Recursive Variational Bayesian Dual Estimation for Nonlinear Dynamics and Non-Gaussian Observations

Yuan Zhao and Il Memming Park
Department of Neurobiology and Behavior
Department of Applied Mathematics and Statistics
Institute for Advanced Computational Science
Stony Brook University, NY 11794
`{yuan.zhao, memming.park}@stonybrook.edu`

Abstract

State space models provide an interpretable framework for complex time series by combining an intuitive dynamical system model with a probabilistic observation model. We developed a flexible online learning framework for latent nonlinear state dynamics and filtered latent states. Our method utilizes the stochastic gradient variational Bayes method to jointly optimize the parameters of the nonlinear dynamics, observation model, and the recognition model. Unlike previous approaches, our framework can incorporate non-trivial observation noise models and infer in real-time. We test our method on point process observations driven by continuous attractor dynamics, demonstrating its ability to recover the phase portrait, filtered trajectory, and produce long-term predictions for neuroscience applications.

1 Introduction

Given observed time series $\{\mathbf{y}_t\}$ and input $\{\mathbf{u}_t\}$, can we build a state space model that captures the underlying nonlinear dynamics responsible for its generation [1]? More specifically, we would like to identify a continuous nonlinear process that captures the temporal structure, and an instantaneous noisy observation process:

$$\mathbf{y}_t \sim P(\mathbf{y} \mid F(\mathbf{x}_t, \mathbf{u}_t)) \qquad \text{(observation model)} \qquad (1a)$$
$$\dot{\mathbf{x}} = G(\mathbf{x}_t, \mathbf{u}_t) \qquad \text{(state dynamics)} \qquad (1b)$$

where F and G are continuous functions, and P denotes a probability distribution. Such continuous state space model is natural in many applications where the changes are slow and follow physical laws and constraints (e.g., object tracking). Our target application is neural data analysis, where inferring state space models underlying neural activity can provide insights into neural dynamics [2, 3, 4, 5], neural computation [6, 7, 8], and development of neural prosthetics and treatment through feedback control [9, 10].

It is often more convenient to formulate the state dynamics in discrete time with noise due to the uniformly sampled time series:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + g(\mathbf{x}_t, \mathbf{u}_t) + \boldsymbol{\epsilon}_t \qquad \text{(state dynamics)} \qquad (2)$$

where $\boldsymbol{\epsilon}_t$ captures unobserved perturbations of the state $\mathbf{x}_t$. The noise in dynamics and observation are essential in finding a concise description (low-dimensional phase portrait) of the observations [11].

If the nonlinear state space model is fully specified, Bayesian methods can estimate the latent states (either the filtering distribution $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$ or the smoothing distribution $p(\mathbf{x}_{1:t} \mid \mathbf{y}_{1:t})$), predict future states $p(\mathbf{x}_{t:t+s} \mid \mathbf{y}_{1:t})$, and predict observations $p(\mathbf{y}_{t+1:t+s} \mid \mathbf{y}_{1:t})$ for $s > 0$ [12, 13]. However, in many applications, the challenge is in learning the parameters of the state space model (a.k.a. the system identification problem). Learning both the latent state trajectory and the latent (nonlinear) state space model is known as the *dual estimation problem* [14]. Expectation maximization (EM) based methods have been widely used in practice [15, 16], and more recently variational autoencoder methods [17, 18, 19, 20, 21, 22] have been proposed, all of which are for the offline batch analysis.

In this paper, we are interested in real-time signal processing and state-space control setting where we need online algorithms that can recursively solve the dual estimation problem on streaming observations. A popular solution to this problem exploits the fact that online state estimators for nonlinear state space models such as extended Kalman filter (EKF) or unscented Kalman filter (UKF) can be used for nonlinear regression formulated as a state space model:

$$\theta_{t+1} = \theta_t + \eta_t \qquad \text{(parameter)} \tag{3}$$

$$\mathbf{z}_t = H(\mathbf{x}_t, \theta_t) + \epsilon_t \qquad \text{(regression)} \tag{4}$$

where H is a function approximator parameterized by θ that maps $\mathbf{x}$ to $\mathbf{z}$. Therefore, by augmenting the state space with the parameters, one can build an online dual estimator using nonlinear Kalman filters [23, 24]. However, they involve coarse approximation of Bayesian filtering, involve many hyperparameters, do not take advantage of modern stochastic gradient optimizers, and are not easily applicable to arbitrary observation likelihoods. There are also closely related online version of EM-type algorithms [11] that share similar concerns. In this paper, we derive a black-box inference framework applicable to a wide range of nonlinear state space models that is truly online meaning, that is, the computational demand of the algorithm is constant per time step.

2 Variational Principle for Online Dual Estimation

The crux of recursive Bayesian filtering is updating the posterior over the latent state one time step at a time.

