in k-Means). The left and the right column show two signals with different frequencies. Only the NVSDS model successfully dissects the signal.

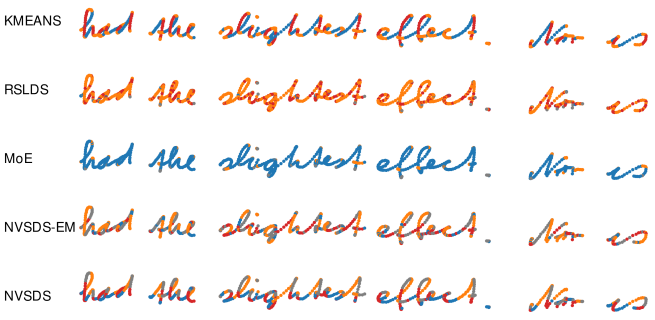

Fig. 4: Dissection of a handwriting signal. We show the original data colored according to the dominating expert.

The right column in Fig. 3 shows a second signal where both modes have full periods. The NVSDS model exclusively finds the correct dissection and learns to switch between frequencies. The comparison between the models suggests that both the non-linear transition function and the attention mechanism within the switching of the NVSDS play a key role in our results.

The main difficulty in this task is the existence of different signal dissections (minima) with very similar energies (in terms of the bound). The “up-down" solution found by the baseline models is but one of these. Indeed we find quite complex learning curves, with the models jumping back and fourth between different minima as the training progresses.

E. Handwriting

To show the performance of our methodology in datasets exhibiting the full range of complex behavior — non-linear, non-Markovian stochastic transitions — we concentrate on a handwriting signal from the IAM-OnDB dataset. This consists of 13,040 handwritten lines written by 500 writers [24]. We train on 900 sequences of 1100 data points and test the behavior on held-out test sequences. We train the model by feeding in the velocity of the signal. After training we select a test sequence and perform one-step ahead prediction. We color the signal according to the predicted expert activity. The intuition is that each mode (expert) should pick a specific stroke style like fast or curved strokes.

Fig. 4 shows a comparison of the results we obtain for the different models. With R-k-Means we basically find upwards, downwards and side-way movements, which serves as a good starting point. The rSLDS model is unable to dissect the signal into different strokes. Presumably this drawback is due to the limiting assumptions of linearity and Markovinity inherent in the rSLDS model. The MoE model fails to dissect the dynamics, a single expert dominates the signal. Now, while the NVSDS-EM and NVSDS share the same modular architecture as the MoE model, they also profit from both the attention mechanism and the internal history of the switch dynamics. This advantage leads to an interpretable dissection of the signal: for example, a downward movement for the NVSDS model is consistently captured by the same sequence of experts (grey-red colored) and a upwards movement to the right is captured by the expert indicated by the orange color. A detailed comparison

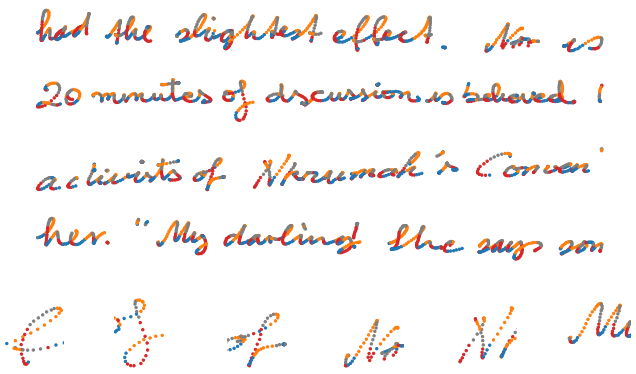

Fig. 5: Dissection of a handwriting signal for the NVSDS model for different sequences. The lower row shows particular letters from the complete sequences for easier comparison.

between the NVSDS-EM and the NVSDS model reveals that the NVSDS-EM model features a more stochastic switching behavior and therefore can be seen slightly inferior to the NVSDS model.

