2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Minimax Two-Stage Gradient Boosting for Parameter Estimation

Braghadeesh Lakshminarayanan and Cristian R. Rojas

Abstract— **Parameter estimation is an important sub-field in statistics and system identification. Various methods for parameter estimation have been proposed in the literature, among which the Two-Stage (TS) approach is particularly promising, due to its ease of implementation and reliable estimates. Among the different statistical frameworks used to derive TS estimators, the min-max framework is attractive due to its mild dependence on prior knowledge about the parameters to be estimated. However, the existing implementation of the minimax TS approach has currently limited applicability, due to its heavy computational load. In this paper, we overcome this difficulty by using a gradient boosting machine (GBM) in the second stage of TS approach. We call the resulting algorithm the Two-Stage Gradient Boosting Machine (TSGBM) estimator. Finally, we test our proposed TSGBM estimator on several numerical examples including models of dynamical systems.**

Index Terms— **Two-Stage approach, estimation theory, statistical decision theory, Gradient Boosting.**

I. Introduction

Most physical and economical systems can be represented by mathematical models involving unknown parameters that need to be estimated. Examples include spring-mass systems, which are modelled by second-order differential equations involving mass and spring constants, and financial systems that can be approximated by the Black-Scholes formula, depending on unknown drift and volatility parameters. To understand and simulate the behaviour of such systems, practitioners estimate these unknown parameters, which constitutes one of the important problems in system identification.

To estimate the unknown parameters of given physical system, or data generating mechanism, various estimation procedures have been developed, such as Maximum Likelihood (ML) [1–3], Instrument Variables (IV) [4], and Prediction Error Methods (PEM) [5]. These methods may achieve statistical consistency [2], however can suffer from implementation drawbacks: (i) they require user specified initialization, and (ii) the final estimation problem is often a non-convex optimization program whose solution may get stuck in local optima. For this purpose, alternative estimation procedures such as Indirect Inference [6], the Method of Simulated Moments [7] and the Two-Stage (TS) approach [8, 9], have been proposed in the literature.

Among the alternative methods for parameter estimation, the TS approach has several advantages: (i) no explicit likelihood computation is needed, and (ii) it can be posed as a simple convex optimization program by carefully designing the second stage, for which minimization algorithms do not get trapped into local optima. The TS approach is based on a supervised learning approach, where one simulates a large number of observations corresponding to different values of the unknown parameters, and these observations are used to train a supervised learning algorithm to predict the true values of unknown parameter. The TS approach consists of two stages: (i) a data compression stage and (ii) a function approximation stage. In the data compression stage (first stage), a large number of observations is compressed into a smaller set of samples, and in the second stage these compressed samples along with their corresponding values of the unknown parameters are fed as a training set to a supervised learning algorithm.

The theoretical justification of the TS approach to parameter estimation was first developed in [10]. Towards this end, its authors provided two statistical frameworks for which TS can be derived. The two frameworks considered are: (i) *Bayes* and (ii) *Minimax* estimation. The derivations of these frameworks were done for the case where the data generating mechanism creates independent and identically distributed (i.i.d.) samples, even though they can be easily extended to parameter estimation of dynamical systems where the data is non-i.i.d. Under the i.i.d. assumption, the first stage can be constructed from quantiles of the observations. It has been demonstrated via numerical example in [10] that the TS approach gives reliable Bayes and minimax estimates of the unknown parameters.

In this paper, we focus on the minimax TS framework. This framework is practically and theoretically attractive because, unlike Bayes estimators, minimax estimators are "robust" to prior knowledge. However, the minimax TS approach derived in [10] suffers from high computational time required to obtain reliable estimates, due to the fact that (i) an optimization algorithm should be run for each sample of unknown parameters, and the number of such samples is large, and (ii) most of the constraints in the epigraph formulation, used in [10] to solve the minimax problem, are inactive, which make the solver run very slow. In this paper, we save computation time by deploying a Gradient Boosted Machine (GBM) in the second stage as the supervised learning algorithm, which is a non-linear function approximator that is constructed based on decision or regression trees [11–13], and by carefully designing the cost function of GBM we solve the minimax problem more efficiently.

