Structured Variational Inference in Partially Observable Unstable Gaussian Process State Space Models

Silvan Melchior[*]   SILVAN.MELCHIOR@ALUMNI.ETHZ.CH

Sebastian Curi [*]   SEBASTIAN.CURI@INF.ETHZ.CH

Felix Berkenkamp   BEFELIX@INF.ETHZ.CH

Andreas Krause   KRAUSEA@INF.ETHZ.CH

Department of Computer Science, ETH Zurich

Editors: A. Bayen, A. Jadbabaie, G. J. Pappas, P. Parrilo, B. Recht, C. Tomlin, M. Zeilinger

Abstract

We propose a new variational inference algorithm for learning in Gaussian Process State-Space Models (GPSSMs). Our algorithm enables learning of unstable and partially observable systems, where previous algorithms fail. Our main algorithmic contribution is a novel approximate posterior that can be calculated efficiently using a single forward and backward pass along the training trajectories. The forward-backward pass is inspired on Kalman smoothing for linear dynamical systems but generalizes to GPSSMs. Our second contribution is a modification of the conditioning step that effectively lowers the Kalman gain. This modification is crucial to attaining good *test* performance where no measurements are available. Finally, we show experimentally that our learning algorithm performs well in stable and unstable *real* systems with hidden states.

Keywords: Variational Inference; Gausian Processes; Dynamical Systems; System Identification

1. Introduction

We consider the problem of learning a probabilistic model of a non-linear dynamical system from data as a first-step of model-based reinforcement learning (Berkenkamp, 2019; Kamthe and Deisenroth, 2017). High-stake control applications require the model to have great predictive performance *in expectation* as well as a correct *uncertainty quantification* over all the prediction sequence. Although parametric models such as deep neural networks successfully achieve the former (Chua et al., 2018; Archer et al., 2015), they do not provide correct probability estimates (Guo et al., 2017; Malik et al., 2019). Instead, we consider Gaussian Processes-State Space Models (GP-SSMs), which were introduced by Wang et al. (2006). These models meet both requirements at the cost of computationally costlier predictions and involved inference methods (Ialongo et al., 2019, Section 3.4).

State-of-the-Art inference methods on GP-SSMs models use doubly stochastic variational inference (Salimbeni and Deisenroth, 2017) on proposed approximate posteriors that are *easy* to sample. The PR-SSM algorithm, by Doerr et al. (2018), uses an approximate posterior that preserves the predictive temporal correlations of the prior distribution. PR-SSM has great test performance in some tasks but in others it fails to learn the system. Ialongo et al. (2019) address PR-SSM limitations and propose an approximate posterior that conditions on measurements using Kalman Filtering (Kalman, 1960), leading to the VCDT algorithm. Although VCDT gives accurate predictions in

---

[*] equal contribution

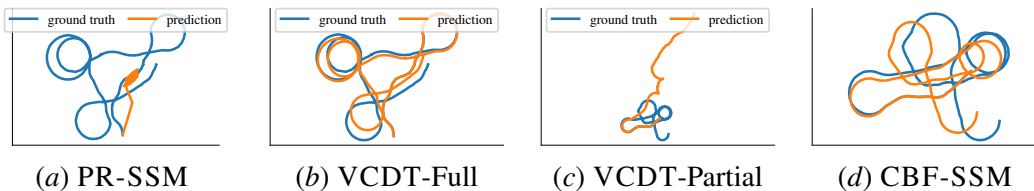| (*a*) PR-SSM | (*b*) VCDT-Full | (*c*) VCDT-Partial | (*d*) CBF-SSM |

Figure 1: Open-loop predictions on test set for a noisy Dubin's car model. In Figures 1(*a*) and 1(*b*) the full state is observed. VCDT learns to predict correctly whereas PR-SSM explains observations with zero mean and high measurement noise. In Figures 1(*b*) and 1(*d*) only partial state information is available. VCDT fails to account for partial observability, it overfits to the training set and the test-predictions diverge. CBF-SSM instead use the smoother pass to infer the hidden states and it has good performance on the training set.

cases where PR-SSM fails, it has worse performance in tasks where PR-SSM successfully learns the system. Furthermore, there are tasks in which both algorithms fail to learn dynamical systems.

