System Identification in Gaussian Process Dynamical Systems

Ryan Turner University of Cambridge
Marc Peter Deisenroth University of Cambridge
Carl Edward Rasmussen University of Cambridge

Inference and learning in linear dynamical systems have long been studied in signal processing, machine learning, and system theory for tracking, localization, and control. In the linear Gaussian case, closed form solutions to inference and learning, also called system identification, are known as the Kalman Filter. For nonlinear dynamical systems (NLDSs), inference and system identification typically require approximations, such as the Extended Kalman Filter (EKF).

We consider the case of the NLDS being given by the state-space formulation

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \boldsymbol{\epsilon}_t \in \mathbb{R}^M\,, \quad \mathbf{y}_t = g(\mathbf{x}_t) + \boldsymbol{\nu}_t \in \mathbb{R}^D\,, \tag{1}$$

where the latent state $\mathbf{x}$ evolves according to a Markovian process. At each time instance t, we obtain a measurement $\mathbf{y}_t$ which depends on the latent state $\mathbf{x}_t$. The terms $\boldsymbol{\epsilon}$ and $\boldsymbol{\nu}$ denote Gaussian system noise and Gaussian measurement noise, respectively.

Assume for a moment that the transition function $f\colon \mathbb{R}^M \to \mathbb{R}^M$ and the measurement function $g\colon \mathbb{R}^M \to \mathbb{R}^D$ in Eq. (1) are known and a sequence $\mathbf{Y} = \mathbf{y}_1, \ldots, \mathbf{y}_T$ of measurements has been obtained. Then, *inference* aims to determine a posterior distribution over the latent state sequence $\mathbf{X} \coloneqq \mathbf{x}_{1:T}$. The requirement for inference in latent space is that the transition function f and the measurement function g are known.

The contribution of this paper is the GPIL algorithm for system identification in nonlinear dynamic systems for the special case where f and g are described by Gaussian processes (GPs). We *learn* GP models for both the transition function f and measurement function g without the necessity of ground truth observations of the latent states.

General Setup

To train a GP, training inputs and training targets are required. Here, training inputs in latent space are not available since the latent states are not observed. Therefore, to learn the GPs $\mathcal{GP}_f$ and $\mathcal{GP}_g$ for f and g, we parameterize them by *pseudo training sets*, which are similar to the pseudo training sets used in sparse GP approximations [3]. The pseudo training set for $\mathcal{GP}_f$ consists of N independent pairs of states $\mathbf{x}_i$ and successor states $f(\mathbf{x}_i)+\boldsymbol{\epsilon}_i$. The parameters of $\mathcal{GP}_f$ are then given by the kernel hyper-parameters, the pseudo training inputs $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_i \in \mathbb{R}^M\}_{i=1}^N$ and the pseudo training targets $\boldsymbol{\beta} = \{\boldsymbol{\beta}_i \in \mathbb{R}^M\}_{i=1}^N$. $\mathcal{GP}_g$ is parameterized by kernel hyper-parameters, pseudo training inputs $\boldsymbol{\xi} = \{\boldsymbol{\xi}_i \in \mathbb{R}^M\}_{i=1}^N$ in latent space and pseudo training targets $\boldsymbol{\upsilon} = \{\boldsymbol{\upsilon}_i \in \mathbb{R}^D\}_{i=1}^N$ in observed space. The role of these pseudo data sets, is simply to provide a flexible parameterization of distributions over nonlinear functions, f and g. Note that the pseudo training sets are *not* given by a time series $\mathbf{x}_{1:T}$ and the corresponding measurements $\mathbf{y}_{1:T}$. Given the GP "parameters", we predict according to

$$x_{ti} = f_i(\mathbf{x}_{t-1}) + \epsilon_{ti} \sim \mathcal{GP}_f(\mathbf{x}_{t-1}|\boldsymbol{\alpha}, \boldsymbol{\beta}_i), \quad y_{tj} = g_j(\mathbf{y}_t) + \nu_{tj} \sim \mathcal{GP}_g(\mathbf{x}_t|\boldsymbol{\xi}, \boldsymbol{\upsilon}_j)\,,$$

where x_{ti} is the i^{th} dimension of $\mathbf{x}_t$ and y_{tj} is the j^{th} dimension of $\mathbf{y}_t$.

System identification determines appropriate "parameters" for $\mathcal{GP}_f$ and $\mathcal{GP}_g$, such that the time series $\mathbf{y}_{1:T}$ can be explained. We use the Expectation Maximization (EM) algorithm to determine the "parameters" of both GP models. EM iterates between two steps. In the E-step (inference step), we determine a posterior distribution $p(\mathbf{X}|\mathbf{Y}, \Theta)$ on the hidden states for a *fixed* parameter setting Θ. In the M-step, we find parameters Θ^* of the GP state-space model that maximize the expected

log-likelihood $Q = \mathbb{E}_{\mathbf{X}}\left[\log p(\mathbf{X}, \mathbf{Y}|\Theta)\right]$, where the expectation is taken with respect to $p(\mathbf{X}|\mathbf{Y}, \Theta)$, the E-step distribution.