In particular, our contributions are:

- use a Gradient Boosted Machine (GBM) as the su-

This work has been supported by the Swedish Research Council under contract number 2016-06079 (NewLEADS) and by the Digital Futures project EXTREMUM. The authors are with the Division of Decision and Control Systems, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden (e-mails: blak@kth.se; crro@kth.se).

pervised learning algorithm in the second stage of TS approach;
- modify the evaluation metric of GBM using a soft-max approximation so that the min-max objective can be efficiently solved;
- numerically validate the minimax statistical framework on several examples, including dynamical systems where the data generation mechanism is non-i.i.d.

The rest of the paper is organized as follows:

In Section II we define the problem setup. Section III outlines the preliminaries. In Section IV, we propose the Two-Stage Gradient Boosting Machine (TSGBM) method for estimating unknown parameters, and evaluate TSGBM on numerical examples in Section V. Finally, we conclude the paper in Section VI.

II. Problem Setup

Consider a data generating mechanism as described in Figure 1. Let $M(\theta)$ be a model, or *data generating mechanism*, that generates an observation $\boldsymbol{y}$ upon receiving an input $\boldsymbol{u}$, where $\boldsymbol{u} \in \mathcal{U} \subseteq \mathbb{R}^p$ is a p-dimensional real vector, and $\boldsymbol{y} \in \mathcal{Y} \subseteq \mathbb{R}^r$ is an r-dimensional real vector. Here, $\mathcal{U}$ and $\mathcal{Y}$ are called *input space* and *observation space* respectively. $\theta \in \Theta \subseteq \mathbb{R}^d$ is an unknown parameter vector in a parameter space Θ that has an explicit influence on the distribution of the output generated by $M(\theta)$.

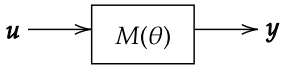

Fig. 1: A data generating mechanism $M(\theta)$. $\boldsymbol{u}$ is an input to $M(\theta)$ and $\boldsymbol{y}$ is its corresponding output.

The goal is to estimate the unknown parameter θ given a set of inputs $\{\boldsymbol{u}_i\}_{i=1}^N$ and their corresponding outputs $\{\boldsymbol{y}_i\}_{i=1}^N$. We assume that an observation $\boldsymbol{y} \in \mathcal{Y}$ has an underlying probability distribution $\mathbb{P}(\boldsymbol{y}|\theta, \boldsymbol{u})$, which is parameterized by θ and a known input $\boldsymbol{u}$. Under this assumption, the goal of estimating the unknown parameter translates to designing an *estimator* or *decision rule* $\delta \colon \mathcal{U} \times \mathcal{Y} \to \Theta$ of the true parameter θ such that the *risk*

$$R(\theta, \delta) = \mathbb{E}_{\boldsymbol{y} \sim \mathbb{P}(\cdot|\theta, \boldsymbol{u})}[L(\theta, \delta(\boldsymbol{y})] \tag{1}$$

is minimized, where $L \colon \Theta \times \Theta \to \mathbb{R}_0^{+1}$ is a *loss* function that evaluates the cost of estimating the unknown parameter as $\delta(\boldsymbol{y})$ when its true value is θ. Due to the dependence of the risk on θ, which is a priori unknown, the estimation problem can be approached in at least two different ways [1, 14], which lead, respectively, to Bayes and minimax rules.

In this paper, we focus on minimax rules, where a decision rule $\delta^*_{\text{minimax}}$ minimizes $\max_{\theta \in \Theta} R(\theta, \delta)$. That is,

$$\delta^*_{\text{minimax}} = \arg\min_{\delta \in \Delta} \max_{\theta \in \Theta} R(\theta, \delta), \tag{2}$$

[1] $\mathbb{R}_0^+$ is the set of all non-negative real numbers

III. Preliminaries

In this section we briefly revise the main concepts behind the TS approach, its minimax variant, and gradient boosting machines.