This paper builds on the observation that PR-SSM cannot learn systems that are not mean square stable (MSS) as the mismatch between the true and the approximate posterior can be arbitrarily large (Fig. 1(*a*)). Informally, a system is not MSS when the state uncertainty increases with time. If the state is fully observed, VCDT learns (Fig. 1(*b*)) as the conditioning step controls the uncertainty in the posterior. However, when there are hidden states, VCDT also fails (Fig. 1(*c*)). To address this issue, we introduce a backward smoother that is similar in spirit to the Kalman smoother. We then condition using the smoothed estimates, instead of conditioning on the raw observations. Our algorithm, Conditional Backward-Forward State Space Model (CBF-SSM), succeeds in these tasks (Fig. 1(*d*)) and reduces to VCDT when full state information is available. The second improvement of our algorithm is that we reduce the Kalman gain in the conditioning step. This is crucial to achieve good *test* predictive performance, where no measurements are available. We parametrize the conditioning level with a single parameter k that explicitly interpolates between the full conditioning (as in VCDT) and no conditioning (as in PR-SSM) to achieve good performance in both MSS and not MSS tasks.

1.1. Related Work

Variational Inference in GP-SSMs   Frigola et al. (2014) introduce variational inference in GP-SSMs using a mean-field approximation over the sequence of states. To incorporate input-output measurements, Mattos et al. (2015) introduce a recognition module that learns the initial state distribution. Eleftheriadis et al. (2017) overcome the mean-field approximation and propose a posterior that preserves the prior temporal correlations for *linear* systems, while Doerr et al. (2018) present a posterior that preserves the prior temporal correlations for *non-linear* systems. Finally, Ialongo et al. (2019) approximate the posterior temporal correlation by conditioning the prior on a single observation (i.e., filtering). We build upon these works and introduce a backward smoother used for conditioning that approximates the true posterior temporal correlations better than previous work.

Variational Inference on Parametric State Space Models   Archer et al. (2015) introduce stochastic variational inference on parametric state-space models using a Gaussian distribution with a structured covariance matrix to obtain a tractable posterior. Krishnan et al. (2017) build on this work relaxing the structure of the covariance matrix and introducing a deterministic smoothing pass. Our

backward pass is similar in spirit, but we consider probabilistic smoothed observations instead of deterministic ones to account for uncertainty in the backward pass explicitly. Godsill et al. (2004) propose a Monte Carlo particle-based approximation to the smoothing step. We instead use a doubly stochastic variational approximation during the backward simulation step.

2. Problem Statement and Background

We consider the problem of model-learning: At test time, we are given a sequence of control actions $u_{1:T}$ together with initial observations $y_{1:t'}$ and we must predict future observations $y_{t':T}$. We need an initial sequence t' of observations as the initial state is hidden, i.e., t' is the system lag (Markovsky and Rapisarda, 2008). During training, we have access to a training data set that consists of sequences of actions and corresponding observations. We evaluate the quality of our model by evaluating the log-likelihood of the true observations and the RMSE of the mean predictions.

Gaussian Process A Gaussian Process (GP) is a distribution over functions $f \colon \mathbb{R}^{d_x} \to \mathbb{R}$ that is parametrized by a mean function $m(\cdot)$ and covariance function $k(\cdot, \cdot)$, which respectively encode the expected value and similarities in the input space. Given a prior $f \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$ and observations (x, f_x), the posterior distribution of f is also a GP with mean and covariance at x'

$$\mu(x') = m(x') + k_{x'} K_{x,x}^{-1}(f_x - m_x), \quad \Sigma(x', x') = k(x', x) - k_{x'}^{\top} K_{x,x}^{-1} k_{x'}, \tag{1}$$

where $m_x = \{m(x_1), \ldots, m(x_n)\}$, $f_x = f(x)$, $[k_{x'}]_{i,j} = k(x'_i, x_j)$ and $[K_{x,x}]_{i,j} = k(x_i, x_j)$.

Gaussian Process State-Space Model We model the process that generates observations with a SSM. The Markovian latent state $x \in \mathbb{R}^{d_x}$ evolves over time based on a transition function f. The key aspect of these models is that we place a GP prior on these functions. At every time step t, we obtain measurements $y_t \in \mathbb{R}^{d_y}$ of the state x_t. The state transitions and observations are corrupted by zero-mean Gaussian noise with covariance matrices Σ_x and Σ_y, respectively. The GP-SSM is

$$f \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot)), \ \ x_1 \sim \mathcal{N}(\mu_1, \Sigma_1), \ \ x_{t+1}|f_t, x_t \sim \mathcal{N}(f(x_t), \Sigma_x), \ \ y_t \sim \mathcal{N}(Cx_t, \Sigma_y). \tag{2}$$

For multi-dimensional transition functions f with $d_x > 1$, we use independent GPs for each dimension to reduce computational complexity, although our method is not limited to this choice. Furthermore, we restrict C to be known, linear, and full row-rank. For example, $C = \begin{bmatrix} I & 0 \end{bmatrix}$ fullfils such condition. Non-linear observation functions may lead to non-identifiability and may be modeled with a richer class of functions f (Frigola et al., 2014). We also restrict Σ_x and Σ_y to be diagonal to capture the correlations between the states components only through f. For brevity, we omit control inputs. However, all derivations extend to controlled systems (Doerr et al., 2018), and the systems in the experiments have controls.

