Bayesian Augmented Lagrangian Algorithm for System Identification

DOI:
10.1016/j.sysconle.2018.07.011

Document Version
Accepted author manuscript

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*

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 penalising 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.

Keywords: System Identification, Weighted l_1 Minimization, Augmented Lagrangian, Bayesian, NARX

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 as a prior. 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 as a prior 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 prior 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 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 penalise 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 centralised 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]:

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

where $u(t)$, $y(t)$ represent system input and output at the time interval t, respectively, with $t = 1, 2, \cdots, N$ 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 $x(t) = [y(t-1), \cdots, y(t-n_y), u(t-1), \cdots, u(t-n_u)]$, then the candidate dictionary can be represented as [9]

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

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

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

which can be described as the following matrix format

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

where vector $\mathbf{y} = [y(1), y(2), \cdots, y(N)]^T$ represents the system output, vector $\xi = [\xi(1), \xi(2), \cdots, \xi(N)]^T$ represents the residual, and $\mathbf{\Theta} = [\Theta_1, \Theta_2, \cdots, \Theta_M]^T$ 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 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 $\mathbf{y} = \mathbf{P\Theta} + \xi$, the likelihood of the data $\mathbf{y}$ given $\mathbf{\Theta}$ is described as

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 $\mathbf{\Theta}, \gamma$, which can be globally minimised by firstly solving γ and then $\mathbf{\Theta}$. More specifically, given $\mathbf{\Theta}$, $\widehat{\gamma}_{k+1}$ can be estimated by

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

with $(\widehat{\gamma}_{k+1})_j = |\Theta_j|/\sqrt{(\alpha_k)_j}$. In turn, injecting $\widehat{\gamma}_{k+1}$ into the equation (4), we can calculate $\widehat{\mathbf{\Theta}}_{k+1}$ by

$$\begin{aligned}\widehat{\mathbf{\Theta}}_{k+1} &= \arg\min_{\mathbf{\Theta}} \parallel \mathbf{P}\mathbf{\Theta} - \mathbf{y} \parallel_2^2 + \lambda \sum_j (\frac{\Theta_j^2}{(\widehat{\gamma}_{k+1})_j} + (\alpha_k)_j (\widehat{\gamma}_{k+1})_j) \\ &= \arg\min_{\mathbf{\Theta}} \|\mathbf{P}\mathbf{\Theta} - \mathbf{y}\|_2^2 + 2\lambda \sum_j \sqrt{(\alpha_k)_j}|\Theta_j|\end{aligned}$$

The equation above can be simplified as

$$\widehat{\mathbf{\Theta}}_{k+1} = \arg\min_{\mathbf{\Theta}} \frac{1}{2}\|\mathbf{P}\mathbf{\Theta} - \mathbf{y}\|_2^2 + \lambda\|G\mathbf{\Theta}\|_1 \tag{5}$$

where $G = \mathrm{diag}[\mathrm{w_k}]$ is a diagonal matrix and $(w_k)_j$ is the j_{th} diagonal element with $(w_k)_j = \sqrt{(\alpha_k)_j}$. The α_{k+1} can be calculated with

$$\alpha_{k+1} = \mathrm{diag}\big[\mathbf{P}^{\mathrm{T}}(\lambda\mathbf{I} + \mathbf{P}\mathbf{\Gamma}_{\mathrm{k+1}}\mathbf{P}^{\mathrm{T}})^{-1}\mathbf{P}\big] \tag{6}$$

with $(\widehat{\gamma}_{k+1})_j = |(\widehat{\mathbf{\Theta}}_{k+1})_j|/\sqrt{(\alpha_k)_j}$.

3. The idea of BAL

In this paper, BAL is proposed to build a sparse model in a computationally efficient manner. This is achieved by transforming the weighted l_1 minimization problem (5) into several subproblems which can be solved separately without using third party solvers. In addition, the value of the regularization parameter is updated at each iteration so that it can increase the opportunity to obtain a satisfied solution.

3.1. Converting to suboptimization problems

The constrained optimization formulation of the weighted l_1 minimization problem (5) can be expressed as

$$\min_{\mathbf{\Theta}, v\in R^M} f_1(\mathbf{\Theta}) + f_2(v) \quad s.t.\ \ v = G\mathbf{\Theta} \tag{7}$$

with $f_1(\mathbf{\Theta}) = \frac{1}{2}\|\mathbf{y} - \mathbf{P}\mathbf{\Theta}\|_2^2$ and $f_2(v) = \lambda\|v\|_1$. Since the constraint of the problem (7) can be rewritten as $\|v - G\mathbf{\Theta}\|_2^2 = \mathbf{0}$, the constrained problem (7) can be converted into a quadratic penalty problem

$$\min_{\mathbf{\Theta}, v\in R^M} f_1(\mathbf{\Theta}) + f_2(v) + \frac{\mu}{2} \parallel G\mathbf{\Theta} - v \parallel_2^2 \quad s.t.\ \ v - G\mathbf{\Theta} = \mathbf{0} \tag{8}$$

here μ is the Lagrange multiplier. Increasing μ helps to force the solution of the problem (8) to approximate that of the weighted l_1 minimization problem (5). With the Augmented Lagrangian method, the optimization problem (8) can be further represented as

$$L_\mu(\mathbf{\Theta}, v, u) = f_1(\mathbf{\Theta}) + f_2(v) - u^T(G\mathbf{\Theta} - v) + \frac{\mu}{2} \parallel G\mathbf{\Theta} - v \parallel_2^2 \tag{9}$$

where u is the dual variable. The problem (9) could be solved by alternating minimization with respect to $\mathbf{\Theta}$, u and v, while keeping other variables fixed. Under the condition that $v = G\mathbf{\Theta}$, the problem (9) can be

simplified as the weighted l_1 minimization problem (5). In other words, given v_k, G and u_k, $\widehat{\mathbf{\Theta}}_{k+1}$ obtained from equation (9) can be considered as the solution of the problem (5), if it satisfies $v_k = G\widehat{\mathbf{\Theta}}_{k+1}$. Given the estimation $\widehat{\mathbf{\Theta}}_{k+1}$, the weighted matrix G can be updated according to the equation (5) and (6), namely

$$\alpha_{k+1} = \mathrm{diag}\big[\mathbf{P}^{\mathrm{T}}(\lambda\mathbf{I} + \mathbf{P}\mathbf{\Gamma}_{\mathrm{k+1}}\mathbf{P}^{\mathrm{T}})^{-1}\mathbf{P}\big]$$

with $(\widehat{\gamma}_{k+1})_j = |(\widehat{\mathbf{\Theta}}_{k+1})_j|/\sqrt{(\alpha_k)_j}$. Therefore, we have $G = \mathrm{diag}[\mathrm{w_{k+1}}]$ with $(w_{k+1})_j = \sqrt{(\alpha_{k+1})_j}$.