A. TS Approach

The TS approach to parameter estimation is an inverse supervise learning method where several values of the unknown parameter θ, namely $\{\theta_i\}_{i=1}^{M_\theta}$, are sampled from a probability distribution over Θ, and for each sample θ_i a large number of observations $\boldsymbol{y}_i = (y_1^{(i)}, \ldots, y_N^{(i)})^T$ is generated (see Fig **??**). The first stage of the TS approach consists in compressing the observations $\boldsymbol{y}_i$ into a smaller set of samples $\boldsymbol{\alpha}_i = (\alpha_1^{(i)}, \ldots, \alpha_n^{(i)})^T$ where $\alpha_j^{(i)} \in \mathbb{R}$ for $j \in \{1, \ldots, n\}$, and $n \ll N$. The compression is achieved via a fixed function $h \colon \mathbb{R}^N \to \mathbb{R}^n$, that is, $h(\boldsymbol{y}_i) = \boldsymbol{\alpha}_i$. In the case of i.i.d. observations, a suitable choice for $h(\boldsymbol{y}_i)$ is given by some quantiles of $\boldsymbol{y}_i$, whereas in the case of non-i.i.d. observations, $h(\boldsymbol{y}_i)$ may correspond, *e.g.*, to the coefficients of an estimated AR(n) model. This compression step aims to mitigate the effects of measure concentration; see [10] for details. Once these compressed samples are obtained, in a second stage a conventional supervised learning algorithm [15] such as Kernel regression, Deep Neural Networks (DNNs) or Gradient Boosting, is used with $\{(\boldsymbol{\alpha}_i, \theta_i)\}_{i=1}^{M_\theta}$ as the training set. That is, a non-linear function approximation is learned using a supervised learning algorithm, denoted by $g \colon \mathbb{R}^n \to \mathbb{R}^d$, which then outputs an estimate of θ.

B. Minimax TS Estimator

Now, we briefly review the minimax TS approach developed in [10]. The minimax estimator, denoted by $\delta^*_{\text{minimax}}$, is given by (2). Using the explicit expression for the risk $R(\theta, \delta)$, we obtain

$$\max_{\theta \in \Theta} R(\theta, \delta) = \max_{\theta \in \Theta} \int_{\mathcal{Y}} L\left(\theta, \delta(\boldsymbol{y})\right) \mathbb{P}(\boldsymbol{y}|\theta) d\boldsymbol{y}. \tag{3}$$

Further, the integral in the above optimization problem can be approximated by a Monte Carlo sum, by drawing several samples of observation $\boldsymbol{y}$ for a fixed θ. This gives

$$\max_{\theta \in \Theta} R(\theta, \delta) \approx \max_{\theta \in \Theta} \left[\frac{1}{M_{\mathbf{y}}} \sum_{j=1}^{M_{\mathbf{y}}} L(\theta, \delta(\mathbf{y}_{j,\theta})) \right], \tag{4}$$

where $M_{\boldsymbol{y}}$ is the number of Monte Carlo samples of $\boldsymbol{y}$ generated from $\mathbb{P}(\cdot|\theta)$, *i.e.*, $\{\boldsymbol{y}_{j,\theta}\} \overset{i.i.d.}{\sim} \mathbb{P}(\cdot|\theta)$. Θ can be potentially infinite, due to which (4) may be computationally intractable. Hence, we use scenario approach [16] to discretize Θ by sampling several values of θ using a proposal distribution s over Θ, i.e., $\theta_i \overset{i.i.d.}{\sim} s$. After discretization, (4) becomes

$$\min_{\delta \in \Delta} \max_{i=1,\ldots,M_\theta} L_i(\delta), \tag{5}$$

where $L_i(\delta) := \sum_{j=1}^{M_{\mathbf{y}}} L(\theta_i, \delta(\mathbf{y}_{ij}))$ and $\mathbf{y}_{ij} := \mathbf{y}_{j,\theta_i}$. Using an epigraph formulation [17, page 134], we obtain the final

approximate optimization problem as

$$\begin{aligned} \min_{\delta\in\Delta, t\in\mathbb{R}} \quad & t \\ \text{s.t.} \quad & L_i(\delta) \le t, \quad i = 1, \dots, M_\theta. \end{aligned} \tag{6}$$

Remark 3.1: While minimax TS was proposed in [10] for i.i.d. data generating mechanisms, this approach can also be applied when the data is non-i.i.d. Taking a closer look at (4), the Monte Carlo samples are in fact i.i.d., *i.e.*, $\{\boldsymbol{y}_{j,\theta}\} \overset{i.i.d.}{\sim} \mathbb{P}(\cdot|\theta)$, although each sequence $\boldsymbol{y}_{j,\theta} = (y_{j,1}^{(\theta)}, \dots, y_{j,N}^{(\theta)})^T$ is non-i.i.d., that is, the k^{th} observation $y_{j,k}^{(\theta)}$ may depend on the the past $k-1$ values $y_{j,1}^{(\theta)}, \dots, y_{j,k-1}^{(\theta)}$.