Sparse GP Approximation The memory needed to compute the approximate posterior of a GP for N observations scales as $\mathcal{O}(N^2)$ and the computational complexity as $\mathcal{O}(N^3)$. These requirements make GPs intractable for large-scale problems. Furthermore, the GP model (1) assumes that the inputs are deterministic, whereas the inputs to the GP in model (2) are probabilistic. To address both issues we use sparse GPs (Titsias, 2009; Hensman et al., 2013). In such models, the GP specifies function values u_f at M input locations z_f such that $p(u_f) = \mathcal{N}(\mu^{u_f}, \Sigma^{u_f})$. The function value at a location x' different to z_f follows a distribution given by $f(x') \sim \int p(f(x')|u_f)p(u_f)\mathrm{d}u_f$, where $p(f(x')|u_f)$ is the posterior of f at location x' given pseudo-observations (z_f, u_f) (see

Eq. (1)). Hence, $f(x')$ is Gaussian and can be computed in closed form. When $M \ll N$, this brings a large computational advantage and does not require the true inputs x to be deterministic. The sparse GP-SSM prior and posterior distribution are

$$p(u_f, x_{1:T}, y_{1:T}) = p(x_1)p(u_f) \prod_{t=1}^{T-1} p(x_{t+1} \mid f_t, x_t)p(f_t \mid u_f) \prod_{t=1}^{T} p(y_t \mid x_t), \qquad (3a)$$

$$p(u_f, x_{1:T} \mid y_{1:T}) = p(x_1 \mid y_{1:T})p(u_f \mid y_{1:T}) \prod_{t=1}^{T-1} p(x_{t+1} \mid x_t, f_t, y_{t+1:T})p(f_t \mid u_f, y_{1:T}). \quad (3b)$$

Prediction with GPSSMs The model (2) specifies a mechanism to generate samples from the GPSSM. For the trajectory to be consistent, the function sampled along the trajectory has to be unique. To ensure this for a trajectory of length T, we need to condition on all the previous observations yielding a computational complexity of $O(T^3)$. Doerr et al. (2018) omit the consistency requirement and uses independent samples of f for each time-step prediction by assuming that $\int p(u_f) \prod_{t=2}^{T} p(f_{t-1} \mid u_f) \mathrm{d}u_f = \prod_{t=2}^{T} \int p(u_f)p(f_{t-1} \mid u_f) \mathrm{d}u_f$, i.e., each transition is independent of each other. Ialongo et al. (2019) criticizes this assumption and instead proposes to sample $u_f \sim p(u_f)$ at the beginning of each trajectory and approximate the integral by using a Monte Carlo approximation. McHutchon et al. (2015) also addresses the cubic sampling by using just the mean of $p(u_f)$ in each trajectory. Another possibility is to degenerate $p(u_f)$ to a delta distribution in which all methods coincide but essentially reduces the model to a parametric one.

Learning in GPSSMs The posterior distribution (3b) is intractable to compute when the transitions are non-linear. Traditional methods such as MCMC (Frigola et al., 2013) do not scale to large datasets. Variational inference methods (Blei et al., 2017) propose an approximate posterior $q(u_f, x_{1:T}, y_{1:T})$ that is easy to sample and minimize the KL divergence between the approximate and the true posterior. This procedure turns out to be equivalent to maximizing the evidence lower bound (ELBO). The approximate posterior of PR-SSM and VCDT are

$$q_{\text{PR-SSM}}(u_f, x_{1:T}, y_{1:T}) = q(x_1|y_{1:t'})q(u_f) \prod_{t=1}^{T-1} p(x_{t+1}|x_t, f_t)p(f_t|u_f), \qquad (4a)$$

$$q_{\text{VCDT}}(u_f, x_{1:T}, y_{1:T}) = q(x_1|y_{1:t'})q(u_f) \prod_{t=1}^{T-1} q(x_{t+1}|x_t, f_t, y_{t+1})p(f_t|u_f), \qquad (4b)$$

