Systems & Control Letters 120 (2018) 9–16

Contents lists available at ScienceDirect

Systems & Control Letters

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

Bayesian augmented Lagrangian algorithm for system identification

Xiaoquan Tang [a], Long Zhang [b,*], Xiuting Li [a]

[a] School of Automation, Huazhong University of Science and Technology, China
[b] School of Electrical and Electronic Engineering, University of Manchester, UK

ARTICLE INFO

Article history:
Received 29 March 2018
Received in revised form 4 June 2018
Accepted 19 July 2018

Keywords:
System identification
Weighted l_1 minimization
Augmented Lagrangian
Bayesian
NARX

ABSTRACT

Nonlinear Auto-Regressive model with eXogenous input (NARX) is one of the most popular black-box model classes that can describe many nonlinear systems. The structure determination is the most challenging and important part during the system identification. NARX can be formulated as a linear-in-the-parameters model, then the identification problem can be solved to obtain a sparse solution from the viewpoint of the weighted l_1 minimization problem. Such an optimization problem not only minimizes the sum squares of model errors but also the sum of reweighted model parameters. In this paper, a novel algorithm named Bayesian Augmented Lagrangian Algorithm (BAL) is proposed to solve the weighted l_1 minimization problem, which is able to obtain a sparse solution and enjoys fast computation. This is achieved by converting the original optimization problem into distributed suboptimization problems solved separately and penalizing the overall complex model to avoid overfitting under the Bayesian framework. The regularization parameter is also iteratively updated to obtain a satisfied solution. In particular, a solver with guaranteed convergence is constructed and the corresponding theoretical proof is given. Two numerical examples have been used to demonstrate the effectiveness of the proposed method in comparison to several popular methods.

1. Introduction

NARX is a popular model class that can describe complex dynamic behaviour of nonlinear systems [1,2]. The importance of identifying nonlinear systems using NARX has been widely recognized owing to the following advantages. First, NARX may provide a more compact model for nonlinear system compared to Volterra series model class. Second, NARX can be formulated as a linear-in-the-parameters model when the unknown parameters in the nonlinear functions are given a priori. Then the model structure can be determined using regression algorithms, such as Least absolute shrinkage and selection operator (Lasso) [3] and sparse Bayesian learning (SBL) [4]. However, the NARX model structure given a priori often contains redundant terms. In other words, the predetermined model term dictionary is generally huge and most terms in the dictionary should not be selected into the final model. Therefore, structure determination is a key challenge and an important part in system identification.

Subset selection methods have been widely used to select important terms from the dictionary, leading to a parsimonious model. For the linear-in-the-parameters model, it can be considered as finding a sparse solution which can be solved from the viewpoint of the l_1 minimization problem. Lasso is a widely used method to solve the l_1 minimization problem, which tends to find a compromise model between model accuracy and complexity. However, when the columns of dictionary are highly correlated rather than orthogonal or nearly so, Lasso algorithm generally leads to a suboptimal model with some redundant terms.

To obtain a more compact model, many regression problems are converted into the weighted l_1 minimization problem to find a maximally sparse solution. It also has been proved that weighted l_1 minimization tends to perform better than conventional l_1 minimization under certain conditions [5]. SBL is recently proposed under the Bayesian framework to solve the weighted l_1 minimization problem and has been proved to be an efficient method in some practical applications. SBL has several advantages summarized as follows. Based on the priori knowledge of the unknown system, it can build a sparse model by selecting candidate dictionary terms. In addition, it can iteratively calculate the solution and can avoid overfitting problem with pruning method. However, the solution is calculated by using third party solvers (e.g. CVX [6]) at each iterative step, leading to large computations.

In this paper, the main objective of the proposed BAL method is to build a sparse NARX model in a computationally efficient manner. This is achieved by transforming the single weighted l_1 optimization problem into several distributed suboptimization problems, and then deriving the corresponding solvers. Meanwhile, the regularization parameters that control the model complexity are iteratively updated under Bayesian framework. The new idea is

* Corresponding author.
E-mail address: long.zhang@manchester.ac.uk (L. Zhang).

https://doi.org/10.1016/j.sysconle.2018.07.011
0167-6911/

inspired by both Split Augmented Lagrangian Shrinkage Algorithm (SALSA) that is recently proposed for solving distributed optimization problem and SBL that is able to produce a sparse model. The new BAL method enjoys the advantages of the both SALSA and SBL methods but avoid their disadvantages as it can build a sparse model than SALSA and runs faster than SBL. More specifically,

- Using Bayesian learning can penalize the complex model to avoid overfitting problem and it is able to capture the model uncertainty [4]. In addition, the information about the unknown system can be converted into priors which can help to identify the unknown system.
- BAL converts the weighted l_1 minimization problem into several subproblems that can be exactly solved without using third party solvers (e.g. CVX). The memory and computational requirement can be reduced in comparison to those centralized methods [7]. Therefore, the running time of procedure could be saved.
- The regularization parameter is iteratively updated to increase the opportunity to find a satisfied solution.