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) = \underbrace{p(\mathbf{y}_t \mid \mathbf{x}_t)}_{\text{likelihood}} \, \underbrace{p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})}_{\text{prior at time } t} \Big/ \underbrace{p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})}_{\text{marginal likelihood}} \tag{5a}$$

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) = \int \underbrace{p(\mathbf{x}_t \mid \mathbf{x}_{t-1})}_{\text{state dynamics}} \, \underbrace{p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})}_{\text{previous posterior}} \, d\mathbf{x}_{t-1} \tag{5b}$$

where the input $\mathbf{u}_t$ is omitted for brevity. Unfortunately, the exact calculations of (5) are not tractable in general, especially for a nonlinear dynamics model and a non-conjugate likelihood. We derive a recursive variational Bayesian filter by deriving a lower bound for the marginal likelihood. Let $q(\mathbf{x}_t)$ denote an arbitrary probability measure which will eventually approximate $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$. From (5a),

$$\log p(\mathbf{y}_t \mid \mathbf{y}_{1:t-1})$$

$$= \mathbb{E}_{q(\mathbf{x}_t)}\left[\log \frac{p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) q(\mathbf{x}_t)}{p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) q(\mathbf{x}_t)}\right] \tag{6a}$$

$$= \underbrace{\mathbb{E}_{q(\mathbf{x}_t)}\left[\log p(\mathbf{y}_t \mid \mathbf{x}_t)\right]}_{\text{reconstruction log-likelihood}} - \mathbb{D}_{\mathrm{KL}}(q(\mathbf{x}_t) \,\|\, p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})) + \underbrace{\mathbb{D}_{\mathrm{KL}}(q(\mathbf{x}_t) \,\|\, p(\mathbf{x}_t \mid \mathbf{y}_{1:t}))}_{\text{variational gap \#1}} \tag{6b}$$

$$\geq \mathbb{E}_{q(\mathbf{x}_t)}\left[\log p(\mathbf{y}_t \mid \mathbf{x}_t)\right] + \mathbb{E}_{q(\mathbf{x}_t)}\left[\log p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})\right] + \underbrace{H(q(\mathbf{x}_t))}_{\text{entropy}} \tag{6c}$$

$$= \mathbb{E}_{q(\mathbf{x}_t)}\left[\log p(\mathbf{y}_t \mid \mathbf{x}_t)\right] + \mathbb{E}_{q(\mathbf{x}_t)}\left[\log \mathbb{E}_{p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})}\left[p(\mathbf{x}_t \mid \mathbf{x}_{t-1})\right]\right] + H(q(\mathbf{x}_t)) \tag{6d}$$

$$= \mathbb{E}_{q(\mathbf{x}_t)}\left[\log p(\mathbf{y}_t \mid \mathbf{x}_t)\right] + \mathbb{E}_{q(\mathbf{x}_t)}\left[\mathbb{E}_{p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})}\left[\log p(\mathbf{x}_t \mid \mathbf{x}_{t-1})\right]\right] + \underbrace{\mathbb{E}_{q(\mathbf{x}_t)}\left[\mathbb{D}_{\mathrm{KL}}(p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1}) \,\|\, p(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{y}_{1:t-1}))\right]}_{\text{variational gap \#2}} + H(q(\mathbf{x}_t)) \tag{6e}$$

$$\geq \mathbb{E}_{q(\mathbf{x}_t)}\left[\log p(\mathbf{y}_t \mid \mathbf{x}_t)\right] + \mathbb{E}_{q(\mathbf{x}_t)} \mathbb{E}_{p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})}\left[\log p(\mathbf{x}_t \mid \mathbf{x}_{t-1})\right] + H(q(\mathbf{x}_t)) \tag{6f}$$

$$\approx \mathbb{E}_{q(\mathbf{x}_t)}\left[\log p(\mathbf{y}_t \mid \mathbf{x}_t)\right] + \underbrace{\mathbb{E}_{q(\mathbf{x}_t)} \mathbb{E}_{q(\mathbf{x}_{t-1})}\left[\log p(\mathbf{x}_t \mid \mathbf{x}_{t-1})\right]}_{\text{dynamics log-likelihood}} + H(q(\mathbf{x}_t)) =: \mathcal{L} \tag{6g}$$

where $\mathbb{D}_{\mathrm{KL}}$ denotes the Kullback-Leibler divergence, and in the last step, we plugged in the variational posterior from the last step to form a recursive estimation.

Online variational inference is achieved by maximizing this approximate lower bound objective $\mathcal{L}$ for the parameters of the generative model ($p(\mathbf{y}_t \mid \mathbf{x}_t)$ and $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$) and the variational posterior distribution $q(\mathbf{x}_t)$ provided that $q(\mathbf{x}_{t-1})$ is estimated from the previous time step. Maximizing $\mathcal{L}$ is equivalent to minimizing the two variational gaps: (1) the variational filtering posterior has to be close to the true filtering posterior, and (2) the filtering posterior from the previous step needs to be close to $p(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{y}_{1:t-1})$. Note that this second gap is invariant to $q(\mathbf{x}_t)$ if $p(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{y}_{1:t-1}) = p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})$, that is, the one-step backward smoothing distribution is identical to the filtering distribution.

On the flip side, intuitively, there are three components in $\mathcal{L}$ that are jointly maximized: (1) reconstruction log-likelihood which is maximized if $q(\mathbf{x}_t)$ concentrates around the maximum likelihood estimate given only $\mathbf{y}_t$, (2) the dynamics log-likelihood which is maximized if $q(\mathbf{x}_t)$ concentrates at around the maximum of $\mathbb{E}_{q(\mathbf{x}_{t-1})}\left[\log p(\mathbf{x}_t \mid \mathbf{x}_{t-1})\right]$, and (3) the entropy term that expands the posterior and keeps it from collapsing to a point mass.

