Computation-Aware State-Space Models for Neural Data

Jonathan Huml[1], Jonathan Wenger[2], John P. Cunningham[1, 2]
(1) Department of Statistics, Columbia University (2) Zuckerman Institute, Columbia University

Summary. Dynamical latent variable models of neural activity fall into two general classes: (1) classical, difficult-to-scale Bayesian methods like Gaussian Process Factor Analysis (GPFA) that exploit prior knowledge and model uncertainty; (2) scalable, deep algorithms like Latent Factor Analysis via Dynamical Systems (LFADS) without explicit priors that may struggle in the low-trial regime and do not model uncertainty. Recent efforts to scale the spatial and temporal resolution of (1) rely on posterior approximations, introducing approximation error and silently skewing uncertainty estimates. One manifestation of this leakage is variance starvation or overconfidence. We introduce a scalable probabilistic framework, the Computation Aware State Space Model (CASSM), that preserves the mathematical interpretation of a previous work, the Computation Aware Kalman Filter (CAKF), that quantifies this inevitable approximation error and appropriately increases uncertainty. Our work introduces a novel loss function to learn both a low-dimensional projection and model of neural dynamics. We show that if GPFA is a Factor Analysis model wrapped in a Gaussian Process, then CASSM is essentially a Principal Components Analysis analogue wrapped in a Kalman Filter. Our optimization scheme achieves linear time complexity instead of the cubic complexity (in temporal resolution) of GPFA, while also rigorously quantifying the approximation error introduced by the low-dimensional projection. On a synthetic spiking task, we show that explicitly modeling approximation error improves generalization even in the case where we only prioritize predictive performance.

Background. We infer states $\{\mathbf{x}_t\}_{t=1}^T$ of an unobserved discrete-time Gauss-Markov process from a set of noisy observations $\{\mathbf{y}_t\}_{t=1}^T$. The Kalman filter is an algorithm for computing exact conditional distributions $\mathbf{x}_t|\mathbf{y}_{1:t}$ in $\mathcal{O}(T)$. However, similar to Gaussian process-based methods like GPFA, the algorithm requires linear solves in the number of neurons N, an $\mathcal{O}(N^3)$ operation. While approximations can yield computational tractability, inference then becomes as much about the approximation method as the model itself. Where these approximations conflict in their uncertainty estimates, their scientific interpretation is unclear. By modeling this approximation in our algorithm through the objective function, we can obtain mathematically interpretable estimates for the difference of exact inference (a Kalman filter) and approximate inference (CASSM, an approximate Kalman filter and smoother). That is, we can return an implicit *combined uncertainty* that is the sum of mathematical and computational posterior uncertainty. For Kalman filtering distribution p and approximate filtering distribution q, we can re-state this mathematically as:

$$\mathcal{L}(\Theta) = \underbrace{-\log p(\mathbf{y}_{1:T}|\Theta)}_{\text{neg. log-marginal likelihood}} + \sum_{t\in[T]} \underbrace{D_{KL}\left(q(\mathbf{x}_t|\mathbf{y}_{1:t})||p(\mathbf{x}_t|\mathbf{y}_{1:t})\right)}_{\text{divergence from exact filtering distribution}} \tag{1}$$

where Θ includes the dynamics model, measurement model, and their associated covariance functions. We achieve the approximation by learning a low, D-dimensional projection of the pre-update covariance function. When $D = N$, the divergence term vanishes and CASSM reduces to a Kalman filter with model selection. Each term of Equation (1) is individually $\mathcal{O}(N^3)$, but a core mathematical contribution of our framework is an equivalent formulation that only scales cubically with respect to the number of latent factors D. While the original Gaussian Process has $\mathcal{O}(T^3N^3)$ computational complexity, we instead achieve $\mathcal{O}(TD^3)$ complexity, so the method scales prohibitively only with respect to the intrinsic dimensionality of the data. Below, we show some preliminary results on a small synthetic spiking task.

Model	$N = 30$		$N = 120$	
	$\mathbf{MSE}_{\mathbf{Test}}$ $\times 10^{-2}$	$\mathbf{NLL}_{\mathbf{Test}}$	$\mathbf{MSE}_{\mathbf{Test}}$ $\times 10^{-2}$	$\mathbf{NLL}_{\mathbf{Test}}$
GPFA	1.06 ± 0.48	-0.99 ± 0.83	2.31 ± 1.07	0.26 ± 0.78
LFADS	0.82 ± 0.26	–	1.07 ± 0.62	–
CASSM (ours)	0.67 ± 0.14	-2.05 ± 0.11	0.63 ± 0.10	-2.15 ± 0.06
$\text{CASSM}_{D=N}$	0.34 ± 0.03	-2.34 ± 0.02	0.15 ± 0.01	-2.51 ± 0.02

Figure 1: Mean-squared error (MSE) and negative log-likelihood (NLL) for latent firing rate predictions on held-out Lorenz test data for $N = 30$ and 120 neurons, with standard deviations from 5 different seeds.

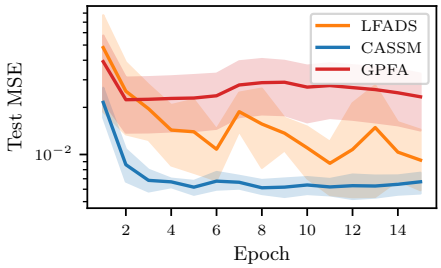

Figure 2: Learning curves