3.2. Solving subproblems

The specific solution of the problem (9) can be solved from several subproblems. Specifically, replace u by the variable $d = u/\mu$ and substitute d into the equation (9), then we have

$$L_\mu(\mathbf{\Theta}, v, d) = f_1(\mathbf{\Theta}) + f_2(v) + \frac{\mu}{2}\|G\mathbf{\Theta} - v - d\|_2^2$$

Then the solution can be obtained by solving the following subproblems [10]

$$\begin{aligned}
&\widehat{\mathbf{\Theta}}_{k+1} = \arg\min_{\mathbf{\Theta}} f_1(\mathbf{\Theta}) + \tfrac{\mu}{2}\|G\mathbf{\Theta} - v_k - d_k\|_2^2 \\
&v_{k+1} = \arg\min_v f_2(v) + \tfrac{\mu}{2}\|G\widehat{\mathbf{\Theta}}_{k+1} - v - d_k\|_2^2 \\
&d_{k+1} = d_k - (G\widehat{\mathbf{\Theta}}_{k+1} - v_{k+1})
\end{aligned} \tag{10}$$

The solution of subproblems (10) can be exactly solved, which will be specifically introduced in **Theorem 2**. It is worth pointing out that G in the original SALSA is chosen as unit diagonal matrix $\mathbf{I}$, namely, $G = \mathbf{I}$. SALSA is a simple and special case of BAL. Here, G is iteratively calculated from the Bayesian viewpoint with the assumption that $v_k = G\widehat{\mathbf{\Theta}}_{k+1}$.

3.3. Tuning regularization parameter λ

To obtain a model with good generalization performance, it is necessary to set a proper value of the regularization parameter. However, it is seldom known as a prior. Therefore, adaptively adjusting the value of λ according to the previous modelling error is used. Now, we define modelling error at $k+1_{th}$ iteration as

$$Err_{k+1} = \frac{1}{2}\|\mathbf{y} - \mathbf{P}\widehat{\mathbf{\Theta}}_{k+1}\|_2^2$$

Then define $\beta_{k+1} = |Err_{k+1}/Err_k - 1|, (k = 0, \cdots, k_{max})$. The principle for adjusting parameter λ_{k+1} is that if $\beta_{k+1} < \pi$ ($\pi \in [0, 1]$), then $\lambda_{k+1} = a\lambda_k$ with $a > 1$, otherwise $\lambda_{k+1} = c\lambda_k$ with $0 < c < 1$. Updating the regularization parameter at each step can increase the opportunity to build a sparse model with less iterations.

3.4. The stopping criterion

BAL could produce a sparse solution if a proper stopping criterion is satisfied. The stopping criterion is important for the iterative algorithm, since the solution could be different at each iterative step. Before introducing the stopping criterion of the proposed method, we first give the following definition and assumption. Define that the set loc_k contains the location of nonzero coefficients of $\mathbf{\Theta}$ at k_{th} iteration and *sign* is a

sign function.

Assumption 1: Assume that the sign and location of nonzero coefficients of $\widehat{\mathbf{\Theta}}_i(i = 1, 2, \cdots, k)$ obtained during the iterative process could be different until the estimation of $\widehat{\mathbf{\Theta}}_{k+1}$ is similar with that of $\widehat{\mathbf{\Theta}}_k$. In addition, suppose that at the $k + 1_{th}$ step, $\widehat{\mathbf{\Theta}}_{k+1}$ could converge to $\mathbf{\Theta}^*$ as long as with a suitable λ_{k+1}.

According to **Assumption 1**, the obtained estimation of $\mathbf{\Theta}$ could be optimal if with a suitable value of λ_{k+1} and the iterative algorithm will stop when

$$\begin{aligned} sign(\widehat{\mathbf{\Theta}}_{k+1}) &= sign(\widehat{\mathbf{\Theta}}_k) \\ loc_{k+1} &= loc_k \end{aligned} \tag{11}$$

3.5. The main procedure

BAL can reduce computations by converting the weighted l_1 minimization problem (5) into several subproblems solved without third party solver. The main procedure of BAL is summarised as follows:

Algorithm BAL

1: Set $k = 0$, choose μ=λ_0, $v_0 = d_0 = \mathbf{0}$ and $G = \mathbf{I}$
2: **Repeat**
3: $\widehat{\mathbf{\Theta}}_{k+1} = \arg\min_{\mathbf{\Theta}} \frac{1}{2}\|\mathbf{P\Theta} - \mathbf{y}\|_2^2 + \frac{\mu}{2}\|G\mathbf{\Theta} - v_k - d_k\|_2^2$
4: $v_{k+1} = \arg\min_v \lambda_k\|v\|_1 + \frac{\mu}{2}\|G\widehat{\mathbf{\Theta}}_{k+1} - v - d_k\|_2^2$
5: $d_{k+1} \leftarrow d_k - (G\widehat{\mathbf{\Theta}}_{k+1} - v_{k+1})$
6: Set $\mathbf{Q}_k = \mathrm{diag}[|\widehat{\mathbf{\Theta}}_{\mathrm{k}}|]$, $\mathrm{diag}[\mathrm{w_k}] = \mathrm{G}$, $\mathbf{W}_k = \mathrm{diag}[\mathrm{w_k}]^{-1}$
$(w_{k+1})_j = \left[\mathbf{P}_j^T(\lambda_k\mathbf{I} + \mathbf{P}\mathbf{W}_k\mathbf{Q}_{k+1}\mathbf{P}^T)^{-1}\mathbf{P}_j\right]^{\frac{1}{2}}$
7: $k \leftarrow k + 1$
8: calculate λ_k according to section 3.3.
9: **until** stopping criterion (11) is satisfied.

