Automatica 80 (2017) 189–199

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

A flexible state–space model for learning nonlinear dynamical systems☆

Andreas Svensson, Thomas B. Schön

Department of Information Technology, Uppsala University, Box 337, 751 05 Uppsala, Sweden

ARTICLE INFO

Article history:
Received 17 March 2016
Received in revised form
25 August 2016
Accepted 5 February 2017

Keywords:
System identification
Nonlinear models
Regularization
Probabilistic models
Bayesian learning
Gaussian processes
Monte Carlo methods

ABSTRACT

We consider a nonlinear state–space model with the state transition and observation functions expressed as basis function expansions. The coefficients in the basis function expansions are learned from data. Using a connection to Gaussian processes we also develop priors on the coefficients, for tuning the model flexibility and to prevent overfitting to data, akin to a Gaussian process state–space model. The priors can alternatively be seen as a regularization, and helps the model in generalizing the data without sacrificing the richness offered by the basis function expansion. To learn the coefficients and other unknown parameters efficiently, we tailor an algorithm using state-of-the-art sequential Monte Carlo methods, which comes with theoretical guarantees on the learning. Our approach indicates promising results when evaluated on a classical benchmark as well as real data.

1. Introduction

Nonlinear system identification (Ljung, 1999, 2010; Sjöberg et al., 1995) aims to learn nonlinear mathematical models from data generated by a dynamical system. We will tackle the problem of learning nonlinear state–space models with only weak assumptions on the nonlinear functions, and make use of the Bayesian framework (Peterka, 1981) to encode prior knowledge and assumptions to guide the otherwise too flexible model.

Consider the (time invariant) state–space model

$$x_{t+1} = f(x_t, u_t) + v_t, \qquad v_t \sim \mathcal{N}(0, Q), \tag{1a}$$

$$y_t = g(x_t, u_t) + e_t, \qquad e_t \sim \mathcal{N}(0, R). \tag{1b}$$

The variables are denoted as the state[1] $x_t \in \mathbb{R}^{n_x}$, which is not observed explicitly, the input $u_t \in \mathbb{R}^{n_u}$, and the output $y_t \in \mathbb{R}^{n_y}$. We will learn the state transition function $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \mapsto \mathbb{R}^{n_x}$ and the observation function $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \mapsto \mathbb{R}^{n_y}$ as well as Q and R from a set of training data of input–output signals $\{u_{1:T}, y_{1:T}\}$.

Consider a situation when a finite-dimensional linear, or other sparsely parameterized model, is too rigid to describe the behavior of interest, but only a limited data record is available so that any too flexible model would overfit (and be of no help in generalizing to events not exactly seen in the training data). In such a situation, a *systematic way to encode prior assumptions and thereby tuning the flexibility of the model* can be useful. For this purpose, we will take inspiration from Gaussian processes (GPs, Rasmussen & Williams, 2006) as a way to encode prior assumptions on $f(\cdot)$ and $g(\cdot)$. As illustrated by Fig. 1, the GP is a distribution over functions which gives a probabilistic model for interpolating and extrapolating from observed data. GPs have successfully been used in system identification for, e.g., response estimation, nonlinear ARX models and GP state–space models (Frigola-Alcade, 2015; Kocijan, 2016; Pillonetto & De Nicolao, 2010).

To parameterize $f(\cdot)$, we expand it using basis functions

$$f(x) = \sum_{j=0}^{m} w^{(j)} \phi^{(j)}(x), \tag{2}$$

and similarly for $g(\cdot)$. The set of basis functions is denoted by $\{\phi^{(j)}(\cdot)\}_{j=0}^{m}$, whose coefficients $\{w^{(j)}\}_{j=0}^{m}$ will be learned from data. By introducing certain *priors* $p(w^{(j)})$ *on the basis function coefficients* the connection to GPs will be made, based on a Karhunen–Loève expansion (Solin & Särkkä, 2014). We will thus

☆ This research is financially supported by the Swedish Research Council via the project *Probabilistic modeling of dynamical systems* (contract number: 621-2013-5524) and the Swedish Foundation for Strategic Research (SSF) via the project *ASSEMBLE*. The material in this paper was partially presented at the 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), December 13–16, 2015, Cancun, Mexico and at the 19th International Conference on Artificial Intelligence and Statistics (AISTATS), May 9–11, 2016, Cadiz, Spain. This paper was recommended for publication in revised form by Associate Editor Brett Ninness under the direction of Editor Torsten Söderström.