where $q(x_1 \mid y_{1:t'}) = \mathcal{N}(\mu^{qx_1}, \Sigma^{qx_1})$ is called the recognition module and $q(u_f) = \mathcal{N}(\mu^{qu_f}, \Sigma^{qu_f})$ is the sparse GP posterior. Both algorithms use the prior $p(f_t|u_f)$ to generate the function samples which simplifies the KL divergence between the function prior and posterior to the KL divergence between $q(u_f)$ and $p(u_f)$ only (Matthews, 2017). The crucial difference between both algorithms is on how they compute the next-state approximate posterior. Whereas PR-SSM uses the prior, VCDT uses a 1-step approximation to the posterior (c.f. Equations (3a) and (3b)). The 1-step VCDT posterior approximation is also a Gaussian that can be efficiently computed using a Kalman-filtering conditioning rule. The ELBO of PR-SSM and VCDT are

$$\mathcal{L}_{\text{PR-SSM}} = \sum_{t=1}^{T} \mathbb{E}_q \left[\log p(y_t|x_t)\right] - \text{KL}(q(u_f) \mid\mid p(u_f)) - \text{KL}(q(x_1|y_{1:t'}) \mid\mid p(x_1)), \quad (5a)$$

$$\mathcal{L}_{\text{VCDT}} = \mathcal{L}_{\text{PR-SSM}} - \sum_{t=1}^{T-1} \text{KL}(q(x_{t+1}|x_t, f_t, y_{t+1}) \mid\mid p(x_{t+1} \mid f_t, x_t)). \qquad (5b)$$

The first term of the ELBO (5a) maximizes the observations conditional likelihood, whereas the first KL divergence term regularizes the inducing points of the GPs and the recognition module. It is common to select $p(x_1)$ as an uninformative prior, so this KL divergence vanishes. The ELBO of VCDT (5b) also regularizes the conditioning step through the KL divergence.

3. Variational Inference in Unstable GP-SSMs

Mean-Square Unstable Systems A system that is mean-square stable (MSS) has a bounded predictive state covariance matrix $\lim_{t \to \infty} \mathbb{E}\left[x_t x_t^\top \mid x_1\right]$ (Soong, 1973; Khasminskii, 2012). Conversely, systems that are not MSS have an unbounded predictive state covariance matrix. A linear system with a spectral radius larger or equal to one, combined with non-zero additive noise, is not MSS. As an illustrative example, we use Dubin's car model as a not MSS system, where the state is the (x, y) position and the orientation, and the controls are the speed and curvature commands.

Learning with PR-SSM on not MSS systems over long-time horizons is challenging because the state-transition term in the approximate posterior (4) does not condition on the observations as the true posterior (3b) does. In such models, the approximate posterior variance increases along the trajectory, whereas the true posterior variance is constant. When optimizing the ELBO (5a), the model assigns high observation noise Σ_y to explain the measurements instead of learning f.

When the sequence is *short*-enough, PR-SSM does not suffer this shortcoming during training, but the test performance on long sequences is poor. VCDT addresses this by using an approximate posterior that conditions on the measurements. Nevertheless, it learns to condition *too much* on the observations, which are not present during testing leading to poor performance. Furthermore, when the system has *unobserved states*, the conditioning step only corrects the measured components of the state. In contrast, the unmeasured ones are given by the prior distribution as in PR-SSM. The Conditional Backward-Forward State-Space Model (CBF-SSM) algorithm explicitly estimates the hidden states and learns even with partial state observation and in unstable systems.

3.1. Conditional Backward-Forward State-Space Model

Ideally, we would like to propose an approximate posterior that uses the full $y_{t:T}$ in the conditional state transition term, yet it is tractable to compute. We propose a backward pass to smooth the measurements $y_{t:T}$ into a distribution over a single pseudo-state $\tilde{x}_t \in \mathbb{R}^{d_x}$ that approximates $p(\tilde{x}_t | y_{t:T})$.

Backward Pass Traditional smoothing algorithms compute the posterior by propagating $p(x_t \mid x_{t+1}, y_t)$ in a backward pass. However, when the forward model has a Gaussian Process prior, the backward probabilities are intractable. Instead, we propose an auxiliary noiseless model that runs from $t = T$ to $t = 1$ that produces the same observations y_t, as shown in Fig. 2. This model has states $\tilde{x}_t \in \mathbb{R}^{d_x}$ and is generated as
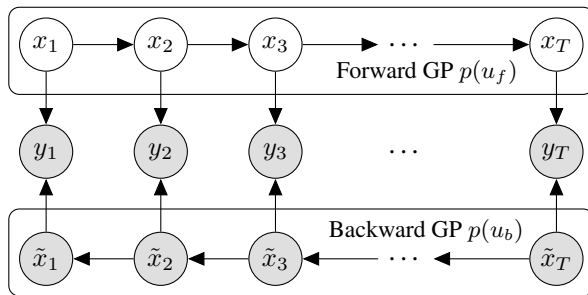