System Identification with EM

For analytic inference (E-step), we extend the filtering algorithm of [1] to smoothing in GP state-space models. The algorithm is based on approximate moment matching but we do not give the details here. In the M-Step, we seek the parameters Θ that maximize the likelihood lower bound $Q = \mathbb{E}_{\mathbf{X}}\left[\log p(\mathbf{X}, \mathbf{Y}|\Theta)\right]$ where the expectation is computed under the distribution from the E-Step, meaning $\mathbf{X}$ is treated as the random variable. We decompose Q into

$$Q = \mathbb{E}_{\mathbf{X}}\left[\log p(\mathbf{X}, \mathbf{Y}|\Theta)\right] = \mathbb{E}_{\mathbf{X}}\left[\log p(\mathbf{x}_1|\Theta) + \sum_{t=2}^{T} \underbrace{\log p(\mathbf{x}_t|\mathbf{x}_{t-1}, \Theta)}_{\text{Transition}} + \sum_{t=1}^{T} \underbrace{\log p(\mathbf{y}_t|\mathbf{x}_t, \Theta)}_{\text{Measurement}}\right]. \quad (2)$$

In the following we use the notation $\mu_i(\mathbf{x}) = \mathbb{E}_{f_i}[f_i(\mathbf{x})]$ to refer to the expected value of the i^{th} dimension of f when evaluated at $\mathbf{x}$. Likewise, $\sigma_i^2(\mathbf{x}) = \text{Var}_{f_i}[f_i(\mathbf{x})]$ refers to the variance of the output of i^{th} dimension of f when evaluated at $\mathbf{x}$.

We focus on finding a lower bound approximation to the contribution from the transition function,

$$\mathbb{E}_{\mathbf{X}}\left[\log p(\mathbf{x}_t|\mathbf{x}_{t-1}, \Theta)\right] = -\frac{1}{2}\sum_{i=1}^{M} \underbrace{\mathbb{E}_{\mathbf{X}}\left[\frac{(x_{ti} - \mu_i(\mathbf{x}_{t-1}))^2}{\sigma_i^2(\mathbf{x}_{t-1})}\right]}_{\text{Data Fit Term}} + \underbrace{\mathbb{E}_{\mathbf{X}}\left[\log \sigma_i^2(\mathbf{x}_{t-1})\right]}_{\text{Complexity Term}} + \text{const.} \quad (3)$$

Eq. (3) amounts to an expectation over a nonlinear function of a normally distributed random variable since $\mathbf{X}$ is approximately Gaussian (E-step). Note that in contrast to most other NLDS system identification algorithms the variance of $f(\mathbf{x}_{t-1})$ depends on the location of $\mathbf{x}_{t-1}$.

Data fit. We first consider the data fit term in eq. (3), which is an expectation over the square Mahalanobis distance. For tractability, we approximate the expectation of the ratio

$$\mathbb{E}_{\mathbf{X}}\left[\frac{(x_{ti} - \mu_i(\mathbf{x}_{t-1}))^2}{\sigma_i^2(\mathbf{x}_{t-1})}\right] \approx \frac{\mathbb{E}_{\mathbf{X}}\left[(x_{ti} - \mu_i(\mathbf{x}_{t-1}))^2\right]}{\mathbb{E}_{\mathbf{X}}\left[\sigma_i^2(\mathbf{x}_{t-1})\right]}. \quad (4)$$

Complexity penalty. We next approximate the complexity penalty in eq. (3), which penalizes uncertainty. The contribution from the logarithm can be lower bounded by Jensen's inequality,

$$\mathbb{E}_{\mathbf{X}}\left[\log \sigma_i^2(\mathbf{x}_{t-1})\right] \leq \log \mathbb{E}_{\mathbf{X}}\left[\sigma_i^2(\mathbf{x}_{t-1})\right]. \quad (5)$$

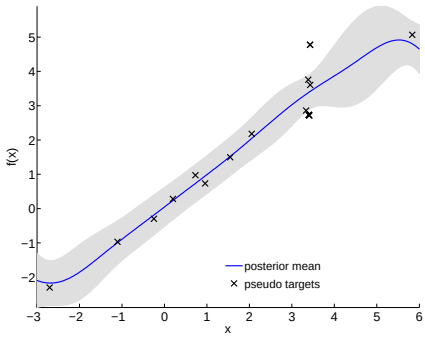
Nearly identical expressions to and eq. (4) and eq. (5) exist for the measurement model.

