Automatica 93 (2018) 343–351

Contents lists available at ScienceDirect

Automatica

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

Brief paper

Recursive nonlinear-system identification using latent variables[☆]

Per Mattsson [a,*], Dave Zachariah [b], Petre Stoica [b]

[a] Department of Electronics, Mathematics and Natural Sciences, University of Gävle, Gävle, Sweden
[b] Department of Information Technology, Uppsala University, Uppsala, Sweden

ARTICLE INFO

Article history:
Received 28 June 2016
Received in revised form 29 August 2017
Accepted 19 December 2017

Keywords:
Nonlinear systems
Multi-input/multi-output systems
System identification

ABSTRACT

In this paper we develop a method for learning nonlinear system models with multiple outputs and inputs. We begin by modeling the errors of a nominal predictor of the system using a latent variable framework. Then using the maximum likelihood principle we derive a criterion for learning the model. The resulting optimization problem is tackled using a majorization–minimization approach. Finally, we develop a convex majorization technique and show that it enables a recursive identification method. The method learns parsimonious predictive models and is tested on both synthetic and real nonlinear systems.

1. Introduction

In this paper we consider the problem of learning a nonlinear dynamical system model with multiple outputs $\mathbf{y}(t)$ and multiple inputs $\mathbf{u}(t)$ (when they exist). Generally this identification problem can be tackled using different model structures, with the class of linear models being arguably the most well studied in engineering, statistics and econometrics (Barber, 2012; Bishop, 2006; Box, Jenkins, Reinsel, & Ljung, 2015; Ljung, 1998; Söderström & Stoica, 1988).

Linear models are often used even when the system is known to be nonlinear (Enqvist, 2005; Schoukens, Vaes, & Pintelon, 2016). However certain nonlinearities, such as saturations, cannot always be neglected. In such cases using block-oriented models is a popular approach to capture static nonlinearities (Giri & Bai, 2010). Recently, such models have been given semiparametric formulations and identified using machine learning methods, cf. Pillonetto (2013) and Pillonetto, Dinuzzo, Chen, De Nicolao, and Ljung (2014). To model nonlinear dynamics a common approach is to use NARMAX models (Billings, 2013; Sjöberg et al., 1995).

In this paper we are interested in recursive identification methods (Ljung & Söderström, 1983). In cases where the model structure is linear in the parameters, recursive least-squares can be applied. For certain models with nonlinear parameters, the extended recursive least-squares has been used (Chen, 2004). Another popular approach is the recursive prediction error method which has been developed, e.g., for Wiener models, Hammerstein models, and polynomial state-space models (Mattsson & Wigren, 2016; Tayamon, Wigren, & Schoukens, 2012; Wigren, 1993).

Nonparametric models are often based on weighted sums of the observed data (Roll, Nazin, & Ljung, 2005). The weights vary for each predicted output and the number of weights increases with each observed datapoint. The weights are typically obtained in a batch manner; in Bai and Liu (2007) and Bijl, van Wingerden, Schön, and Verhaegen (2015) they are computed recursively but must be recomputed for each new prediction of the output.

For many nonlinear systems, however, linear models work well as an initial approximation. The strategy in Paduart et al. (2010) exploits this fact by first finding the best linear approximation using a frequency domain approach. Then, starting from this approximation, a nonlinear polynomial state-space model is fitted by solving a nonconvex problem. This two-step method cannot be readily implemented recursively and it requires input signals with appropriate frequency domain properties.

In this paper, we start from a nominal model structure. This class can be based on insights about the system, e.g. that linear model structures can approximate a system around an operating point. Given a record of past outputs, $\mathbf{y}(t)$ and inputs $\mathbf{u}(t)$, that is,

$$\mathcal{D}_t \triangleq \left\{ (\mathbf{y}(1), \mathbf{u}(1)), \dots, (\mathbf{y}(t), \mathbf{u}(t)) \right\},$$

a nominal model yields a predicted output $\mathbf{y}_0(t+1)$ which differs from the output $\mathbf{y}(t+1)$. The resulting prediction error is denoted $\boldsymbol{\varepsilon}(t+1)$ (Ljung, 1999). By characterizing the nominal prediction errors in a data-driven manner, we aim to develop a refined predictor model of the system. Thus we integrate classic and data-driven system modeling approaches in a natural way.