Figure 2: Backward-Forward GP-SSM Model

$$f_b \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot)), \quad \tilde{x}_T \sim \mathcal{N}(\mu_1, \Sigma_1), \quad \tilde{x}_t = f_b(\tilde{x}_{t+1}), \quad y_t = C\tilde{x}_t. \tag{6}$$

Using a sparse GP approximation for the backward pass, the CBF-SSM approximate posterior is:

$$q_{\text{CBF-SSM}}(u_f, x_{1:T}, y_{1:T}) = q(x_1 \mid y_{1:t'})q(u_f) \prod_{t=1}^{T-1} q(x_{t+1} \mid x_t, f_t, \tilde{x}_{t+1})p(f_t \mid u_f)$$

$$\cdot \, q(\tilde{x}_T \mid y_T)q(u_b) \prod_{t=1}^{T-1} p(\tilde{x}_t \mid \tilde{x}_{t+1}, f_{t+1}, y_t)p(f_t \mid u_b). \tag{7}$$

5

The second line of Eq. (7) is computed with a single backward pass and the first line with a single forward pass, conditioning on $\tilde{x}_t$ at every time step. The first d_y components of $\tilde{x}_t$ are $C^\top (CC^\top)^{-1} y_t$ and the rest are predicted with the backward GP. Due to the full row-rank restriction on C, the inverse exists. When the state is fully observed, the second line of Eq. (7) reduces to a dirac distribution at $\tilde{x}_t = y_t$ and CBF-SSM and VCDT algorithms coincide. This forward-backward algorithm is similar in spirit to the smoother from Krishnan et al. (2017), but our models are probabilistic to approximate the true posterior. The ELBO of CBF-SSM is

$$\mathcal{L}_{\text{CBF-SSM}} = \mathcal{L}_{\text{PR-SSM}} - \sum_{t=1}^{T-1} \text{KL}(q(x_{t+1}|x_t, f_t, \tilde{x}_{t+1}) \ || \ p(x_{t+1} \mid f_t, x_t)) - \text{KL}(q(u_b) \ || \ p(u_b)).$$

Soft Conditioning Step   The conditioning step of VCDT for full state observations can be summarized as follows. As both $q(x_t \mid f_{t-1}, x_{t-1}) \equiv \mathcal{N}(\mu_t^-, \Sigma_t^-)$ and $p(\tilde{x}_t \mid x_t) \equiv \mathcal{N}(x_t, \tilde{\Sigma}_x)$ are Gaussian distributions, the approximate posterior $q(x_t \mid f_{t-1}, x_{t-1}, \tilde{x}_t) = \mathcal{N}(\mu_t, \Sigma_t)$ with

$$\mu_t = \mu_t^- + K(\tilde{x}_t - \mu_t^{-1}), \quad \Sigma_t = (I - K)\Sigma_t^-(I - K)^\top + K\tilde{\Sigma}_x K^\top, \tag{8}$$

where K is the Kalman gain $K = \Sigma_t^-(\tilde{\Sigma}_x + \Sigma_t^-)^{-1}$. Note that we condition using the simulated state $\tilde{x}_t$ instead of the measurement y_t. Our second contribution is a soft conditioning step. We propose to use a free factor $k \geq 1$ such that the Kalman gain is $K_{\text{soft}} = \Sigma_t^-(\tilde{\Sigma}_x + k\Sigma_t^-)^{-1}$ and the conditioning step is still given by Eq. (8). When $k = 1$, this reduces to the VCDT conditioning step and, when $k \to \infty$ then $K_{\text{soft}} \to 0$, and CBF-SSM does not condition, as in PR-SSM. The soft-conditioning parameter k trades off one-step and long-term accuracy. This soft-conditioning step is a particular case of the most general posterior proposed by Ialongo et al. (2019). However, their function class is time-varying and much larger than our restricted soft-conditioning step. Hence, VCDT tends to overfit and produce poor test results, as we found in experiments.