We choose the variational posterior over the state $q(\mathbf{x}_t)$ to be a multivariate normal with diagonal covariance $\mathcal{N}(\boldsymbol{\mu}_t, \operatorname{diag}(\mathbf{s}_t))$. To amortize the computational cost of optimization to obtain the best $q(\mathbf{x}_t)$ on each time step, we employ the variational autoencoder architecture [25] to parametrize $q(\mathbf{x}_t)$ with a recognition model. Intuitively, the recognition model embodies the optimization process of finding $q(\mathbf{x}_t)$, that is, it performs an approximate Bayesian filtering computation (in constant time) of (5) according to the objective function $\mathcal{L}$. We use a recursive recognition network model that maps $q(\mathbf{x}_{t-1})$ and $\mathbf{y}_t$ to $q(\mathbf{x}_t)$. In particular, the recognition model is a deterministic recurrent neural network (RNN; with no extra hidden state):

$$\boldsymbol{\mu}_t, \mathbf{s}_t = \mathbf{h}(\mathbf{y}_t, \mathbf{u}_{t-1}, \boldsymbol{\mu}_{t-1}, \mathbf{s}_{t-1}) \qquad \text{(recognition RNN)} \tag{7}$$

We use a simple multi-layer perceptron as $\mathbf{h}$. Note that the Markovian architecture of the recognition model reflects the Markovian structure of filtering computation (c.f., smoothing networks often use bidirectional RNN [26] or graphical models [17, 20]).

We use the reparameterization trick and stochastic variational Bayes [27, 28]: we rewrite the expectations over q as expectation over a standard normal random variable, and we use a single sample for each time step. Hence, in practice, we optimize the following objective function,

$$\hat{\mathcal{L}} = \log p(\mathbf{y}_t \mid \tilde{\mathbf{x}}_t, \mathbf{u}_t) + \log p(\tilde{\mathbf{x}}_t \mid \tilde{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}) + H(q(\mathbf{x}_t)) \tag{8}$$

where $\tilde{\mathbf{x}}_t$ and $\tilde{\mathbf{x}}_{t-1}$ represents symbols sampled from $q(\mathbf{x}_t)$ and $q(\mathbf{x}_{t-1})$ respectively. Thus, our method can handle arbitrary observation and dynamics model unlike dual form nonlinear Kalman filtering methods.

Denote the set of all parameters by $\boldsymbol{\Theta}$ from both the generative and recognition model. The objective in Eq. (8) is differentiable w.r.t. the model parameters. We optimize the objective function $\hat{\mathcal{L}}$ through gradient ascent (using Adam [28]) implemented within TensorFlow [29]. Algorithm 1 is an overview of the recursive estimation algorithm. We outline the algorithm for a single vector time series, but we can filter multiple sequences with a common state space model simultaneously, in which case the gradients are averaged across the instantiations.

Algorithm 1 Online Variational Dual Estimation (single step update)

```
procedure FILTER($\mathbf{y}_t, \mathbf{u}_{t-1}, \boldsymbol{\mu}_{t-1}, \mathbf{s}_{t-1}, \boldsymbol{\Theta}$)
    $\epsilon_t \leftarrow \mathcal{N}(\mathbf{0}, \mathbf{I}),\ \epsilon_{t-1} \leftarrow \mathcal{N}(\mathbf{0}, \mathbf{I})$        ▷ Draw random samples
    $\boldsymbol{\mu}_t, \mathbf{s}_t \coloneqq \mathbf{h}(\mathbf{y}_t, \mathbf{u}_{t-1}, \boldsymbol{\mu}_{t-1}, \mathbf{s}_{t-1})$        ▷ Symbolic assignments
    $\tilde{\mathbf{x}}_t \coloneqq \boldsymbol{\mu}_t + \mathbf{s}_t^{1/2}\epsilon_t$
    $\tilde{\mathbf{x}}_{t-1} \coloneqq \boldsymbol{\mu}_{t-1} + \mathbf{s}_{t-1}^{1/2}\epsilon_{t-1}$
    Update $\boldsymbol{\Theta}$ with $\nabla_{\boldsymbol{\Theta}} \hat{\mathcal{L}}(\boldsymbol{\Theta}; \mathbf{y}_t, \tilde{\mathbf{x}}_t, \tilde{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1})$        ▷ Gradient ascent
    return $\boldsymbol{\mu}_t, \mathbf{s}_t$ and $\boldsymbol{\Theta}$
end procedure
```

Note that this algorithm has constant time complexity per time step.

3 Application to Latent Neural Dynamics

Our primary application is real-time neural interfaces where a population of neurons are recorded while a low-dimensional stimulation is delivered [30, 31]. Latent state space modeling of such neural time series have been successful in describing population dynamics [32, 33]. Moreover, models neural computation are often described as dynamical systems, for exmaple, attractor dynamics where the convergence to one of the attractors represents the result of computation [8]. Here we propose a parameterization and tools for visualization of the model suitable for studying neural dynamics and building neural interfaces [8].