Results

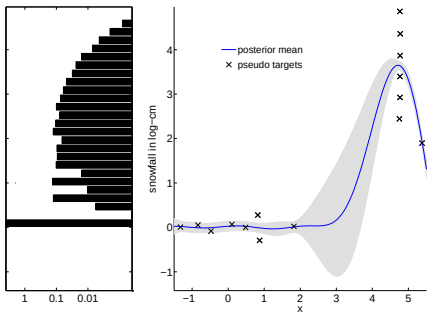
We evaluate our EM based system identification on both real and synthetic data sets using one-step-ahead prediction. We compare GPIL predictions to eight other methods, the time independent model (TIM) with $\mathbf{y}_t \sim \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$, the Kalman filter, the UKF, the EKF, NDFA, GPDM, the Autoregressive GP (ARGP) trained on a set of pairs $(\mathbf{y}_i, \mathbf{y}_{i+1})$, and the GP-UKF [2], which uses the GPIL pseudo training set. Note that the EKF, the UKF, and the GP-UKF require access to the true functions f and g. For synthetic data, f and g are known, for the real data set, we used "true" functions that resemble the mean functions of the GPIL learned GP models.

Real data. We use historical snowfall data in Whistler, BC, Canada to evaluate GPIL on real data. We evaluate the models' ability to predict next day's snowfall using 35 years of test data; we trained on daily snowfall from Jan. 1 1972–Dec. 31 1973 and tested on next day predictions for 1974–2008. The results are shown in table 1. Snowfall time series have many observations of zero, corresponding to days when it does not snow.

The GPIL learns a GP model for a close-to-linear stochastic latent transition function (Fig. 1(a)). A possible interpretation of the results is that the daily precipitation is almost linear. Note that for positive temperatures no snow occurs, which results in a hinge measurement model. The GPIL learns a hinge like function for the measurement model, Fig. 1(b), which allows for predicting a high probability of zero snowfall the next day. The Kalman filter is incapable of such predictions since it assumes linear functions f and g, respectively.



(a) Learned GP model for the stochastic transition function (real data set).



(b) Learned measurement function with log histogram (real data set)

Figure 1: The gray area is twice the predictive standard deviation. The histograms (black) represents the marginal distribution on x_t (left panel) and on y_t (right panel).

Table 1: Comparison of the GPIL with eight other methods on the sinusoidal dynamics example and the Whistler snowfall data. We trained on daily snowfall from Jan. 1 1972–Dec. 31 1973 and tested on next day predictions for 1974–2008. We report the NLL per data point and the RMSE as well as the NLL 95% error bars. We do not report results for GPDM on the real data since it was too slow to run on the large test set.

	Method	NLL synth.	RMSE synth.	NLL real	RMSE real
general	TIM	2.21±0.0091	2.18	1.47±0.0257	1.01
	Kalman	2.07±0.0103	1.91	1.29±0.0273	0.783
	ARGP	1.01±0.0170	0.663	1.25±0.0298	0.793
	NDFA	2.20±0.00515	2.18	14.6±0.374	1.06
	GPDM	3330±386	2.13	N/A	N/A
	GPIL ⋆	$\mathbf{0.917 \pm 0.0185}$	**0.654**	$\mathbf{0.684 \pm 0.0357}$	**0.769**
requires	UKF	4.55±0.133	2.19	1.84±0.0623	0.938
prior	EKF	1.23±0.0306	0.665	1.46±0.0542	0.905
knowledge	GP-UKF	6.15±0.649	2.06	3.03±0.357	0.884

Discussion and Conclusions

We proposed a general method for inference and learning (system identification) in nonlinear stochastic state-space models, where both the transition function and the measurement function are modeled by GPs. The GPs are parameterized by their hyper-parameters and a pseudo training set that are similar to the pseudo training sets in sparse GP approximations. Based on EM, where the inference step can be performed in closed form, we learn the parameters of the transition GP and the measurement GP, respectively.

Note that in our model, the latent states $\mathbf{x}_t$ are *never* observed directly. We solely have access to noisy measurements $\mathbf{y}_t$ to train the latent dynamics and measurement functions. By contrast, [1] and [2] require direct access to ground truth observations of a latent state sequence to train the dynamics model. We showed that our learning approach can successfully learn nonlinear (latent) dynamics based on noisy observations only. Moreover, in our experiments, our algorithm performs better than commonly used approaches for time series predictions.

References

[1] M. P. Deisenroth, M. F. Huber, and U. D. Hanebeck. Analytic Moment-based Gaussian Process Filtering. In L. Bouttou and M. L. Littman, editors, *Proceedings of the 26th International Conference on Machine Learning*, pages 225–232, Montreal, Canada, June 2009. Omnipress.

[2] J. Ko and D. Fox. GP-BayesFilters: Bayesian Filtering using Gaussian Process Prediction and Observation Models. *Autonomous Robots*, 27(1):75–90, July 2009.

[3] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18*, pages 1257–1264. The MIT Press, 2006.