The theoretical analysis regarding to solution existence, uniqueness and algorithm convergence is given. Two nonlinear examples are used to illustrate the effectiveness of BAL, and several popular methods are used for comparison, including SBL, Lasso, SALSA and Orthogonal Forward Regression method (OFR) method.

2. Preliminary

2.1. NARX model

NARX model is a widely used representation for input–output relationship of an unknown nonlinear system. The system can be described by some unknown function of lagged system inputs and outputs [8]:

$$\begin{aligned} y(t) &= f(y(t-1),\ldots,y(t-n_y),u(t-1),\ldots,u(t-n_u))+\xi(t)\\ &= f(x(t))+\xi(t)\end{aligned}$$

where $u(t)$, $y(t)$ represent system input and output at the time interval t, respectively, with $t = 1,2,\ldots,N$ and N being the training data size. n_u and n_y are the largest lags of input and output. Assuming $\xi(t)$ is i.i.d. Gaussian distributed noise with zero mean and variance σ^2.

Suppose the model input is $x(t) = [y(t-1),\ldots,y(t-n_y),u(t-1),\ldots,u(t-n_u)]$, then the candidate dictionary can be represented as [9]

$$\mathbf{P} = [p_1(x(t)), p_2(x(t)),\ldots,p_M(x(t))]$$

Here $\mathbf{P}$ is the $N \times M$ matrix which includes some linear and nonlinear terms of $x(t)$. The NARX model representation can be rewritten as a linear combination of some nonlinear functions such as polynomials and neural networks

$$y(t) = \sum_{i=1}^{M} p_i(x(t))\Theta_i + \xi(t)$$

which can be described as the following matrix format:

$$\mathbf{y} = \mathbf{P}\boldsymbol{\Theta} + \boldsymbol{\xi} \tag{1}$$

where vector $\mathbf{y} = [y(1),y(2),\ldots,y(N)]^T$ represents the system output, vector $\boldsymbol{\xi} = [\xi(1),\xi(2),\ldots,\xi(N)]^T$ represents the residual, and $\boldsymbol{\Theta} = [\Theta_1,\Theta_2,\ldots,\Theta_M]^T$ represents the parameter being estimated.

For obtaining an optimal representation of the unknown nonlinear system, the size of predetermined candidate pool $\mathbf{P}$ is often large enough so that it owns the ability to describe nonlinearities of the unknown nonlinear system. However, most of the terms in the candidate pool are redundant and should not be selected into the final model. A sparse solution with good generalization performance is always desirable.

2.2. Sparse Bayesian Learning

Recently, SBL is proposed as an iterative reweighted l_1 method to build a sparse model. The main idea of SBL is briefly reviewed as following. All the unknowns are considered as stochastic variables which have certain probability distributions in the process of Bayesian modelling [4]. For $\mathbf{y} = \mathbf{P}\boldsymbol{\Theta} + \boldsymbol{\xi}$, the likelihood of the data $\mathbf{y}$ given $\boldsymbol{\Theta}$ is described as

$$\mathcal{P}(\mathbf{y}|\boldsymbol{\Theta}) = \mathcal{N}(\mathbf{y}|\mathbf{P}\boldsymbol{\Theta},\lambda\mathbf{I}) \propto \exp\Big[-\frac{1}{2\lambda}\|\mathbf{y}-\mathbf{P}\boldsymbol{\Theta}\|_2^2\Big]$$

where $\lambda = \sigma^2$. Suppose $\mathcal{P}(\boldsymbol{\Theta})$ has the following prior distribution:

$$\mathcal{P}(\boldsymbol{\Theta}) \propto \exp\Big[-\frac{1}{2}\sum_{i=1}^{M} g_c(\Theta_i)\Big]$$

The function $g_c(\Theta)$ is usually concave, non-decreasing for $|\Theta|$, which can enforce sparsity of the solution. Meanwhile, suppose $\mathcal{P}(\boldsymbol{\Theta}) = \prod_{i=1}^{M}\mathcal{P}(\Theta_i)$, then according to the Bayes' rule, the posterior distribution over $\boldsymbol{\Theta}$ can be calculated:

$$\mathcal{P}(\boldsymbol{\Theta}|\mathbf{y}) = \frac{\mathcal{P}(\mathbf{y}|\boldsymbol{\Theta})\mathcal{P}(\boldsymbol{\Theta})}{\int \mathcal{P}(\mathbf{y}|\boldsymbol{\Theta})\mathcal{P}(\boldsymbol{\Theta})d\boldsymbol{\Theta}}$$