3.1 Parameterization of the generative model

We consider high temporal resolution spike train data, where each time bin has at most one action potential. Hence our observed time series $\mathbf{y}_t$ is a stream of sparse binary vectors. All analysis are done with a 1 ms time bin in this study.

Our generative model assumes that the spike train observation $\mathbf{y}_t$ is sampled from a probability distribution P determined by the state $\mathbf{x}_t$ though a linear-nonlinear map possibly together with extra parameters at each time t,

$$\mathbf{y}_t \sim P(f(\mathbf{C}\mathbf{x}_t + \mathbf{b})) \tag{9}$$

where f is a point nonlinearity $\mathbb{R} \mapsto \mathbb{R}^+$. We use the canonical link $f(\cdot) = \exp(\cdot)$ for point process observation in this study. Note that this model is not identifiable since $\mathbf{C}\mathbf{x}_t = (\mathbf{C}\mathbf{R})(\mathbf{R}^{-1}\mathbf{x}_t)$ where $\mathbf{R}$ is an arbitrary invertible matrix. Also, the mean of $\mathbf{x}_t$ can be traded off with the bias term in $\mathbf{b}$. It is straightforward to include more additive exogenous variables or history-filter for refractory period and stimulation artifacts.

We propose to use a specific parametrization for state dynamics with additive state transition function and locally linear input interaction as a special case of Eq. (2),

$$\begin{aligned}
\mathbf{x}_{t+1} &= \mathbf{x}_t + g(\mathbf{x}_t) + \mathbf{B}(\mathbf{x}_t)\mathbf{u}_t + \boldsymbol{\epsilon}_{t+1} && \text{(10a)}\\
g(\mathbf{x}_t) &= \mathbf{W}_g \boldsymbol{\phi}(\mathbf{x}_t) && \text{(10b)}\\
\operatorname{vec}(\mathbf{B}(\mathbf{x}_t)) &= \mathbf{W}_B \boldsymbol{\phi}(\mathbf{x}_t) && \text{(10c)}\\
\boldsymbol{\epsilon}_t &\sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}) && \text{(10d)}\\
\mathbf{x}_0 &\sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}) && \text{(10e)}
\end{aligned}$$

where $\boldsymbol{\phi}(\cdot)$ is a vector of r continuous basis functions, $\boldsymbol{\phi}(\cdot) = (\phi_1(\cdot), \ldots, \phi_r(\cdot))^\top$. In this study, we use squared exponential radial basis functions [11, 8],

$$\phi_i(\mathbf{x}) = \exp\left(-\frac{1}{2}\gamma_i \|\mathbf{x} - \mathbf{c}_i\|_2^2\right) \tag{11}$$

with centers $\mathbf{c}_i$ and corresponding inverse squared kernel width γ_i.

Let n, m, p, q, r denote the dimensions of observation, latent space, input, the numbers of hidden units and radial basis functions for this specific parametrization. The time complexity of our algorithm is $\mathcal{O}(mpr + n(m + p + q))$. If we compare this to an efficient offline algorithm such as PLDS [36] run repeatedly for every new observation (online mode), its time complexity is $\mathcal{O}(t \cdot (m^3 + mn))$ per time step at time t which grows as time passes, making it impractical for real-time application.

3.2 Phase portrait analysis

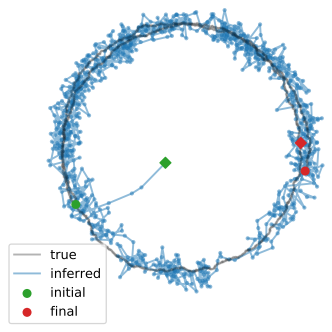
The function $g(\mathbf{x})$ directly represents the velocity field of an underlying smooth dynamics (1b) in the absence of input [11, 8]. We visualize the phase portrait of the estimated dynamics that consists of the vector field, example trajectories, and estimated dynamical features (namely fixed points). We numerically identify candidate fixed points $\mathbf{x}^*$ that satisfy $g(\mathbf{x}^*) \approx 0$. For the simulation studies, we do an affine transformation to orient phase portrait to match the canonical equations in the main text.

4 Simulated Experiments

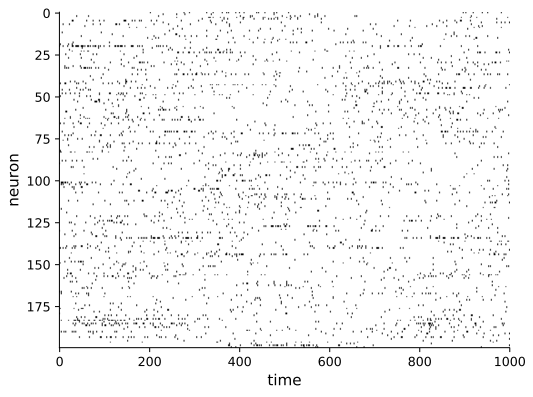
For the purposes of visualization, we chose to simulate from two dimensional dynamical systems, and chose $n = 200$ neurons to generate spikes via state dynamics driven point processes. Average firing rate of the simulated spike trains were kept around 20~40 Hz.