Remark 1: During the iterations, there still might be no exact zero coefficients. Therefore, the small estimated weights, e.g. $\|\Theta_j\|_2^2 \ll \|\widehat{\mathbf{\Theta}}\|_2^2$, could also be pruned at each iteration with a predetermined threshold. This pruning procedure is also used by SBL.

4. Theoretical analysis

4.1. The existence of solution

The cost function of BAL is

$$L_\mu(\mathbf{\Theta}, v, u) = f_1(\mathbf{\Theta}) + f_2(v) - u^T(G\mathbf{\Theta} - v) + \frac{\mu}{2} \parallel G\mathbf{\Theta} - v \parallel_2^2$$

To analysis the existence of solution, we turn to discuss the alternative format of the problem (9), namely

$$L_\mu(\mathbf{\Theta}, v, d) = f_1(\mathbf{\Theta}) + f_2(v) + \frac{\mu}{2}\|G\mathbf{\Theta} - v - d\|_2^2$$

where $f_1(\mathbf{\Theta}) = \frac{1}{2}\|\mathbf{y}-\mathbf{P\Theta}\|_2^2$ and $f_2(v) = \lambda\|v\|_1$. Assume G, v, d are bounded, therefore one can alternatively consider another constrained form such that

$$\min_{\mathbf{\Theta}} \{\frac{1}{2}\|\mathbf{y}-\mathbf{P\Theta}\|_2^2 + \lambda\|G\mathbf{\Theta}\|_1\} \quad s.t \;\; \|G\mathbf{\Theta} - v - d\|_2^2 \leq R \tag{12}$$

for some radius $R > 0$. Since G is a diagonal matrix with each element being positive,the equation (12) can be rewritten as

$$\min_{\mathbf{\Theta}} \{\frac{1}{2}\|\mathbf{y}-\mathbf{P\Theta}\|_2^2 + \lambda\|G\mathbf{\Theta}\|_1\} \quad s.t \;\; \|\mathbf{\Theta}\|_2^2 \leq R^{'} \tag{13}$$

where radius $R^{'} > 0$. According to boundedness theorems, maximum and minimum theorems, the optimal solution to equation (13) exists, since $\frac{1}{2}\|\mathbf{y}-\mathbf{P\Theta}\|_2^2+\lambda\|G\mathbf{\Theta}\|_1$ is convex [6]. Therefore, according to Lagrangian duality theory, the problem (9) exists optimal solution.

4.2. The uniqueness of solution

The problem (9) can be simplified as the weighted l_1 minimization problem with the assumption that $v_k = G\widehat{\mathbf{\Theta}}_{k+1}$. Therefore, under this condition, we could directly discuss the theoretical properties of the weighted l_1 minimization problem (5). We first consider the Lasso problem

$$\widehat{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta}} \{\frac{1}{2}\|\mathbf{y}-\mathbf{P\Theta}\|_2^2 + \lambda\|\mathbf{\Theta}\|_1\} \tag{14}$$

Firstly , we define a support set $S(\mathbf{\Theta}) = \{i|\mathbf{\Theta}_i \neq 0\}$ and cardinality $k = |S(\mathbf{\Theta})|$ which means the number of nonzero coefficients in $\mathbf{\Theta}$ with $k < N$. In addition, the objective function of Lasso problem is not differentiable, because l_1 penalty is actually a piecewise linear function. Therefore, we apply zero subgradient condition to solve the optimal solution of Lasso type problem [11]. Suppose vector $z \in R^M$ is a subgradient for l_1 norm estimated at $\mathbf{\Theta} \in R^M$, if it satisfies

$$\begin{cases} z_i = sign(\mathbf{\Theta}_i), & if \;\; \mathbf{\Theta}_i \neq 0 \\ z_i \in [-1, 1], & if \;\; \mathbf{\Theta}_i = 0 \end{cases} \tag{15}$$

Under these definitions, we can start the following discussions.

Lemma 1: The solution uniqueness of Lasso problem [11]

1. Vector $\widehat{\mathbf{\Theta}} \in R^M$ is an optimal solution of the problem (14) if and only if there exists a subgradient vector z which satisfies $\mathbf{P}^T\mathbf{P}(\widehat{\mathbf{\Theta}} - \mathbf{\Theta}^*) - \mathbf{P}^T\xi + \lambda z = 0$.
2. Assume that the subgradient vector z satisfies strict dual feasibility condition $|z_j| < 1, \forall j \notin S(\widehat{\mathbf{\Theta}})$. Then any optimal solution $\mathbf{\Theta}^*$ to Lasso satisfies $\mathbf{\Theta}_j^* = 0, \forall j \notin S(\widehat{\mathbf{\Theta}})$.
3. With the conditions of part (2), if $k \times k$ matrix $\mathbf{P}_{S(\widehat{\mathbf{\Theta}})}^T\mathbf{P}_{S(\widehat{\mathbf{\Theta}})}$ is invertible, then the optimal solution $\widehat{\mathbf{\Theta}}$ of Lasso problem is unique.

The proof of **Lemma 1** for Lasso problem has been given in the literature [11]. Here, we tend to prove the solution of the problem (5) is unique based on this lemma, since the subgradient of $\|G\mathbf{\Theta}\|_1$ can also be

written as the form of equation (15)

$$\begin{cases} \widetilde{z}_i = sign(w_i\boldsymbol{\Theta}_i), & if \quad w_i\boldsymbol{\Theta}_i \neq 0 \\ \widetilde{z}_i \in [-1,1], & if \quad w_i\boldsymbol{\Theta}_i = 0 \end{cases} \tag{16}$$

where w_i is denoted as the i_{th} diagonal element of the matrix G. In addition, the problem (5) can be rewritten as

$$\min_{G\boldsymbol{\Theta}\in R^M} \|G\boldsymbol{\Theta}\|_1 \quad s.t. \;\; \mathbf{P}G^{-1}G\boldsymbol{\Theta} = \mathbf{y} \tag{17}$$

Before giving the proof of the unique solution of the problem (17), we first give the following lemma.