We further investigate the behavior of the NVSDS model in different writing styles in Fig. 5. Considering the same letter written by different writers (the three f’s in the lower row) we observe that our model consistently dissect the letter into the same sequence of experts. Furthermore, we compare the disection of the capital letter ’N’ (lower row). Here, the sequence of experts is different in the first stroke of the letter, which presumably reflects that these strokes where written in the opposite direction. In conclusion, our model is able to capture different strokes and thus identifying the building blocks of the individual letters.

Let us note that in this dataset we do not show the actual signal prediction of the models. The reason behind this is that we did not optimize our architecture to perfectly learn this non-continuous signal. Our actual predictions would result in non-readable writings. We are nonetheless only interested in a dissection of the given signal into meaningful dynamical modes.

F. Basketball Dataset

Finally, we perform an exploratory analysis by applying the NVSDS model to basketball data. Concretely, we consider player trajectories from games of the National Basket asso-

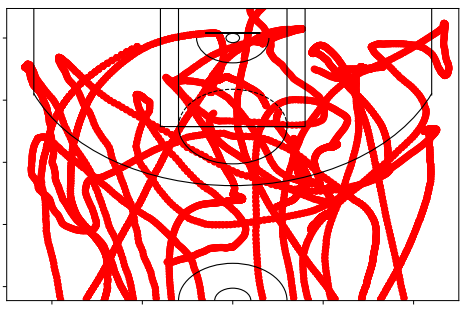

Fig. 6: Segment of example trajectory for a single player

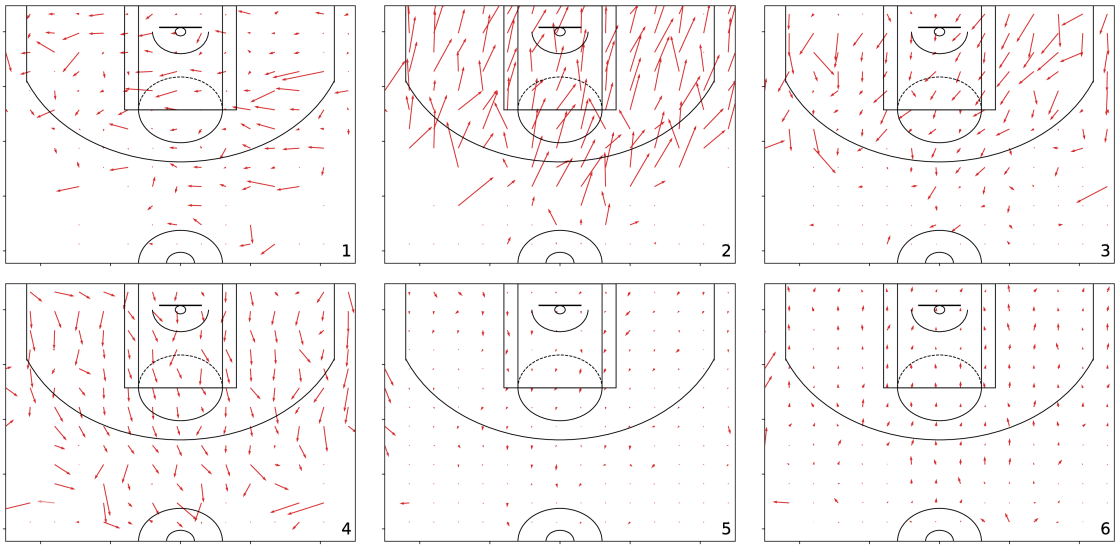

Fig. 7: Different movement patterns in the basketball dataset. We parcellate the court in $0.75\text{m} \times 0.75\text{m}$ quadrants. For each quadrant we pick the trajectories which 'belong' to a particular expert, i.e. for which the predicted ρ is largest. Then, we average the velocities and obtain for each expert an averaged flow field shown in the different panels.

ciation (NBA). First, we study the 2015/2016 season for the Detroit Pistons team. This dataset consists of the court positions (x, y) at time t, for all players in 22 games. We selected a total of 199 trajectories. Among those, 100 trajectories are selected as the train set. We test our model on the 99 remaining trajectories. Fig. 6 shows the complicated pattern emerging from one quarter (10000 points) of one of those trajectories. We train our model on the velocity of the players, which we compute from the data.