Many theoretical models have been proposed in theoretical neuroscience to represent different modes of computation. We apply the proposed method to a two such low-dimensional models: the ring attractor model as a model of internal head direction representation, and an nonlinear oscillator as a model of rhythmic population-wide activity. We refer to their conventional formulations under different coordinate systems, but our simulations and inferences are all done in Cartesian coordinates.

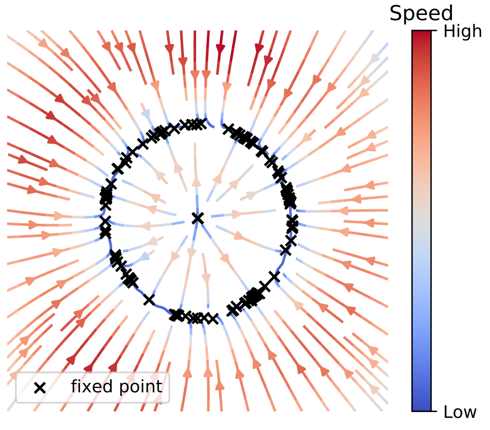
The approximate posterior distribution is defined recursively in Eq. (7) as diagonal Gaussian with mean and variance determined by corresponding observation, input and previous step via a recurrent neural network. We use one hidden layer MLP in this study. Typically the state noise variance σ^2 is unknown and has to be estimated from data. To be consistent with Eq. (10e), we set the starting value of σ^2 to be 1, and hence $\boldsymbol{\mu}_0 = \mathbf{0}, \mathrm{s}_0 = \mathbf{I}$. We initialize the loading matrix $\mathbf{C}$ by factor analysis, and normalized $\mathbf{C}$ every iteration to keep the system identifiable.



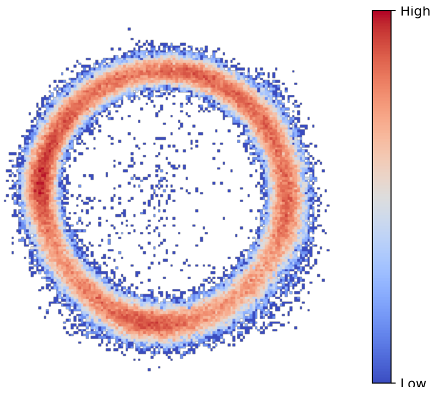
(a)



(b)



(c)



(d)

Figure 1: Ring attractor model. (a) One latent trajectory (black) in the training set and the corresponding filtered mean μ_t (blue). (b) Raster plot of the spike train corresponding to the left trajectory. (c) Velocity field reconstructed from the trained proposed model. The color contour indicates the speed and the arrows represent the directions. The black crosses are candidate fixed points obtained from inferred dynamics. Note the collection of fixed points around the ring shape. The central fixed point is unstable. (d) Density of the posterior means. The density of inferred means of all trajectories in the training set. The higher it is, the more confidence we have on the inferred dynamics where we have more data.

4.1 Ring attractor

We study the following two-variable ring attractor system

$$\begin{aligned}\tau_r \dot{r} &= r_0 - r, \\ \tau_\theta \dot{\theta} &= I,\end{aligned} \tag{12}$$

where θ represents the direction driven by input I, and r is the radial component representing the overall activity in the bump. We simulate 100 trajectories of 1000 steps with step size 0.1, $r_0 = 1$, $\tau_r = 1$, $\tau_\theta = 1$ plus Gaussian noise (std $= 0.005$). We use strong input (tangent drift) to keep the trajectories flowing around the ring clockwise or counter-clockwise respectively. We use 20 radial basis functions for dynamics model and 100 hidden units for the recognition model.

Figure 1a illustrates one trajectory (black) and its posterior mean (blue). These two trajectories start at green circle and diamond respectively and end at the red markers. The inference starts near the center (origin) that is relatively far from the true location because the initial posterior is set at zero mean. The final states are very close which implies that the recognition model works well. Figure 1c shows the reconstructed velocity field by the model. We visualize the speed as colored contour and the direction by red arrows. We can see the velocity toward the ring attractor and the speed is as lower as closer to the ring. The model also identifies a number of fixed points arranged around the ring attractor via numerical roots finding. Figure 1d shows the distribution of posterior means of all data in the state space. The estimated generative model has higher state noise, and also higher attraction around the ring attractor. This is also evident if we compare the simulated trajectories from the inferred dynamics (Figure 2).

Figure 3 shows the three components of (6g) and the objective lower bound clearly, demonstrating the convergence of the algorithm. We can see each component reaches a plateau within 2000 sec. As the reconstruction and dynamics log-likelihoods increase, the recognition model and dynamical model are getting more accurate while the decreasing entropy indicates the increasing confidence (inverse posterior variance) on the inferred latent states. The average computation time of a dual estimation step is 1.6 ms on a machine with Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz (32 cores) and 126GB RAM (no GPU).

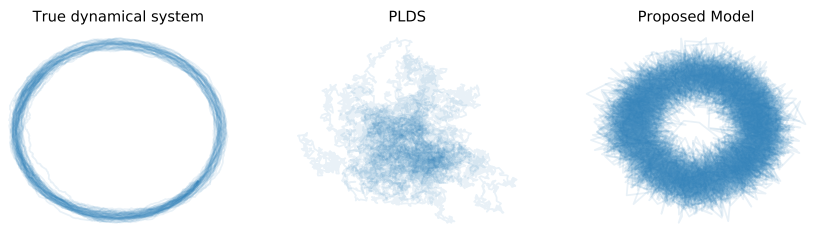


Figure 2: Ring attractor simulation. We simulated 20 trajectories each using the true ring attractor, trained PLDS and proposed model. The proposed model captures the ring attractor but has a higher state noise, while PLDS failed to learn the dynamics.