Lemma 2: The solution uniqueness of weighted l_1 minimization problem

1. Vector $\widehat{\boldsymbol{\Theta}} \in R^M$ is an optimal solution of the problem (17) if and only if there exists a subgradient vector $\widetilde{z}$ which satisfies $\mathbf{P}^T\mathbf{P}(\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^*) - \mathbf{P}^T\xi + \lambda\widetilde{z} = 0$.
2. Assume that the subgradient vector z satisfies strict dual feasibility condition $|\widetilde{z}_j| < 1, \forall j \notin S(\widehat{\boldsymbol{\Theta}})$. Then any optimal solution $\boldsymbol{\Theta}^*$ to Lasso satisfies $\boldsymbol{\Theta}^*_j = 0, \forall j \notin S(\widehat{\boldsymbol{\Theta}})$.
3. With the conditions of part (2), if $k \times k$ matrix $\mathbf{P}^T_{S(\widehat{\boldsymbol{\Theta}})}\mathbf{P}_{S(\widehat{\boldsymbol{\Theta}})}$ is invertible, then the optimal solution $\widehat{\boldsymbol{\Theta}}$ of the weighted l_1 minimization problem is unique.

Proof of Lemma 2: Since $w_1, w_2, \cdots, w_M$ are positive coefficients, the problem (5), namely $\frac{1}{2}\|\mathbf{y} - \mathbf{P}\boldsymbol{\Theta}\|_2^2 + \lambda\sum_{i=1}^M w_i|\Theta_i|$ is a convex. Here w_i also represents the i_{th} diagonal element of the matrix G. According to standard optimal conditions in convex program, $\widehat{\boldsymbol{\Theta}}$ is an optimal solution for problem (5) if and only if $\mathbf{P}^T\mathbf{P}\widehat{\boldsymbol{\Theta}} - \mathbf{P}^T\mathbf{y} + \lambda\widetilde{z} = 0$ with the subgradient $\widetilde{z} \in \partial\|G\boldsymbol{\Theta}\|_1$. Meanwhile, $\mathbf{y} = \mathbf{P}\boldsymbol{\Theta}^* + \xi$, so the solution of the problem (5) satisfies condition (1) of **Lemma 2**. Next, according to duality theory [6], the optimal solution of the Lasso problem must satisfy $\widetilde{z}^T\boldsymbol{\Theta}^* = \|\boldsymbol{\Theta}^*\|_1$, which can established if and only if $\boldsymbol{\Theta}^*_j = 0$ for all j such that $|\widetilde{z}_j| < 1$. The solution of the weighted l_1 minimization problem still keeps a similar condition as follows. At the beginning, we prove that the conjugate of $f_0(G\boldsymbol{\Theta}) = \|G\boldsymbol{\Theta}\|_1$ with $G\boldsymbol{\Theta}$ a new variable satisfies

$$f_0^*(\widetilde{z}) = \begin{cases} 0, & \|\widetilde{z}\|_{1*} \leq 1 \\ \infty, & \textbf{otherwise} \end{cases} \tag{18}$$

with $\|\cdot\|_{1*}$ being dual norm of $\|\cdot\|_1$. If $\|\widetilde{z}\|_{1*} > 1$, then according to dual norm, there exists $s \in R^M$ with $\|s\|_1 \leq 1$ and $\widetilde{z}^T s > 1$. If choosing $G\boldsymbol{\Theta} = ts$ and $t \to \infty$, we have

$$\widetilde{z}^T G\boldsymbol{\Theta} - \|G\boldsymbol{\Theta}\| = t\widetilde{z}^T s - \|ts\|_1 \leq t(\widetilde{z}^T s - \|s\|_1)$$

Therefore, $f_0^*(\widetilde{z}) = \widetilde{z}^T G\boldsymbol{\Theta} - \|G\boldsymbol{\Theta}\|_1 \to \infty$. Conversely, when $\|\widetilde{z}\|_{1*} \leq 1$, we have [6]

$$\widetilde{z}^T G\boldsymbol{\Theta} - \|G\boldsymbol{\Theta}\|_1 \leq 0$$

Therefore, $f_0^*(\widetilde{z}) = \widetilde{z}^T G\boldsymbol{\Theta} - \|G\boldsymbol{\Theta}\|_1$ can be maximized with $\boldsymbol{\Theta} = 0$.

The dual function for problem (17) can be described as

$$g(\tau) = \inf_{G\boldsymbol{\Theta}} \left(f_0(G\boldsymbol{\Theta}) + \tau^T(\mathbf{P}G^{-1}G\boldsymbol{\Theta} - \mathbf{y})\right)$$

$$= -\mathbf{y}^T\tau - f_0^*(-(\mathbf{P}G^{-1})^T\tau)$$

where τ is the vector of Lagrangian multipliers τ_i. Using the result of (18), the dual function $g(\tau)$ given by

$$g(\tau) = \begin{cases} -\mathbf{y}^T\tau, & \|(\mathbf{P}G^{-1})^T\tau)\|_{1*} \leq 1 \\ -\infty, & \textbf{otherwise} \end{cases}$$

It means that the optimal solution of the problem (5) satisfies $\widetilde{z}^T G\mathbf{\Theta}^* = \|G\mathbf{\Theta}^*\|_1$ if and only if $\mathbf{\Theta}_j^* = 0$ for all j such that $|\widetilde{z}_j| < 1$. Therefore, the solution of the problem (5) satisfies condition (2) of **Lemma 2**. Lastly, since $\widehat{\mathbf{\Theta}}_{k+1}$ can be rewritten as $(\mathbf{\Theta}_{S(\widehat{\mathbf{\Theta}}_{k+1})}, \mathbf{0})$, then we have

$$\mathbf{P}\mathbf{\Theta} = \mathbf{P}_{S(\widehat{\mathbf{\Theta}}_{k+1})}\mathbf{\Theta}_{S(\widehat{\mathbf{\Theta}}_{k+1})}$$

In addition, the columns of matrix $\mathbf{P}_{S(\widehat{\mathbf{\Theta}}_{k+1})}$ are independent, so we can get the conclusion that matrix $\mathbf{P}_{S(\widehat{\mathbf{\Theta}}_{k+1})}$ is full column rank. Therefore, $\mathbf{P}^T_{S(\widehat{\mathbf{\Theta}})}\mathbf{P}_{S(\widehat{\mathbf{\Theta}})}$ is positive definite since

$$R(\mathbf{P}^T_{S(\widehat{\mathbf{\Theta}}_{k+1})}\mathbf{P}_{S(\widehat{\mathbf{\Theta}}_{k+1})}) = R(\mathbf{P}_{S(\widehat{\mathbf{\Theta}}_{k+1})}) = k$$

with $k < N$.