[☆] This work has been partly supported by the Swedish Research Council (VR) under contracts 621-2014-5874 and 2016-06079. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Brett Ninness under the direction of Editor Torsten Söderström.

[*] Corresponding author.
E-mail addresses: per.mattsson@hig.se (P. Mattsson), dave.zachariah@it.uu.se (D. Zachariah), ps@it.uu.se (P. Stoica).

https://doi.org/10.1016/j.automatica.2018.03.007

The general model class and problem formulation are introduced in Section 2. Then in Section 3 we apply the principle of maximum likelihood to derive a statistically motivated learning criterion. In Section 4 this nonconvex criterion is minimized using a majorization–minimization approach that gives rise to a convex user-parameter free method. We derive a computationally efficient recursive algorithm for solving the convex problem, which can be applied to large data sets as well as online learning scenarios. In Section 5, we evaluate the proposed method using both synthetic and real data examples.

In a nutshell, the contribution of the paper is a modeling approach and identification method for nonlinear multiple input–multiple output systems that:

- explicitly separates modeling based on application-specific insights from general data-driven modeling,
- circumvents the choice of regularization parameters and initialization points,
- learns parsimonious predictor models,
- admits a computationally efficient implementation.

Notation: $\mathbf{E}_{i,j}$ denotes the ijth standard basis matrix. $\otimes$ and $\odot$ denote the Kronecker and Hadamard products, respectively. $\text{vec}(\cdot)$ is the vectorization operation. $\|\mathbf{x}\|_2$, $\|\mathbf{x}\|_1$ and $\|\mathbf{X}\|_{\mathbf{W}} = \sqrt{\text{tr}\{\mathbf{X}^\top \mathbf{W}\mathbf{X}\}}$, where $\mathbf{W} \succ \mathbf{0}$, denote ℓ_2-, ℓ_1- and weighted norms, respectively. The Moore–Penrose pseudoinverse of $\mathbf{X}$ is denoted $\mathbf{X}^\dagger$.

Remark 1. An implementation of the proposed method is available at https://github.com/magni84/lava.

2. Problem formulation

Given $\mathcal{D}_{t-1}$, the n_y-dimensional output of a system can always be written as

$$\mathbf{y}(t) = \mathbf{y}_0(t) + \boldsymbol{\varepsilon}(t), \tag{1}$$

where $\mathbf{y}_0(t)$ is any one-step-ahead predictor based on a nominal model. Here we consider nominal models on the form

$$\mathbf{y}_0(t) = \boldsymbol{\Theta}\boldsymbol{\varphi}(t), \tag{2}$$

where the $p \times 1$ vector $\boldsymbol{\varphi}(t)$ is a given function of $\mathcal{D}_{t-1}$ and $\boldsymbol{\Theta}$ denotes the unknown parameters.

Remark 2. A typical example of $\boldsymbol{\varphi}(t)$ is

$$\boldsymbol{\varphi}(t) = [\mathbf{y}^\top(t-1) \cdots \mathbf{y}^\top(t-n_a)\, \mathbf{u}^\top(t-1) \cdots \\ \mathbf{u}^\top(t-n_b)\; 1]^\top, \tag{3}$$

in which case the nominal predictor is linear in the data and therefore captures the linear system dynamics. Nonlinearities can be incorporated if such are known about the system, in which case $\boldsymbol{\varphi}(t)$ will be nonlinear in the data.

The popular ARX model structure, for instance, can be cast into the framework (1) and (2) by assuming that the nominal prediction error $\boldsymbol{\varepsilon}(t)$ is a white noise process (Ljung, 1998; Söderström & Stoica, 1988). For certain systems, (2) may accurately describe the dynamics of the system around its operation point and consequently the white noise assumption on $\boldsymbol{\varepsilon}(t)$ may be a reasonable approximation. However, this ceases to be the case even for systems with weak nonlinearities, cf. Enqvist (2005).

Next, we develop a data-driven model of the prediction errors $\boldsymbol{\varepsilon}(t)$ in (1), conditioned on the past data $\mathcal{D}_{t-1}$. Specifically, we assume the conditional model

$$\boldsymbol{\varepsilon}(t) \mid \mathcal{D}_{t-1} \;\sim\; \mathcal{N}(\mathbf{Z}\boldsymbol{\gamma}(t), \boldsymbol{\Sigma}), \tag{4}$$

where $\mathbf{Z}$ is an $n_y \times q$ matrix of unknown latent variables, $\boldsymbol{\Sigma}$ is an unknown covariance matrix, and the $q \times 1$ vector $\boldsymbol{\gamma}(t)$ is any given function of $\mathcal{D}_{t-1}$. This is a fairly general model structure that can capture correlated data-dependent nominal prediction errors.