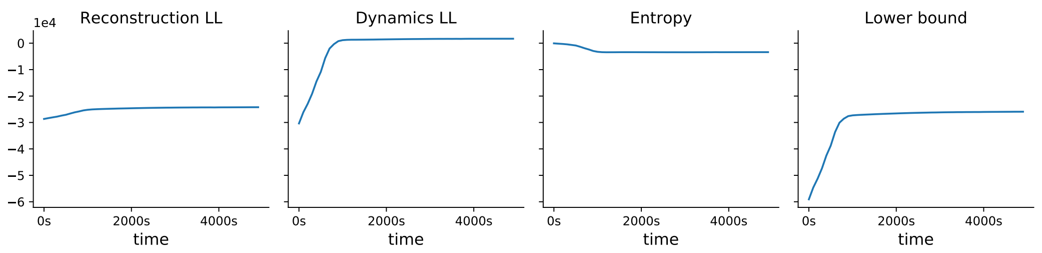


Figure 3: Convergence on the ring attractor. We display the three components of the objective lower bound: reconstruction log-likelihood, dynamics log-likelihood, entropy, and the lower bound itself (Eq. (6)) from left to right. The x-axes represents the number of elapsed filtering steps.

4.2 Nonlinear oscillator

We use a 2-dimensional relaxation oscillator with the following nonlinear state dynamics:

$$\begin{aligned} \dot{v} &= v(a-v)(v-1) - w + I, \\ \dot{w} &= bv - cw, \end{aligned} \tag{13}$$

where v is the membrane potential, w is a recovery variable and I is the magnitude of stimulus current in modeling single neuron biophysics [34]. This model was used to model global brain state that fluctuates between two levels of excitability in anesthetized cortex [35]. We use the following parameter values $a = -0.1$, $b = 0.01$, $c = 0.02$ and $I = 0.1$ to simulate 100 trajectories of 1000 steps with step size 0.5 and Gaussian noise (std=0.002). At this regime, unlike the ring attractor, the spontaneous dynamics is a periodic oscillation, and the trajectory follows a limit cycle..

We use 20 radial basis functions for dynamics model and 100 hidden units for recognition model. While training the model, we do not feed input and expect the model to learn the oscillator.

We also reconstruct the phase portrait (Fig. 4c). The two dashed lines are the theoretical nullclines of the true model on which the velocity of corresponding dimension is zero. The reconstructed field shows a low speed valley overlapping with the nullcline especially on the right half figure. The intersection of two nullclines is a unstable fixed point. We can see the identified fixed point is close to the intersection.

We run a long-term prediction using the proposed model. The prediction contains both latent trajectory and observation which last for T steps by sampling from:

$$p(\mathbf{x}_{t+1:T} \mid \mathbf{y}_{1:t}) = \int p(\mathbf{x}_{t+1:T} \mid \mathbf{x}_t) q(\mathbf{x}_t) \, \mathrm{d}\mathbf{x}_t \tag{14a}$$

$$p(\mathbf{y}_{t+1:T} \mid \mathbf{y}_{1:t}) = \int p(\mathbf{y}_{t+1:T} \mid \mathbf{x}_{t+1:T}) p(\mathbf{x}_{t+1:T} \mid \mathbf{y}_{1:t}) \, \mathrm{d}\mathbf{x}_{t+1:T} \tag{14b}$$

given estimated parameters without seeing the data $\mathbf{y}_{t+1:T}$ during these steps ($T = 1000 = 1$ sec). We give the prediction in Figure 4e. The upper row is the true latent trajectory and corresponding observations. The light colored half are the 1000 steps before prediction and the solid colored are during prediction. We only show the observation in the prediction period. The lower row is the filtered trajectory and prediction by our proposed method. The prediction begins after the 2000 step.

One of the popular latent process modeling tools for point process observation that can make prediction is Poisson Linear Dynamical System (PLDS) [36] which assumes latent linear dynamics. We compare PLDS fit with EM on its long-term prediction on both the states and spike trains (Fig. 4e).

We also tried to apply the same data to the unscented Kalman filter (UKF; result not shown) for the dual estimation. However, the UKF is not flexible with arbitrary likelihood e.g. Poisson. Even when we used true dynamics equation instead of the radial basis function network, it suffers numerical singularity issues, requires fine hyperparameter tunning, and initialization close to the true value.

5 Discussion

We proposed an online algorithm for recursive variational Bayesian inference for both system identification and filtering. Thanks to the recognition network that amortizes the optimization cost over time, our online algorithm can achieve real-time performance. Neural dynamics is highly nonlinear, and neural activity is sparse and non-Gaussian. In closed-loop neurophysiological setting, real-time adaptive algorithms are extremely valuable. It is easy to incorporate spike history filter [37], and a simple model of electrical stimulation artifacts to make the proposed method more useful in practice. Furthermore, a more complex (or perhaps deep) observation model can be used without rederiving the algorithm. Our method fills the gap where previously there wasn't an appropriate tool.