As mentioned above, the optimal solution of the regression problem (5) satisfies all these three conditions in **Lemma 2** and **Lemma 1** is a special case of **Lemma 2** with $G = \mathbf{I}$. Therefore, similar with the problem (14), the solution of the problem (5) is unique.

4.3. The convergence of algorithm

The convergence of BAL can be guaranteed based on the theorem proposed by [12].

Theorem 1 [12]: Consider the problem (7), where f_1 and f_2 are closed, proper convex functions, and $G \in R^{M\times M}$ has full column rank. Consider arbitrary $\mu > 0$ and $v_0, d_0 \in R^M$. Let $\{\eta_k \geq 0, k = 0, 1, \cdots \infty\}$ and $\{\nu_k \geq 0, k = 0, 1, \cdots \infty\}$ be two sequences such that

$$\sum_{k=0}^{\infty} \eta_k < \infty \quad and \quad \sum_{k=0}^{\infty} \nu_k < \infty$$

Consider three sequences $\{\widehat{\mathbf{\Theta}}_k \in R^M, k = 0, 1, \cdots\}$, $\{v_k \in R^M, k = 0, 1, \cdots\}$ and $\{d_k \in R^M, k = 0, 1, \cdots\}$ that satisfy

$$\eta_k \geq \left\| \widehat{\mathbf{\Theta}}_{k+1} - \arg\min_{\mathbf{\Theta}} f_1(\mathbf{\Theta}) + \tfrac{\mu}{2}\|G\mathbf{\Theta} - v_k - d_k\|_2^2 \right\|$$
$$\nu_k \geq \left\| v_{k+1} - \arg\min_v f_2(v) + \tfrac{\mu}{2}\|G\widehat{\mathbf{\Theta}}_{k+1} - v - d_k\|_2^2 \right\|$$
$$d_{k+1} = d_k - (G\widehat{\mathbf{\Theta}}_{k+1} - v_{k+1})$$

If problem (7) has a solution, the sequence $\{\widehat{\mathbf{\Theta}}_k\}$ converges, namely, $\widehat{\mathbf{\Theta}}_k \to \mathbf{\Theta}^*$, where $\mathbf{\Theta}^*$ is a solution of (7). If there does not exist a solution for (7), then at least one of $\{v_k\}$ or $\{d_k\}$ diverges.

The convergence of SALSA has been proved with **Theorem 1**, for more details, please review the literature [10]. It is worth noting that the matrix G in BAL is iteratively calculated from the Bayesian viewpoint.

However, we still can prove the proposed method is convergent.

Theorem 2: If the subproblems (10)

$$\widehat{\boldsymbol{\Theta}}_{k+1} = \arg\min_{\boldsymbol{\Theta}} f_1(\boldsymbol{\Theta}) + \frac{\mu}{2}\|G\boldsymbol{\Theta} - v_k - d_k\|_2^2 \tag{10.a}$$

and

$$v_{k+1} = \arg\min_{v} f_2(v) + \frac{\mu}{2}\|G\widehat{\boldsymbol{\Theta}}_{k+1} - v - d_k\|_2^2 \tag{10.b}$$

can be solved exactly and G is full-column-rank, then the convergence of BAL can be guaranteed.

Proof of Theorem 2 : The reweighted matrix $G = \mathrm{diag}\big[\mathbf{P}^{\mathrm{T}}(\lambda\mathbf{I} + \mathbf{P}\boldsymbol{\Gamma}_{\mathrm{k}}\mathbf{P}^{\mathrm{T}})^{-1}\mathbf{P}\big]^{\frac{1}{2}}$ is calculated during each iterative step. It is worth pointing out that G is a diagonal matrix and each diagonal element is positive. Therefore, matrix G is full-column-rank. In addition, $f_1(\boldsymbol{\Theta}) = \frac{1}{2}\|\mathbf{P}\boldsymbol{\Theta} - \mathbf{y}\|_2^2$ and $f_2(v) = \lambda\|v\|_1$, so the minimizations of subproblems (10.a) and (10.b) can be solved exactly. Specifically, the cost function of the problem (10.a) can be represented as

$$J(\boldsymbol{\Theta}) = \frac{1}{2}(\mathbf{P}\boldsymbol{\Theta} - \mathbf{y})^T(\mathbf{P}\boldsymbol{\Theta} - \mathbf{y}) + \frac{\mu}{2}(G\boldsymbol{\Theta} - v_k - d_k)^T(G\boldsymbol{\Theta} - v_k - d_k) \tag{19}$$

where the equation (19) is a quadratic function and is differentiable while the equation (10.a) is not differentiable [6]. The solution of that optimization problem (19) is optimal if and only if the derivative of J is equal to zero, namely

$$\nabla J(\boldsymbol{\Theta}) = \mu(G^TG\boldsymbol{\Theta} - G^T(v_k + d_k)) + \mathbf{P}^T\mathbf{P}\boldsymbol{\Theta} - \mathbf{P}^T\mathbf{y} = 0 \tag{20}$$

By simplifying (20), the solution can be calculated as

$$\widehat{\boldsymbol{\Theta}}_{k+1} = (\mathbf{P}^T\mathbf{P} + \mu G^TG)^{-1}\big(\mathbf{P}^T\mathbf{y} + \mu G^T(v_k + d_k)\big)$$

Before giving the specific solution of problem (10.b), similar to SBL, we also use the soft thresholding operator $S_{\mu/\lambda}$ defined as follows [7]:

$$S_{\mu/\lambda}(x) = \max(0, x - \mu/\lambda) - \max(0, -x - \mu/\lambda)$$

where λ is the penalty parameter. Based on this function, the solution of equation (10.b) has the following format

$$v_{k+1} = \max\big(0, (G\widehat{\boldsymbol{\Theta}}_{k+1} - d_k) - \mu/\lambda_k\big) - \max\big(0, -(G\widehat{\boldsymbol{\Theta}}_{k+1} - d_k) - \mu/\lambda_k\big)$$

We have shown the two subproblems (10.a) and (10.b) have exact solutions and the weighted matrix G is full-column-rank, according to the proof of the convergence of SALSA, BAL is guaranteed to converge.