Note that when $\mathbf{Z} \equiv \mathbf{0}$, the prediction errors are temporally white and the nominal model (2) captures all relevant system dynamics. The latent variable is modeled as random here. Before data collection, we assume $\mathbf{Z}$ to have mean $\mathbf{0}$ as we have no reason to depart from the nominal model assumptions until after observing data. Using a Gaussian distribution, we thus get

$$\text{vec}(\mathbf{Z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{D}), \tag{5}$$

where $\mathbf{D}$ is an unknown covariance matrix.

Our goal here is to identify a refined predictor of the form

$$\boxed{\hat{\mathbf{y}}(t) = \underbrace{\widehat{\boldsymbol{\Theta}}\boldsymbol{\varphi}(t)}_{\hat{\mathbf{y}}_0(t)} + \underbrace{\widehat{\mathbf{Z}}\boldsymbol{\gamma}(t)}_{\hat{\boldsymbol{\varepsilon}}(t)},} \tag{6}$$

from a data set $\mathcal{D}_{t-1}$, by maximizing the likelihood function. The first term is an estimate of the nominal predictor model while the second term tries to capture structure in the data that is not taken into account by the nominal model. Note that when $\widehat{\mathbf{Z}}$ is sparse we obtain a parsimonious predictor model.

Remark 3. The model (1)–(4) implies that we can write the output in the equivalent form

$$\mathbf{y}(t) = \boldsymbol{\Theta}\boldsymbol{\varphi}(t) + \mathbf{Z}\boldsymbol{\gamma}(t) + \mathbf{v}(t),$$

where $\mathbf{v}(t)$ is a white process with covariance $\boldsymbol{\Sigma}$. In order to formulate a flexible data-driven error model (4), we overparametrize it using a high-dimensional $\boldsymbol{\gamma}(t)$. In this case, regularization of $\mathbf{Z}$ is desirable, which is achieved by (5). Note that $\mathbf{D}$ and $\boldsymbol{\Sigma}$ are both unknown. Estimating these covariance matrices corresponds to using a data-adaptive regularization, as we show in subsequent sections.

Remark 4. The nonlinear function $\boldsymbol{\gamma}(t)$ of $\mathcal{D}_{t-1}$ can be seen as a basis expansion which is chosen to yield a flexible model structure of the errors. In the examples below we will use the Laplace operator basis functions (Solin & Särkkä, 2014). Other possible choices include the polynomial basis functions, Fourier basis functions, wavelets, etc. Ljung (1998), Sjöberg et al. (1995) and Van den Hof and Ninness (2005).

Remark 5. In (6), $\hat{\mathbf{y}}(t)$ is a one-step-ahead predictor. However, the framework can be readily applied to k-step-ahead prediction where $\boldsymbol{\varphi}(t)$ and $\boldsymbol{\gamma}(t)$ depend on $\mathbf{y}(1), \ldots, \mathbf{y}(t-k)$.

3. Latent variable framework

Given a record of N data samples, $\mathcal{D}_N$, our goal is to estimate $\boldsymbol{\Theta}$ and $\mathbf{Z}$ to form the refined predictor (6). In Section 3.1, we employ the maximum likelihood approach based on the likelihood function $p(\mathbf{Y}|\boldsymbol{\Theta}, \mathbf{D}, \boldsymbol{\Sigma})$, which requires the estimation of nuisance parameters $\mathbf{D}$ and $\boldsymbol{\Sigma}$. For notational simplicity, we write the parameters as $\boldsymbol{\Omega} = \{\boldsymbol{\Theta}, \mathbf{D}, \boldsymbol{\Sigma}\}$ and in Section 3.2 we show how an estimator of $\mathbf{Z}$ is obtained as a function of $\boldsymbol{\Omega}$ and $\mathcal{D}_N$.

3.1. Parameter estimation

We write the output samples in matrix form as

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}(1) & \cdots & \mathbf{y}(N) \end{bmatrix} \in \mathbb{R}^{n_y \times N}.$$