Concise description of collective neural activity can be crucial to understanding cognitive behavior. We hope that this tool enables on-the-fly analysis of high-dimensional neural spike train experiments. Clinically, a nonlinear state space model provides a basis for feedback control as a potential treatment for neurological diseases that arise from diseased dynamical states.

One weakness of the current online algorithm is its slow convergence. Since the RNN recognition model is performing an approximate Bayesian filtering, we could pretrain with true Bayesian inference

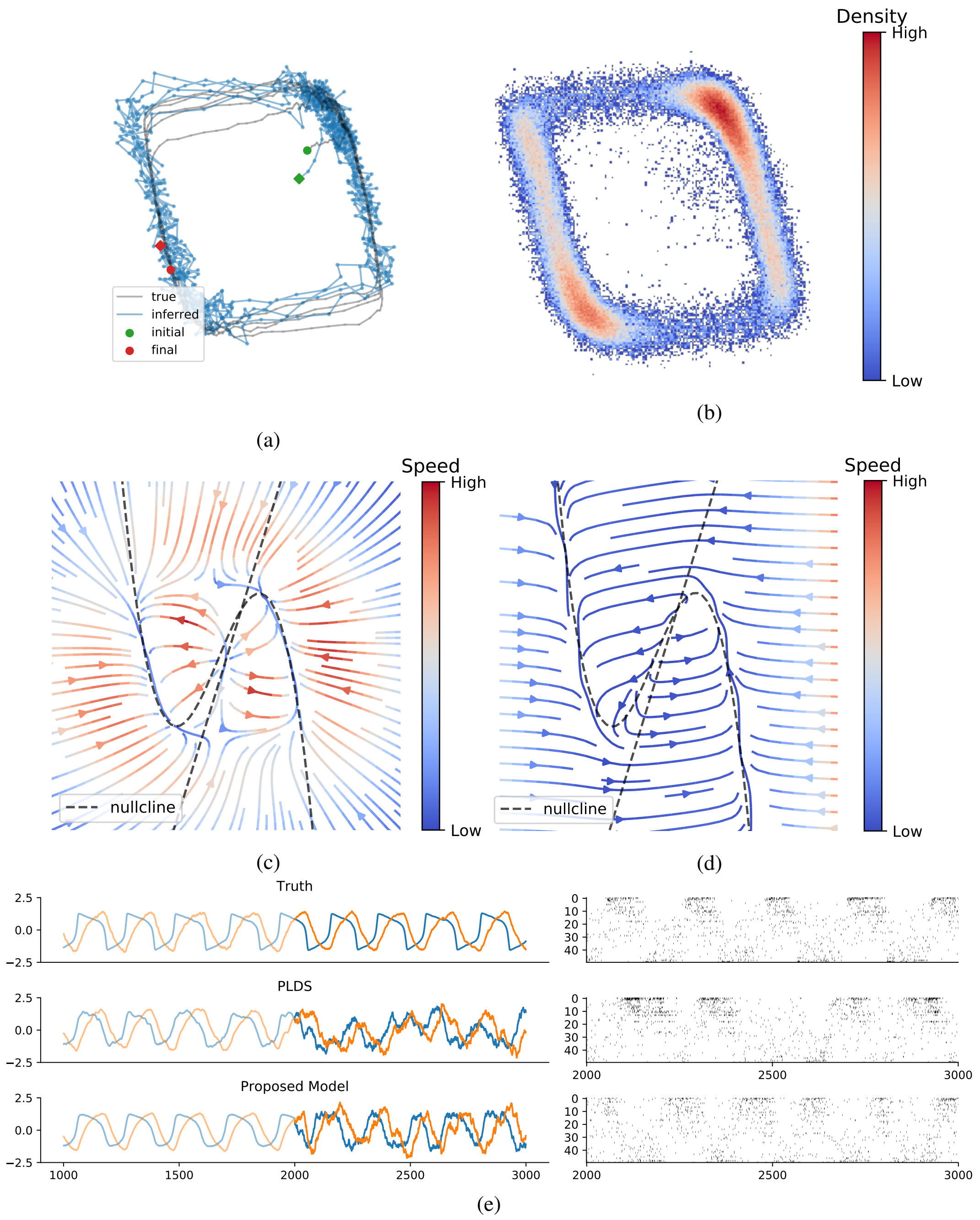


Figure 4: Nonlinear oscillator (FitzHugh-Nagumo) model. (a) One latent trajectory in the training set and its inferred mean (first 1000 steps). (b) Density of inferred means. Most of the inferred trajectory lie on the oscillation path. (c) Velocity field reconstructed by the inferred dynamical system. The dashed lines are two nullclines of the true model on which the gradients are zero so as the velocity. (d) Velocity field of the true dynamical system. (e) Prediction of the trajectory in (a).

examples under a pretrained generative model. We leave the issue of better recognition model architecture and initialization as future work.

References

[1] S. Haykin and J. Principe. Making sense of a complex world [chaotic events modeling]. *IEEE Signal Processing Magazine*, 15(3):66–81, May 1998.