Tuning Hyper-Parameters   In standard stochastic variational inference (Hoffman et al., 2013), the KL-divergence terms are re-weighted by factors to account for batch-size relative to the full dataset. In our setting, the i.i.d. assumption of the dataset is violated, and this leads to sub-optimal results in all three algorithms. We introduce a scaling parameter β to reweigh the KL-divergence terms in the ELBO. This re-weighting scheme is based on the β-VAE algorithm by Higgins et al. (2017). Furthermore, we notice that when sampling independent functions along a trajectory as in PR-SSM, the KL divergence of the inducing points has to be scaled by the trajectory length.

4. Experiments

System Identification Benchmarks   We compare CBF-SSM against PR-SSM and VCDT on the datasets used by Doerr et al. (2018), where PR-SSM outperforms other methods. Table 1 shows CBF-SSM-1 without soft conditioning, CBF-SSM-50 with a soft conditioning factor of $k = 50$ and CBF-SSM-1S without soft conditioning but with the function sampling method proposed by Ialongo et al. (2019). We first remark that our implementation of PR-SSM has better performance than the original paper, and this is because we correctly compute the KL divergence between the inducing points when the functions are sampled independently along a trajectory. The second observation is that VCDT performs considerably worse than PR-SSM in these tasks. If we compare VCDT to CBF-SSM-1 (both methods coincide except for the function sampling method and the backward pass), we see that CBF-SSM-1 outperforms VCDT. If we compare VCDT to CBF-SSM-1S (both methods coincide except for the backward step), we see that the methods perform

(*a*) Dubin's car model.

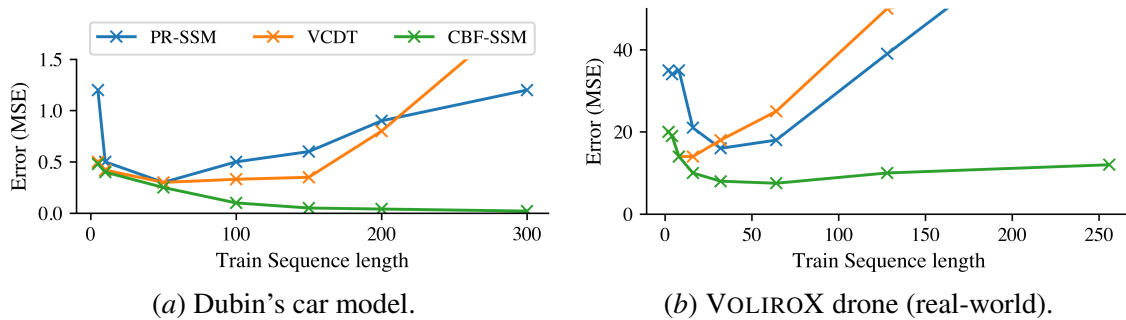(*b*) VOLIROX drone (real-world).

Figure 3: Effect of training sequence length on test error for MSU systems. PR-SSM can only train on short trajectories. CBF-SSM achieves lower error by training on longer sequences.

relatively similarly. This suggests that the function sampling method proposed by Ialongo et al. (2019) is too noisy to be useful for learning. Finally, if we compare CBF-SSM-1 to CBF-SSM-50, we see that the performance is comparable, except for the large-scale Sarcos data set where soft conditioning is crucial to attaining good performance. In summary, we see that CBF-SSM-50 outperforms or is comparable to all other methods in all data sets.

Dataset	PR-SSM	VCDT	CBF-SSM-1	CBF-SSM-50	CBF-SSM-1S
ACTUATOR	**0.446 (0.017)**	1.060 (0.490)	**0.452 (0.014)**	**0.452 (0.013)**	0.985 (0.360)
BALLBEAM	**0.045 (0.003)**	0.080 (0.010)	**0.050 (0.004)**	**0.050 (0.004)**	0.083 (0.007)
DRIVES	**0.332 (0.059)**	0.757 (0.040)	**0.340 (0.073)**	**0.335 (0.068)**	0.752 (0.019)
DRYER	**0.083 (0.005)**	0.665 (0.230)	**0.080 (0.006)**	**0.082 (0.004)**	0.650 (0.380)
FURNACE	**1.260 (0.013)**	3.020 (1.200)	**1.210 (0.018)**	**1.220 (0.010)**	2.500 (0.640)
FLUTTER	**0.264 (0.012)**	0.424 (0.060)	**0.273 (0.017)**	**0.269 (0.020)**	0.714 (0.410)
TANK	0.178 (0.007)	0.846 (0.370)	**0.168 (0.013)**	**0.172 (0.014)**	1.140 (0.340)
SARCOS	**0.046 (0.002)**	0.170 (0.013)	0.159 (0.012)	**0.049 (0.002)**	0.265 (0.100)

Table 1: Test RMSE [mean (std)] over five runs for different datasets. In bold typeface we indicate the best performing algorithms. CBF-SSM-1 and CBF-SSM-50 differ on the conditioning ($k = 1$ or $k = 50$). CBF-SSM-1S ($k = 1$) uses the VCDT function sampling step. CBF-SSM-50 achieves lowest error in all data sets.