5. Simulation

Consider the nonlinear benchmark example [13]:

$$z(t) = 0.2z^3(t-1) + 0.7z(t-1)u(t-1)$$

$$
\begin{aligned}
&+ 0.6u^2(t-2) - 0.5z(t-2) \\
&- 0.7z(t-2)u^2(t-2) \\
y(t) &= z(t) + e(t). \qquad (21)
\end{aligned}
$$

where $u(t)$ and $z(t)$ are the system input and output at interval t, respectively. The system is excited with a uniformly distributed white noise u(t) $\in$ [-1,1]. The system output $z(t)$ is disturbed by a Gaussian noise sequence $e(t)$ with the signal-to-noise rate (SNR) $15dB$. The delayed input and output $\{z(t-1), z(t-2), z(t-3), z(t-4), u(t-1), u(t-2), u(t-3)\}$ of the unknown nonlinear system are used as model input. 2000 samples are used as training data and there are another 1000 samples for testing data. Mean square error (MSE) is used to test the model performance with

$$MSE = \frac{1}{N}\sum_{t=1}^{N}\left(y(t) - \widehat{y}(t)\right)^2$$

where $\widehat{y}(t)$ is the prediction of the unknown system.

To show the efficiency of BAL on structure determination for NARX model, several popular methods are used for comparison. The first method is OFR algorithm which belongs to forward selection method. The error reduction ratio (ERR) is a popular criterion for model selection. According to ERR criterion, the term with largest ERR value is firstly selected into the model at a time until a stopping criterion is satisfied [9]. The selection procedure generally stops at k step if it satisfies

$$1 - \sum_{i=1}^{k} ERR_i < \rho$$

where ρ is predetermined. ρ should be carefully tuned since it is related to noise and has a critical effect on selecting terms [14]. The second method is Lasso which is an effective method to obtain a sparse solution. 100 different regularized parameters are used to produce 100 models based on 5-folder cross validation (CV) scheme and the best model is determined as the final model. The third method is SBL recently presented by the literature [4], while the solution is calculated by using CVX solver to directly address the original optimization problem (5). The last one is SALSA which is a distributed algorithm. It should be noted that during the iterations of SALSA, there might also be no exact zero coefficients. Therefore, if we do not prune those small weights, it will be hard to obtain a sparse solution by directly using SALSA. For fair comparison, we determine the same stopping criterion for both SALSA and BAL.

For convenience, define λ_L, λ_{SB}, λ_B and λ_{SA} are the pre-determined parameter for Lasso, SBL, BAL and SALSA, respectively. Since different values of λ_L, λ_{SB}, λ_B and λ_{SA} may lead to different solutions, so for fair comparison, we repeat SBL, OFR, SALSA and BAL many times and choose the best one as the final model. It is worth pointing out that we define $\mu = \lambda_B$ in BAL. All the test results for are listed in Table 1.

From Table 1, we can get several conclusions. First, as long as with a suitable predetermined parameter, BAL, SALSA, OFR, Lasso and SBL all can have a satisfied test performance with MSE being about 0.003. Second, since all the important terms have been emphasised, therefore, one can obviously see that OFR

Table 1: The simulation results for the example 1

Algorithm	Selected Terms	Coefficient	Error	Steps	Time
	$\mathbf{z(t-2)}$	-0.5025			
SALSA	$\mathbf{u(t-1)z(t-1)}$	0.6872			
	$\mathbf{u(t-2)u(t-2)}$	0.5984	0.0034	3	1.36s
	$\mathbf{z(t-2)u(t-2)^2}$	-0.6490			
	$\mathbf{z(t-1)^3}$	0.1589			
	other 115 terms	$\vdots$			
	$z(t-4)u(t-2)^2$	-0.0191			
	$\mathbf{u(t-2)u(t-2)}$	0.6034			
OFR	$\mathbf{z(t-2)}$	-0.4944	0.0032	-	1.41s
	$\mathbf{u(t-1)z(t-1)}$	0.6876			
	$\mathbf{z(t-2)u(t-2)^2}$	-0.7150			
	$\mathbf{z(t-1)^3}$	0.1932			
	$\mathbf{z(t-2)}$	-0.4898			
Lasso	$\mathbf{u(t-1)z(t-1)}$	0.6634			
	$\mathbf{u(t-2)u(t-2)}$	0.5856	0.0034	-	4.13s
	$\mathbf{z(t-2)u(t-2)^2}$	-0.6683			
	$\mathbf{z(t-1)^3}$	0.1754			
	$\mathbf{z(t-2)}$	-0.4983			
SBL	$\mathbf{u(t-1)z(t-1)}$	0.6881			
	$\mathbf{u(t-2)u(t-2)}$	0.5979	0.0033	5	21.7s
	$\mathbf{z(t-2)u(t-2)^2}$	-0.6800			
	$\mathbf{z(t-1)^3}$	0.1812			
	$\mathbf{z(t-2)}$	-0.4922			
BAL	$\mathbf{u(t-1)z(t-1)}$	0.6862			
	$\mathbf{u(t-2)u(t-2)}$	0.6008	0.0032	6	2.34s
	$\mathbf{z(t-2)u(t-2)^2}$	-0.6996			
	$\mathbf{z(t-1)^3}$	0.1951			

The parameters are determined as $\lambda_L = 0.0044$, $\lambda_{SA} = 0.084$, $\lambda_B = 0.1$, $\lambda_{SB} = 0.4$ and $\rho = 0.03$, respectively.

selects a redundant term into final model. The redundant term $z(t-4)u(t-2)^2$ tends to be firstly selected into the final model according to ERR criterion since it has the largest ERR value. This leads to the fact that no mater how to tune ρ, the final model always includes $z(t-4)u(t-2)^2$. In addition,the estimation of important terms made by SALSA is similar with true values. However, it can not build a sparse model even if most values of other selected terms are small. Other three methods based on l_1 regularization technique obtain an optimal model without redundant terms as long as with a suitable tuning parameter, although the estimations of parameters made by Lasso are not as accurate as that made by BAL and SBL. Third, from the table, one can see that algorithms have different running time. The running time of SALSA and OFR is less than others due to the computation efficient. Among the reweighted l_1 methods, BAL can obtain a satisfied model faster from the raw data alone. The iterative step of BAL is larger than SBL, however, the running time is less than that of SBL. The reason is that the solution of suboptimization problem (10.a) and (10.b) can be solved exactly, therefore BAL does not need any solvers (e.g. CVX).