[2] M. Breakspear. Dynamic models of large-scale brain activity. *Nature Neuroscience*, 20(3):340–352, February 2017.

[3] J. C. Kao, P. Nuyujukian, S. I. Ryu, et al. Single-trial dynamics of motor cortex and their applications to brain-machine interfaces. *Nature Communications*, 6:7759+, July 2015.

[4] L. Paninski, Y. Ahmadian, D. G. G. Ferreira, et al. A new look at state-space models for neural data. *Journal of computational neuroscience*, 29(1-2):107–126, August 2010.

[5] B. M. Yu, J. P. Cunningham, G. Santhanam, et al. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Journal of neurophysiology*, 102(1):614–635, July 2009.

[6] V. Mante, D. Sussillo, K. V. Shenoy, and W. T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84, November 2013.

[7] D. Sussillo and O. Barak. Opening the black box: Low-Dimensional dynamics in High-Dimensional recurrent neural networks. *Neural Computation*, 25(3):626–649, December 2012.

[8] Y. Zhao and I. M. Park. Interpretable nonlinear dynamic modeling of neural trajectories. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[9] L. R. Hochberg, M. D. Serruya, G. M. Friehs, et al. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442(7099):164–171, July 2006.

[10] S. Little and P. Brown. What brain signals are suitable for feedback control of deep brain stimulation in parkinson's disease? *Annals of the New York Academy of Sciences*, 1265(1):9–24, August 2012.

[11] S. Roweis and Z. Ghahramani. *Learning nonlinear dynamical systems using the expectation-maximization algorithm*, pages 175–220. John Wiley & Sons, Inc, 2001.

[12] Y. Ho and R. Lee. A Bayesian approach to problems in stochastic estimation and control. *IEEE Transactions on Automatic Control*, 9(4):333–339, October 1964.

[13] S. Särkkä. *Bayesian filtering and smoothing*. Cambridge University Press, 2013.

[14] S. S. Haykin. *Kalman filtering and neural networks*. Wiley, 2001.

[15] Z. Ghahramani and S. T. Roweis. Learning nonlinear dynamical systems using an EM algorithm. In M. J. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 431–437. MIT Press, 1999.

[16] H. Valpola and J. Karhunen. An unsupervised ensemble learning method for nonlinear dynamic State-Space models. *Neural Computation*, 14(11):2647–2692, November 2002.

[17] E. Archer, I. M. Park, L. Buesing, J. Cunningham, and L. Paninski. Black box variational inference for state space models. *ArXiv e-prints*, November 2015.

[18] R. G. Krishnan, U. Shalit, and D. Sontag. Deep Kalman filters, November 2015.

[19] R. G. Krishnan, U. Shalit, and D. Sontag. Structured inference networks for nonlinear state space models. In *Thirty-First AAAI Conference on Artificial Intelligence*, February 2017.

[20] M. Johnson, D. K. Duvenaud, A. Wiltschko, R. P. Adams, and S. R. Datta. Composing graphical models with neural networks for structured representations and fast inference. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2946–2954. Curran Associates, Inc., 2016.

[21] M. Karl, M. Soelch, J. Bayer, and P. van der Smagt. Deep variational Bayes filters: Unsupervised learning of state space models from raw data. In *5th International Conference on Learning Representations*, 2017.

[22] M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2746–2754. Curran Associates, Inc., 2015.

[23] E. A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, pages 153–158. IEEE, August 2000.

[24] E. A. Wan and A. T. Nelson. *Dual extended Kalman filter methods*, pages 123–173. John Wiley & Sons, Inc, 2001.

[25] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal. The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, May 1995.

[26] D. Sussillo, R. Jozefowicz, L. F. Abbott, and C. Pandarinath. LFADS - latent factor analysis via dynamical systems, August 2016.

[27] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, May 2014.

[28] D. P. Kingma and M. Welling. Auto-Encoding variational bayes. In *International Conference on Learning Representation*, May 2014.

[29] M. Abadi, A. Agarwal, P. Barham, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[30] J. P. Newman, M.-f. Fong, D. C. Millard, et al. Optogenetic feedback control of neural activity. *eLife*.

[31] A. El Hady. *Closed Loop Neuroscience*. Academic Press, 2016.

[32] Y. Zhao and I. M. Park. Variational latent Gaussian process for recovering single-trial dynamics from population spike trains. *Neural Computation*, 29(5), May 2017.

[33] Y. Zhao, J. Yates, and I. M. Park. Low-dimensional state-space trajectory of choice at the population level in area MT. In *Computational and Systems Neuroscience (COSYNE)*, 2017.

[34] E. M. Izhikevich. *Dynamical systems in neuroscience : the geometry of excitability and bursting*. Computational neuroscience. MIT Press, 2007.

[35] C. Curto, S. Sakata, S. Marguet, V. Itskov, and K. D. Harris. A simple model of cortical dynamics explains variability and state dependence of sensory responses in Urethane-Anesthetized auditory cortex. *The Journal of Neuroscience*, 29(34):10600–10612, August 2009.

[36] J. H. Macke, L. Buesing, J. P. Cunningham, et al. Empirical models of spiking in neural populations. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1350–1358. Curran Associates, Inc., 2011.

[37] W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of Neurophysiology*, 93(2):1074–1089, February 2005.