Simulated unstable system We evaluate the performance on the toy car dataset introduced in Section 3. Fig. 1 shows a qualitative comparison of the variational inference algorithms when trained on sequence lengths of 300 and the resulting test error for different sequence lengths is shown in Fig. 3(*a*). CBF-SSM achieves lower test error when training on longer sequences, while PR-SSM fails to learn the system accurately on long sequences.

VoliroX To demonstrate that CBF-SSM can be applied to real-world, complex, and unstable systems, we use it to learn the dynamics of a flying robotic vehicle. VoliroX (Bodie et al., 2018) is a drone consisting of twelve rotors mounted on six tiltable arms, which cause airflow interference that is difficult to model. The dataset includes measured position and orientation $p \in \mathbb{R}^6$, while linear and angular velocities $v \in \mathbb{R}^6$ are unobserved. Control inputs are the arm tilt angles $\alpha \in \mathbb{R}^6$ and

7

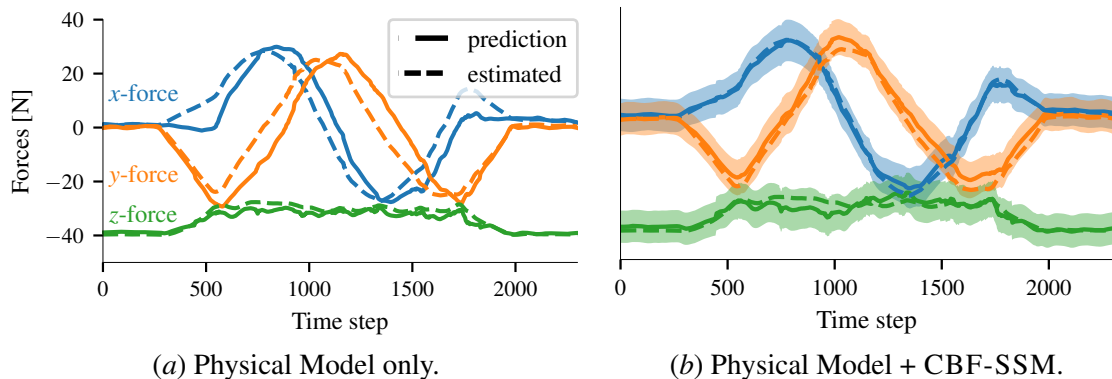(a) Physical Model only.   (b) Physical Model + CBF-SSM.

Figure 4: Test-set predictions on Voliro-X. In Fig. 4(a) we show the forces predicted by the physical model and the forces estimated from data. In Fig. 4(b) we plot the predictions by CBF-SSM. The shaded regions are ± 1.96 the predicted std. deviation.

motor signals $\eta \in \mathbb{R}^6$. Bodie et al. (2018) model the rigid body (RB) dynamics with an integrated ordinary differential equation (ODE), $(p_{t+1}, v_{t+1}) = f_{\text{RB-ODE}}(p_t, v_t, \xi_t, \tau_t)$, which depends on the forces ξ_t and torques τ_t acting on the system. While Bodie et al. (2018) predict forces and torques with a physical model, f_{PM}, we additionally learn a GP correction term to account for modeling errors, $(\xi_t, \tau_t) = f_{\text{PM}}(\eta_t, \alpha_t) + f_{\text{GP}}(\eta_t, \alpha_t)$. We integrate the resulting ODE in a differentiable way using TensorFlow (Abadi et al., 2015) and estimate the velocities $\mathbf{v}$ with our backward model. Although the system is high-dimensional, we use the GP only to model external forces and torques, $\mathbb{R}^{12} \rightarrow \mathbb{R}^6$. Since we combine this prediction with the rigid body dynamics, we can effectively exploit prior physics knowledge and avoids learning about basic physics facts.

The physical model does not model airflow interference, which leads to significant prediction errors in Fig. 4(a). In contrast, CBF-SSM provides accurate predictions with reliable uncertainty information in Fig. 4(b). We compare these predictions to PR-SSM and VCDT for different training sequence lengths in Fig. 3(b). Since the drone is unstable and has large process noise, PR-SSM and VCDT can only train on short sequences. In contrast, CBF-SSM can reliably train on longer sequence lengths and hence achieve lower predictive errors without overfitting.