E-mail addresses: andreas.svensson@it.uu.se (A. Svensson), thomas.schon@it.uu.se (T.B. Schön).

[1] v_t and e_t are i.i.d. with respect to t, and x_t is thus Markov.

http://dx.doi.org/10.1016/j.automatica.2017.02.030
0005-1098/

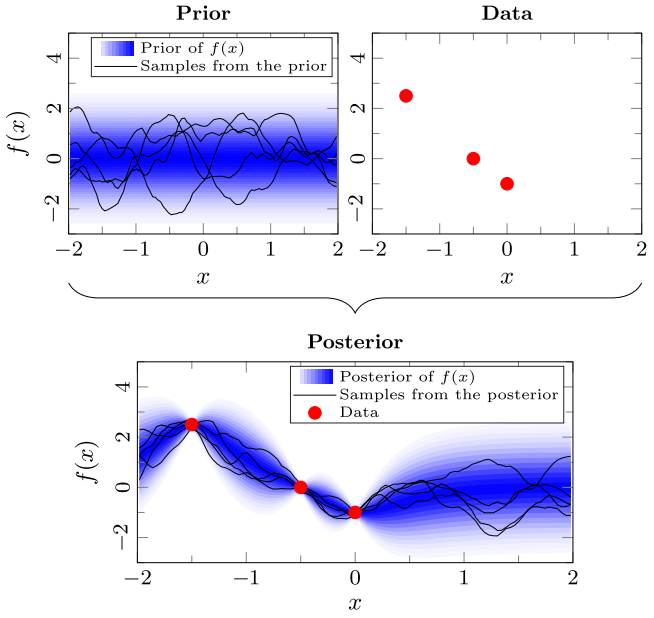

Fig. 1. The Gaussian process as a modeling tool for a one-dimensional function $f : \mathbb{R} \mapsto \mathbb{R}$. The prior distribution (upper left plot) is represented by the shaded blue color (the more intense color, the higher density), as well as 5 samples drawn from it. By combining the prior and the data (upper right plot), the posterior (lower plot) is obtained. The posterior mean basically interpolates between the data points, and adheres to the prior in regions where the data is not providing any information. This is clearly a desirable property when it comes to generalizing from the training data—consider the thought experiment of using a 2nd order polynomial instead. Further, the posterior also provides a quantification of the uncertainty present, high in data-scarce regions and low where the data provides knowledge about $f(\cdot)$.

be able to understand our model in terms of the well-established and intuitively appealing GP model, but still benefit from the computational advantages of the linear-in-parameter structure of (2). Intuitively, the idea of the priors $p(w^{(j)})$ is to keep $w^{(j)}$ 'small unless data convinces otherwise', or equivalently, introduce a regularization of $w^{(j)}$.

To learn the model (1), i.e., determine the basis function coefficients $w^{(j)}$, we tailor a learning algorithm using recent sequential Monte Carlo/particle filter methods (Kantas, Doucet, Singh, Maciejowski, & Chopin, 2015; Schön et al., 2015). The learning algorithm infers the posterior distribution of the unknown parameters from data, and come with theoretical guarantees. We will pay extra attention to the problem of finding the maximum mode of the posterior, or equivalent, regularized maximum likelihood estimation.

Our contribution is the development of a flexible nonlinear state–space model with a tailored learning algorithm, which together constitutes a new nonlinear system identification tool. The model can either be understood as a GP state–space model (generalized allowing for discontinuities, Section 3.2.3), or as a nonlinear state–space model with a regularized basis function expansion.