We present averaged expert specific vector fields (see caption for details) in Fig. 7. We train a model with $K = 16$ and present the most interpretable 6 patterns on the test set. We obtain a dissection of player movements based on speed and direction. For example, we uncover fast movements which cut to the upper right corner (expert 2) or horizontal movements below the basket (expert 1). We also observe movements away from the basket with different directions (expert 3 and expert 4), possibly showing defense behavior. Further, some experts pick up slow movements (expert 5 and 6) or even a standing still states. In general, the experts obtain similar movement patterns as found in [5]. In contrast to the experiments in [5], however, we train on larger datasets showing the scaling abilities of the neural model developed in this work.

We now aim to investigate the movements of individual players, whereby we focus on two players from the California Golden State Warriors, Andrew Bogut and Stephen Curry. The former one is playing as a center whereas the latter plays as a point guard. Traditionally, center players tend to position themselves near the basket, whereas point guards tend to cover the whole court, as to organize the general strategies. For each of the two players we create a different dataset picking 16 games for Bogut and 17 games for Curry and train on these dataset separately. We show the two clearest patterns in Fig. 8 and uncover vertical (1b) and horizontal movements (2b) near the basket for the center player Bogut. In contrast, the patterns for Stephen Curry extend over the whole court. This player is known for preferring 3-point shots slightly more than other players which is indicated by the pattern in expert 1c: Some velocity vectors seem to be tangential to the 3-point line, indicating that the player moves along the line to position himself for a shot. Furthermore, expert 2c seems to indicate counter-attack movements. Although qualitative in nature, these results highlight the ability of our model to provide insights into the dynamical modes of rather complex datasets.

VI. Discussion

In the present work we have provided a neural network solution to the problem of switching dynamical systems (SDS). Our methodology builds upon variational approximate inference for the categorical variables indexing of the dynamical modes. An attention mechanism, as well as an entropy regularizer are introduced to improve the detection of the modes. We applied our methodology successfully in diverse empirical datasets.

Beyond the results shown in this work we have also made a couple of observations which we believe to be important for SDS. We have noticed that dynamical modes with long-time scales are difficult to capture, e.g. to capture a complete letter in the handwriting dataset. There are several reasons for these behaviour. One is of technical nature: it was argued recently that the memory time-scale of the RNN is limited [25]. We suspect that this small timescale hinders our architecture to uncover switches on a larger time-scale. Another issue we have observed is the complicated energy-landscape: the time-scale of the switching is not explicitly incorporated in our loss function, therefore slow and fast switching solutions can have similar energy. Moreover, the statistics of the modes is crucial: e.g. in

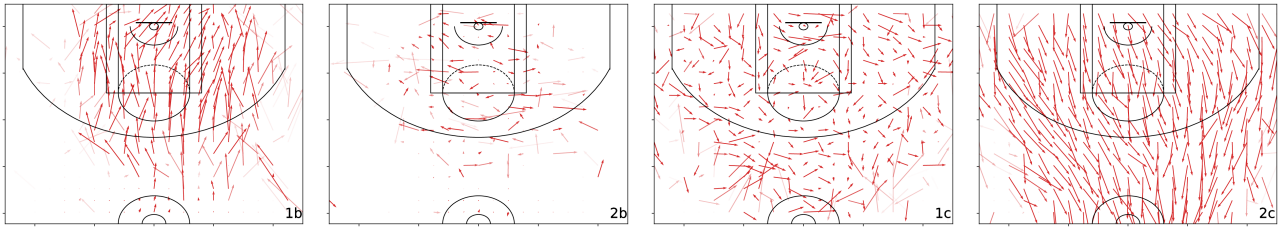

Fig. 8: Different movement patterns in the basketball dataset for individual Players. Visualisation is the same as Fig. 7 with $0.6\text{m} \times 0.6\text{m}$ quadrants. Upper row corresponds to Andrew Bogut, lower row to Stephen Curry.

the handwriting dataset one could consider a particular user letter style as an individual mode. If only few examples per user are provided, then this particular mode appears only rarely. The same holds true for a dynamical mode with a very long time-scale compared to the sample signal size. In this work we have focused on the dissection of complex signals. Our model learns the dynamics of the switches, which provides a strong competitive advantage against simple detection algorithms, say MoE, i.e. we are able to predict the sequence of modes not just detecting them. In terms of future work, a major goal is to exploit the NVSDS model to perform both outlier prediction and detection. We also plan to dispense with the RNN methodology and use Diluted Temporal Convolutions to overcome the time-scale problem.