Computational Performance   The prediction time of all algorithms is identical as all use the model (2). As a function of T, all algorithms take $\mathcal{O}(T)$ to compute the forward and the backward pass. However, the extra backward pass in our algorithm makes training $3.7\times$ slower.

5. Conclusions

We presented a new algorithm, CBF-SSM, to learn on GPSSMs using Variational Inference. Compared to previous work, our algorithm learns in both MSS and MSU systems with hidden states and achieves superior performance to all other algorithms. We present two algorithmic innovations in CBF-SSM: the backward pass that provides a better approximation to the true posterior and the soft conditioning that trades-off training and testing accuracy. Finally, we demonstrate the capabilities of CBF-SSM in small and large-scale benchmarks and simulated and real robots.

Acknowledgments

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Evan Archer, Il Memming Park, Lars Buesing, John Cunningham, and Liam Paninski. Black box variational inference for state space models. *arXiv preprint arXiv:1511.07367*, 2015.

Felix Berkenkamp. *Safe Exploration in Reinforcement Learning: Theory and Applications in Robotics*. PhD thesis, ETH Zurich, 2019.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Karen Bodie, Zachary Taylor, Mina Kamel, and Roland Siegwart. Towards efficient full pose omnidirectionality with overactuated mavs. *CoRR*, abs/1810.06258, 2018.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, pages 4754–4765, 2018.

Andreas Doerr, Christian Daniel, Martin Schiegg, Nguyen-Tuong Duy, Stefan Schaal, Marc Toussaint, and Trimpe Sebastian. Probabilistic recurrent state-space models. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1280–1289, 2018.

Stefanos Eleftheriadis, Tom Nicholson, Marc Deisenroth, and James Hensman. Identification of gaussian process state space models. In *Advances in Neural Information Processing Systems*, pages 5309–5319, 2017.

Roger Frigola, Fredrik Lindsten, Thomas B Schön, and Carl Edward Rasmussen. Bayesian inference and learning in gaussian process state-space models with particle mcmc. In *Advances in Neural Information Processing Systems*, pages 3156–3164, 2013.

Roger Frigola, Yutian Chen, and Carl Edward Rasmussen. Variational gaussian process state-space models. In *Advances in Neural Information Processing Systems*, pages 3680–3688, 2014.

Simon J Godsill, Arnaud Doucet, and Mike West. Monte carlo smoothing for nonlinear time series. *Journal of the american statistical association*, 99(465):156–168, 2004.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. In *Uncertainty in Artificial Intelligence*, page 282. Citeseer, 2013.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6, 2017.

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

Alessandro Davide Ialongo, Mark Van Der Wilk, James Hensman, and Carl Edward Rasmussen. Overcoming mean-field approximations in recurrent gaussian process models. In *International Conference on Machine Learning*, pages 2931–2940, 2019.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

Sanket Kamthe and Marc Peter Deisenroth. Data-efficient reinforcement learning with probabilistic model predictive control. *arXiv preprint arXiv:1706.06491*, 2017.

Rafail Khasminskii. *Stochastic Stability of Differential Equations*. Stochastic Modelling and Applied Probability. Springer-Verlag, Berlin Heidelberg, 2 edition, 2012.

Rahul G Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Ali Malik, Volodymyr Kuleshov, Jiaming Song, Danny Nemer, Harlan Seymour, and Stefano Ermon. Calibrated model-based deep reinforcement learning. In *International Conference on Machine Learning*, pages 4314–4323, 2019.

Ivan Markovsky and Paolo Rapisarda. Data-driven simulation and control. *International Journal of Control*, 81(12):1946–1959, 2008.

Alexander Graeme de Garis Matthews. *Scalable Gaussian process inference using variational methods*. PhD thesis, University of Cambridge, 2017.

César Lincoln C Mattos, Zhenwen Dai, Andreas Damianou, Jeremy Forth, Guilherme A Barreto, and Neil D Lawrence. Recurrent gaussian processes. *arXiv preprint arXiv:1511.06644*, 2015.

Andrew James McHutchon et al. *Nonlinear modelling and control using Gaussian processes*. PhD thesis, Citeseer, 2015.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep gaussian processes. In *Advances in Neural Information Processing Systems*, pages 4588–4599, 2017.

Tsu T Soong. *Random differential equations in science and engineering*. Elsevier, 1973.

Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

Jack Wang, Aaron Hertzmann, and David J Fleet. Gaussian process dynamical models. In *Advances in neural information processing systems*, pages 1441–1448, 2006.