C. Gradient Boosting

Gradient boosting is a machine learning paradigm for functional estimation. In function estimation problems, we have a training sample $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^H$, and the goal is to estimate a function F that maps $\boldsymbol{x}_i \in \mathcal{X}$ to $\boldsymbol{y}_i \in \mathcal{Y}$. For simplicity, we will assume in this section that $\mathcal{Y} \subseteq \mathbb{R}$. The estimated function, denoted by F^*, is obtained by solving the optimization problem

$$F^* = \arg\min_{F\in\mathcal{F}} \frac{1}{H} \sum_{i=1}^{H} L(y_i, F(\boldsymbol{x}_i)), \tag{7}$$

where L is the loss function defined in (1) and $\mathcal{F}$ contains functions of the additive form

$$F(\boldsymbol{x}) = \sum_{p=1}^{P} w_p k(\boldsymbol{x}; \beta_p), \tag{8}$$

where $k_p(\cdot; \beta_p)$ are *weak models* or learners that are simple functions parameterized by β_p, *e.g.*, $k(\boldsymbol{x}; \beta_p) = \beta_p^T \boldsymbol{x}$. Substituting (8) in (7) yields an optimization problem in $\{w_p\}_{p=1}^P$ and $\{\beta_p\}_{p=1}^P$, which is usually solved in a forward "stage-wise" fashion, where we start with an initial guess F_0 and then for $p = 2, \dots, P$,

$$w_p, \beta_p = \arg\min_{w,\beta} \sum_{i=1}^{H} L(y_i, F_{p-1}(\boldsymbol{x}_i) + wk(\boldsymbol{x}_i; \beta)), \tag{9}$$

with $F_p(\boldsymbol{x}) = F_{p-1}(\boldsymbol{x}) + w_p k(\boldsymbol{x}; \beta_p)$.

Gradient Boosted Machine (GBM) is a tree-based gradient boosting algorithm that uses a regression tree as a weak learner $k(\boldsymbol{x}, \beta)$. In particular, at each step p in the stage-wise optimization (9), a regression tree partitions the input $\boldsymbol{x}$ into B-disjoint leafs, denoted by the set $\{R_{jp}\}_{j=1}^B$, *i.e.*, $k(\boldsymbol{x}, \beta)$ is a regression tree of depth C where the i^{th} entry of $\boldsymbol{x}$ falls into one of these leaves. Thus, β_p is replaced by the set $\{R_{jp}\}_{j=1}^B$. It has been shown in [13] that (9) can be equivalently solved by

$$\begin{aligned} \gamma_{jp} &= \arg\min_{\gamma} \sum_{\boldsymbol{x}_i \in R_{jp}} L(y_i, F_{p-1}(\boldsymbol{x}_i) + \gamma), \\ F_p(\boldsymbol{x}) &= F_{p-1}(\boldsymbol{x}) + \eta\gamma_{jp} \mathbf{1}\{\boldsymbol{x} \in R_{jp}\}, \end{aligned} \tag{10}$$

where $\mathbf{1}\{A\}$ denotes the indicator function of the set A, and $0 < \eta \le 1$ is the step size.

There are several open source libraries such as XGBoost [18], CatBoost [19], LightGBM [20] that implement GBM. Each library has its own hyperparameters that characterize a regression tree.