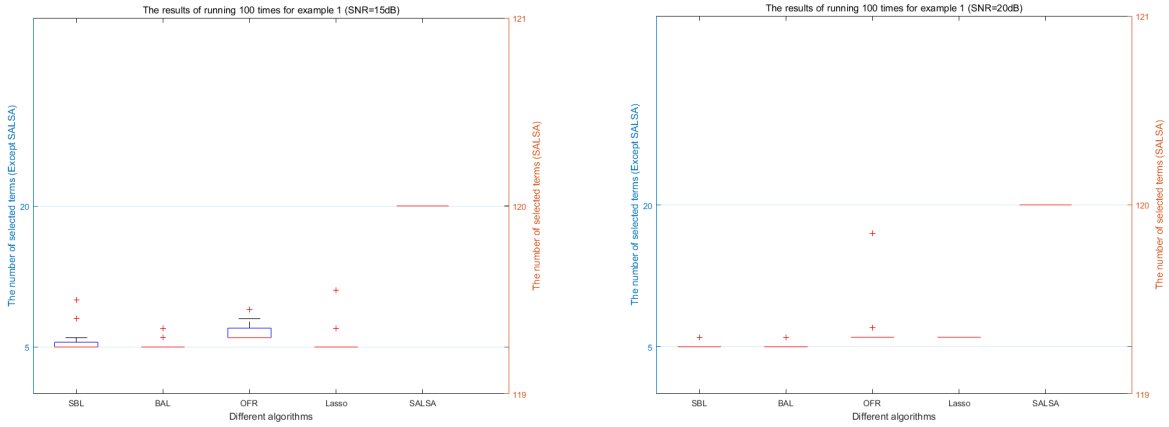

Figure 1: Box plots of the number of model terms produced by five methods for example 1.

Meanwhile, the Box plots of the number of model terms generated from Monte Carlo simulation with 100 repetitions are shown in Figure 1, which are used to consider the sensitivity of algorithm to noise (level). To make simulation results more readable, the plots are drawn in the form of two different y axes. From these two figures, one can see that most algorithms performed better when noise with a larger SNR. Meanwhile, the proposed BAL method could obtain a more parsimonious model in most cases comparing with other algorithms. In addition, the original SALSA can not build a sparse model although most variables have small weights.

5.1. Example 2

Consider the following sparse nonlinear system [15]

$$\begin{aligned} z(t) = &-0.3u(t-2) + 0.8z(t-1) + u(t-1) \\ &- 0.4u(t-3) + 0.25u(t-1)u(t-2) \end{aligned}$$

$$
\begin{aligned}
&- 0.3u^3(t-1) + 0.24u^3(t-2) \\
&- 0.2u(t-2)u(t-3) \\
y(t) &= z(t) + e(t). \qquad (22)
\end{aligned}
$$

where $u(t)$ and $z(t)$ are the system input and output at interval t, respectively. A uniformly distributed white noise $u(t) \in [-1, 1]$ is used to excite the nonlinear system above and the system with noise being SNR 15dB. 4000 samples are generated for system identification, 25 percent samples are used for testing data and others for training data. In addition, the delayed input and output $\{z(t-1), z(t-2), u(t-1), u(t-2), u(t-3)\}$ are used for model input, which means there are total 56 polynomial terms.

We repeated each algorithm 100 times and choose the optimal model as final results. All the simulation results are listed in Table 2. From the table, one can see that the performance of these algorithms are similar with test error being about 0.01 (MSE). In addition, the solution of OFR, SALSA and Lasso is suboptimal since there are redundant terms in the final model. The simulation results of Lasso are not satisfied since Lasso obtain a model with many redundant terms, leading to the estimation of coefficients of important terms is not as accurate as that made by other algorithms. Meanwhile, one can see that SALSA can not produce a sparse solution even though most values of other terms are small. BAL and SBL can obtain an optimal model without unimportant terms, leading to a more parsimonious model. And the running time of SALSA and OFR is less than other methods. Next comes BAL with running time being 3.99s while the time of SBL is 14.0s. As discussed above, the novel algorithm has the ability to give a more satisfied solution with less or no redundant terms and obtain more accurate estimation of parameters. In addition, BAL solves the original optimization problem by converting into several suboptimization problem, therefore the running time can be saved.

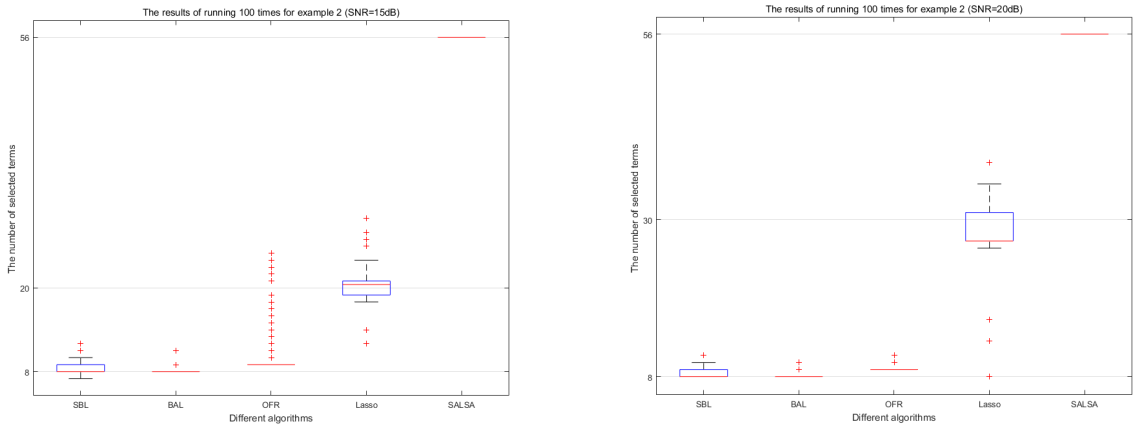

Figure 2: Box plots of the number of model terms produced by five methods for example 2.