However, the posterior $\mathcal{P}(\boldsymbol{\Theta}|\mathbf{y})$ is non-Gaussian, which makes the identification problem intractable. Generally, one tends to approximate $\mathcal{P}(\boldsymbol{\Theta}|\mathbf{y})$ as the Gaussian distribution, then the problem can be solved efficiently. Therefore, an optimal hyperparameter $\gamma = [\gamma_1,\ldots,\gamma_M] \in \mathcal{R}_+^{\mathcal{M}}$ is rationally estimated such that the Gaussian-distribution $\mathcal{P}(\boldsymbol{\Theta}|\mathbf{y},\widehat{\gamma})$ is a good relaxation to $\mathcal{P}(\boldsymbol{\Theta}|\mathbf{y})$. For more details, please review [4]. Under the Bayesian framework, the problem can be solved from the following viewpoint [7]:

$$\min_{\gamma\geq 0,\boldsymbol{\Theta}} \|\mathbf{P}\boldsymbol{\Theta}-\mathbf{y}\|_2^2 + \lambda\boldsymbol{\Theta}^T\boldsymbol{\Gamma}^{-1}\boldsymbol{\Theta} + \log|\lambda\mathbf{I}+\mathbf{P}\boldsymbol{\Gamma}\mathbf{P}^T| \tag{2}$$

with $\boldsymbol{\Gamma} = \mathrm{diag}[\gamma]$. However, it is difficult to directly obtain model coefficients $\boldsymbol{\Theta}$ and γ according to the formula (2). Therefore, we rewrite Eq. (2) as

$$\min_{\gamma\geq 0,\boldsymbol{\Theta}} g(\boldsymbol{\Theta},\gamma) - h(\gamma)$$

with $g(\boldsymbol{\Theta},\gamma) = \|\mathbf{P}\boldsymbol{\Theta}-\mathbf{y}\|_2^2 + \lambda\sum_j \frac{\Theta_j^2}{\gamma_j}$ and $h(\gamma) = -\log|\lambda\mathbf{I}+\mathbf{P}\boldsymbol{\Gamma}\mathbf{P}^T|$. Here, $g(\boldsymbol{\Theta},\gamma)$ is jointly convex for $\boldsymbol{\Theta},\gamma$ and $h(\gamma)$ is convex for γ. Since function $h(\gamma)$ is differentiable over γ, $\widehat{\boldsymbol{\Theta}}_{k+1}$ and $\widehat{\gamma}_{k+1}$ can be obtained by

$$[\widehat{\boldsymbol{\Theta}}_{k+1},\widehat{\gamma}_{k+1}] = \arg\min_{\gamma\geq 0,\boldsymbol{\Theta}} g(\boldsymbol{\Theta},\gamma) - \nabla_\gamma h(\widehat{\gamma}_k)^T\gamma \tag{3}$$

Based on the principles in convex analysis, the negative gradient of $h(\gamma)$ at γ can be expressed as

$$\begin{aligned} -\nabla_\gamma h(\widehat{\gamma}_k)^T &= -\nabla_\gamma(-\log|\lambda\mathbf{I}+\mathbf{P}\boldsymbol{\Gamma}\mathbf{P}^T|)|_{\gamma=\widehat{\gamma}_k}\\ &= \mathrm{diag}\big[\mathbf{P}^T(\lambda\mathbf{I}+\mathbf{P}\boldsymbol{\Gamma}_k\mathbf{P}^T)^{-1}\mathbf{P}\big]\end{aligned}$$

For convenience, define $\alpha_k = \mathrm{diag}\big[\mathbf{P}^T(\lambda\mathbf{I}+\mathbf{P}\boldsymbol{\Gamma}_k\mathbf{P}^T)^{-1}\mathbf{P}\big]$. With these definitions, the optimization problem (3) can be further formulated as

$$[\widehat{\boldsymbol{\Theta}}_{k+1},\widehat{\gamma}_{k+1}] = \arg\min_{\gamma\geq 0,\boldsymbol{\Theta}} \|\mathbf{P}\boldsymbol{\Theta}-\mathbf{y}\|_2^2 + \lambda\sum_j\Big(\frac{\Theta_j^2}{\gamma_j} + (\alpha_k)_j\gamma_j\Big) \tag{4}$$

here $(\alpha_k)_j$ is the j_{th} diagonal element of the matrix α_k. It is worth pointing out that the function (4) is jointly convex in $\boldsymbol{\Theta}$, γ, which