ACKNOWLEDGMENT

The authors of this work were supported by the Fraunhofer Research Center for Machine Learning (RCML) and by the Competence Center for Machine Learning Rhine Ruhr (ML2R), which is funded by the Federal Ministry of Education and Research of Germany (grant no. 01|S18038A). Part of the work was also funded by the BIFOLD-Berlin Institute for the Foundations of Learning and Data (ref. 01IS18025A and ref 01IS18037A). We gratefully acknowledge this support.

REFERENCES

[1] A. J. Zeevi, R. Meir, and R. J. Adler, "Time series prediction using mixtures of experts," in *Advances in Neural Information Processing Systems 9*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. MIT Press, 1997, pp. 309–318. [Online]. Available: http://papers.nips.cc/paper/1203-time-series-prediction-using-mixtures-of-experts.pdf

[2] X. Xuan and K. Murphy, "Modeling changing dependency structure in multivariate time series," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 1055–1062.

[3] S. M. Oh, J. M. Rehg, T. Balch, and F. Dellaert, "Learning and inferring motion patterns using parametric segmental switching linear dynamic systems," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 103–124, 2008.

[4] Y. Saatçi, R. D. Turner, and C. E. Rasmussen, "Gaussian process change point models," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. Citeseer, 2010, pp. 927–934.

[5] S. Linderman, M. Johnson, A. Miller, R. Adams, D. Blei, and L. Paninski, "Bayesian learning and inference in recurrent switching linear dynamical systems," in *Artificial Intelligence and Statistics*, 2017, pp. 914–922.

[6] J. Nassar, S. Linderman, M. Bugallo, and I. M. Park, "Tree-structured recurrent switching linear dynamical systems for multi-scale modeling," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=HkzRQhR9YX

[7] E. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "Nonparametric bayesian learning of switching linear dynamical systems," in *Advances in Neural Information Processing Systems*, 2009, pp. 457–464.

[8] F. Stimberg, A. Ruttor, and M. Opper, "Bayesian inference for change points in dynamical systems with reusable states-a chinese restaurant process approach," in *Artificial Intelligence and Statistics*, 2012, pp. 1117–1124.

[9] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[10] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2980–2988.

[11] I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. C. Courville, and Y. Bengio, "A hierarchical latent variable encoder-decoder model for generating dialogues." in *AAAI*, 2017, pp. 3295–3301.

[12] P. Becker-Ehmck, J. Peters, and P. Van Der Smagt, "Switching linear dynamics for variational Bayes filtering," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, California, USA: PMLR, 09–15 Jun 2019, pp. 553–562.

[13] G. Ackerson and K.-S. Fu, "On state estima-tion in switching environments," *IEEE Transactionson Automatic Control*, vol. 15, no. 1, pp. 10–17, 1970.

[14] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.

[17] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[18] P. Schwab, D. Miladinovic, and W. Karlen, "Granger-causal attentive mixtures of experts: Learning important features with neural networks," *arXiv preprint arXiv:1802.02195*, 2018.

[19] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *ICLR-2017*, 2017.

[20] E. Bengio, P.-L. Bacon, J. Pineau, and D. Precup, "Conditional computation in neural networks for faster models," *arXiv preprint-arXiv:1511.06297*, 2015.

[21] S. Linderman, https://github.com/slinderman/recurrent-slds.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] E. N. Lorenz, "Deterministic Nonperiodic Flow." *Journal of Atmospheric Sciences*, vol. 20, pp. 130–148, Mar. 1963.

[24] M. Liwicki and H. Bunke, "Iam-ondb-an on-line english sentence database acquired from handwritten text on a whiteboard," in *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*. IEEE, 2005, pp. 956–961.

[25] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.