2. Related work

Important work using the GP in system identification includes impulse response estimation (Chen, Ohlsson, & Ljung, 2012; Pillonetto, Chiuso, & De Nicolao, 2011; Pillonetto & De Nicolao, 2010), nonlinear ARX models (Bijl, Schön, van Wingerden, & Verhaegen, 2016; Kocijan, Girard, Banko, & Murray-Smith, 2005), Bayesian learning of ODEs (Calderhead, Girolami, & Lawrence, 2008; Macdonald, Higham, & Husmeier, 2015; Wang & Barber, 2014) and the latent force model (Alvarez, Luengo, & Lawrence, 2013). In the GP state–space model (Frigola-Alcade, 2015) the transition function $f(\cdot)$ in a state–space model is learned with a GP prior, particularly relevant to this paper. A conceptually interesting contribution to the GP state–space model was made by Frigola, Lindsten, Schön, and Rasmussen (2013), using a Monte Carlo approach (similar to this paper) for learning. The practical use of Frigola et al. (2013) is however very limited, due to its extreme computational burden. This calls for approximations, and a promising approach is presented by Frigola, Chen, and Rasmussen (2014) (and somewhat generalized by Mattos et al., 2016), using inducing points and a variational inference scheme. Another competitive approach is Svensson, Solin, Särkkä, and Schön (2016), where we applied the GP approximation proposed by Solin and Särkkä (2014) and used a Monte Carlo approach for learning (Frigola-Alcade, 2015 covers the variational learning using the same GP approximation). In this paper, we extend this work by considering basis function expansions in general (not necessarily with a GP interpretation), introduce an approach to model discontinuities in $f(\cdot)$, as well as including both a Bayesian and a maximum likelihood estimation approach to learning.

To the best of our knowledge, the first extensive paper on the use of a basis function expansion inside a state–space model was written by Ghahramani and Roweis (1998), who also wrote a longer version (Roweis & Ghahramani, unpublished). The recent work by Tobar, Djurić, and Mandic (2015) resembles that of Ghahramani and Roweis (1998) on the modeling side, as they both use basis functions with locally concentrated mass spread in the state space. On the learning side, Ghahramani and Roweis (1998) use an expectation maximization (EM, Dempster, Laird, & Rubin, 1977) procedure with extended Kalman filtering, whilst (Tobar et al., 2015) use particle Metropolis–Hastings (Andrieu, Doucet, & Holenstein, 2010). There are basically three major differences between (Tobar et al., 2015) and our work. We will (i) use another (related) learning method, particle Gibbs, allowing us to take advantage of the linear-in-parameter structure of the model to increase the efficiency. Further, we will (ii) mainly focus on a different set of basis functions (although our learning procedure will be applicable also to the model used by Tobar et al., 2015), and – perhaps most important – (iii) we will pursue a systematic encoding of prior assumptions further than (Tobar et al., 2015), who instead assume $g(\cdot)$ to be known and use 'standard sparsification criteria from kernel adaptive filtering' as a heuristic approach to regularization.

There are also connections to Paduart et al. (2010), who use a polynomial basis inside a state–space model. In contrast to our work, however, Paduart et al. (2010) prevent the model from overfitting to the training data not by regularization, but by manually choosing a low enough polynomial order and terminating the learning procedure prematurely (early stopping). Paduart et al. are, in contrast to us, focused on the frequency properties of the model and rely on optimization tools. An interesting contribution by Paduart et al. is to first use classical methods to find a linear model, which is then used to initialize the linear term in the polynomial expansion. We suggest to also use this idea, either to initialize the learning algorithm, or use the nonlinear model only to describe deviations from an initial linear state–space model.

Furthermore, there are also connections to our previous work (Svensson, Schön, Solin, & Särkkä, 2015), a short paper only outlining the idea of learning a regularized basis function expansion inside a state–space model. Compared to Svensson et al. (2015), this work contains several extensions and new results. Another recent work using a regularized basis function expansion for nonlinear system identification is that of Delgado, Agüero, Goodwin, and Mendes (2015), however not in the state–space model framework. Delgado et al. (2015) use rank constrained optimization, resembling an L^0-regularization. To