Again, the Box plots of the number of model terms generated from Monte Carlo simulation with 100 repetitions are shown in Figure 2. One can see that in most cases, BAL could build a more compact model and other algorithms may often select redundant terms into the final model. Meanwhile, as the value of SNR

Table 2: The simulation results for example 2

Algorithm	Selected Terms	Coefficient	Error	Steps	Time
SALSA	$\mathbf{u(t-1)}$	1.0055	0.0100	3	0.65s
	$\mathbf{u(t-2)}$	-0.3061			
	$\mathbf{u(t-3)}$	-0.4025			
	$\mathbf{z(t-1)}$	0.7908			
	$\mathbf{u(t-1)u(t-2)}$	0.2797			
	$\mathbf{u(t-2)u(t-3)}$	-0.1609			
	$\mathbf{u(t-1)^3}$	-0.3148			
	$\mathbf{u(t-2)^3}$	0.1749			
	other 48 terms	$\vdots$			
OFR	$\mathbf{u(t-1)}$	1.0092	0.0099	-	0.63s
	$\mathbf{u(t-2)}$	-0.3832			
	$\mathbf{u(t-1)u(t-2)}$	0.2475			
	$\mathbf{u(t-1)^3}$	-0.3164			
	$\mathbf{u(t-2)u(t-3)}$	-0.2067			
	$u(t-3)^3$	-0.0111			
	$\mathbf{z(t-1)}$	0.8722			
	$\mathbf{u(t-3)}$	-0.4337			
	$\mathbf{u(t-2)^3}$	0.2799			
Lasso	$\mathbf{u(t-1)}$	0.9535	0.0108	-	10.6s
	$\mathbf{u(t-2)}$	0.4777			
	$\mathbf{z(t-1)}$	0.0046			
	$\mathbf{u(t-1)u(t-2)}$	0.2440			
	$\mathbf{u(t-1)^3}$	-0.2340			
	other 6 terms	$\vdots$			
SBL	$\mathbf{u(t-1)}$	1.0084	0.0099	5	14.0s
	$\mathbf{u(t-2)}$	-0.2854			
	$\mathbf{u(t-3)}$	-0.3925			
	$\mathbf{z(t-1)}$	0.7761			
	$\mathbf{u(t-1)u(t-2)}$	0.2472			
	$\mathbf{u(t-2)u(t-3)}$	-0.1818			
	$\mathbf{u(t-1)^3}$	-0.3153			
	$\mathbf{u(t-2)^3}$	0.2481			
BAL	$\mathbf{u(t-1)}$	1.0093	0.0100	6	3.99s
	$\mathbf{u(t-2)}$	-0.3082			
	$\mathbf{u(t-3)}$	-0.3991			
	$\mathbf{z(t-1)}$	0.7958			
	$\mathbf{u(t-1)u(t-2)}$	0.2479			
	$\mathbf{u(t-2)u(t-3)}$	-0.1695			
	$\mathbf{u(t-1)^3}$	-0.3160			
	$\mathbf{u(t-2)^3}$	0.2613			

The parameters are determined as $\lambda_L = 0.0026$, $\lambda_B = 0.043$, $\lambda_{SB} = 0.03$, $\lambda_{SA} = 0.0385$ and $\rho = 0.031$, respectively.

increasing, most algorithms performed better since the decreasing noise level makes variable selection get easier. According to discussions aforementioned, the effectiveness of BAL has been demonstrated.

6. Conclusion

In this paper, we have proposed a Bayesian Augmented Lagrangian (BAL) method to solve the weighted l_1 minimization problem by converting the original optimization problem into several subproblems. The reweighted matrix can be iteratively calculated from Bayesian viewpoint rather than setting as identity matrix used in conventional methods, leading to a sparse model with fewer or no redundant terms. Theoretical proof regarding to solution existence, uniqueness, algorithm convergence has been given. The simulation results show that BAL is able to build a compact model with less running time compared with other reweighted l_1 methods and also keeps a satisfied model performance.

7. Acknowledgment

The authors would like to thank Prof Ye Yuan from the Huazhong University of Science and Technology for his advice on Bayesian methods and constructive comments.

References

[1] Guo Y., Guo L.Z., Billings S.A., Wei H.L. (2015). An Iterative Orthogonal Forward Regression Algorithm. *Internation Journal of Systems Science*, 46(5): 776–789.

[2] F. He, H.L. Wei, S.A. Billings, Identification and frequency domain analysis of non-stationary and nonlinear systems using time-varying NARMAX models, *International Journal of Systems Science*, 46(11): 2087–2100, 2015.

[3] Zhang L., Li K. (2015). Forward and backward least angle regression for nonlinear system identification. *Automatica*, 53(6): 94-102.

[4] Pan W., Yuan Y., Goncalves J., Stan G.B. (2015). A Sparse Bayesian Approach to the Identification of Nonlinear State-Space Systems. *IEEE Trasactions on Automatic Control*, 61(1): 182-187.

[5] Candes E., Wakin M., Boyd S. (2008). Enhancing sparsity by reweighted l_1 minimization. *Journal of Fourier Analysis & Applications*, 14(5-6): 877-905.

[6] Boyd S., Vandenberghe L.(2004). Convex Optimization. *Cambridge University Press.*

[7] Pan W., Sootla A., Stan G.B. (2014). Distributed Reconstruction of Nonlinear Networks: An ADMM Approach. *IFAC Proceedings Volumes*, 47(3): 3208-3213.

[8] Chen S., Billings S.A., Luo W. (1989). Orthogonal least squares methods and their application to non-linear system identification. *International Journal of Control*, 50: 1873-1896.

[9] Zhang L., Li K., Bai E.W., Irwin G.W. (2015). Two-Stage Orthogonal Least Squares Methods for Neural Network Construction. *IEEE Transactions on Neural Networks & Learning Systems*, 26(8): 1608.

[10] Afonso M.V., Bioucas-Dias J.M., and Figueiredo M.A.T. (2010). Fast image recovery using variable splitting and constrained optimization. *IEEE Transactions on Image Processing*, 19(9): 2345-2356.

[11] Wainwright M.J.(2009). Sharp Thresholds for High-Dimensional and Noisy Sparsity Recovery Using-Constrained Quadratic Programming (Lasso). *IEEE Transactions on Information Theory*, 55(5): 2183-2202.

[12] Eckstein J., Bertsekas D. (1992). On the DouglasCRachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 5: 293-318.

[13] Mao K.Z., Billings S.A.(2003). Algorithms for minimal model structure detection in nonlinear dynamic system identification. *International Journal of Control*, 68(2): 311-330.

[14] Tang X., Zhang L.(2018). Stability orthogonal regression for system identification. *Systems & Control Letters*, 117: 30-36.

[15] Piroddi L., Spinelli W. (2015). An identification algorithm for polynomial NARX models based on simulation error minimization. *International Journal of Control*, 76(17): 1767-1781.