IV. Minimax TSGBM

In this section, we propose the new minimax Two-Stage Gradient Boosted Machine (TSGBM) algorithm for parameter estimation. Towards that end, we use LightGBM in the second stage of TS, which is a gradient boosting library developed by Microsoft [20]. In particular, we are interested using it to minimize $\max_{\theta\in\Theta} R(\theta, \delta)$, where δ here is the non-linear map to be delivered by LightGBM. Recall from Section III-B that the approximate minimax risk is given by

$$\min_{\delta} \max_{i=1,\dots,M_\theta} L_i(\delta). \tag{11}$$

Using LightGBM in the context of the minimax TS approach to parameter estimation requires minimizing the "custom loss" $\max_{i=1,\dots,M_\theta} L_i(\delta)$. Such a loss does not have closed form expression, so it cannot be directly used for LightGBM. Therefore, we use a soft-max approximation [21] for $\max_{i=1,\dots,M_\theta} L_i(\delta)$ which is given by

$$\max_{i=1,\dots,M_\theta} L_i(\delta) \approx \frac{\log\left(\sum_{i=1}^{M_\theta} \exp(KL_i(\delta))\right)}{K}, \tag{12}$$

where K is chosen large enough (in the order of $10^3 - 10^4$).

Remark 4.1: In case the data generating mechanism produces i.i.d. or stationary samples, it is possible to set $M_{\boldsymbol{y}} = 1$ in (4) by choosing N sufficiently large; this can be justified from the Ergodic Theorem (see, *e.g.*, [22, Lemma B.1, page 548]). In our examples we thus choose $M_{\boldsymbol{y}} = 1$ and $N \approx 10^4$.

Algorithm 1 describes the implementation of TSGBM. The inputs to this algorithm are the proposal distribution s and a collection of tuning parameters of LightGBM, which includes learning rate (l_r), number of iterations (itr), maximum depth of a regression tree (B_max), number of leaves (l_num), fraction of data used (bg_fr), minimum number of data points in a leaf (d_l_min), and amount of ℓ_1 regularization (λ).

The output of Algorithm 1 is $\delta_{\text{TSGBM}} = g \circ h$, which is the desired TSGBM estimator. Here, h is the data compression function and g is the function approximation provided by LightGBM. To evaluate the performance of this estimator, its Mean Square Error (MSE) is computed via Algorithm 2. The inputs to Algorithm 2 are δ_{TSGBM}, $\tilde{\theta}_{\text{test}}$ and MC: $\tilde{\theta}_{\text{test}}$ is a specific value of the unknown parameter at which the MSE is evaluated by running MC times Algorithm 1, each time returning an estimator δ_{TSGBM} based on a different training sample.

Remark 4.2: In case the parameter vector θ is d-dimensional (with $d > 1$), Algorithm 1 is implemented for each entry of θ separately. Likewise, Algorithm (2) is implemented for each dimension of the unknown parameter.

Remark 4.3: To account for non-linearities in the true system, we propose a non-linear transformation (monomials

True Values		CRLB		MSE, Minimax		MSE, Minimax TSGBM	
η	γ	η	γ	$\hat{\eta}$	$\hat{\gamma}$	$\hat{\eta}$	$\hat{\gamma}$
2	2	1.11×10^{-4}	2.43×10^{-4}	2.58×10^{-4}	5.77×10^{-2}	0.62×10^{-4}	2.17×10^{-4}
2	8	6.93×10^{-6}	3.89×10^{-3}	1.11×10^{-5}	5.61×10^{-2}	0.36×10^{-5}	0.44×10^{-2}
4	2	4.43×10^{-4}	2.43×10^{-4}	6.74×10^{-4}	1.05×10^{-1}	3.76×10^{-4}	2.11×10^{-4}
4	8	2.77×10^{-5}	3.89×10^{-3}	3.84×10^{-5}	6.40×10^{-2}	3.86×10^{-5}	0.52×10^{-2}
8	2	1.77×10^{-3}	2.43×10^{-4}	2.26×10^{-3}	1.89×10^{-1}	2.37×10^{-3}	2.16×10^{-4}

TABLE I: MSE of the orignal minimax TS [10] and minimax TSGBM estimators of the scale and shape parameters, and their corresponding CRLBs.

of degree upto two) of the compressed data, which will then be a part of the training set to GBM. However, the dimension of the transformed data will be much less compared to N.

Algorithm 1 TSGBM Estimator

1: **Input**: s, K, tuning parameters
2: Generate $\theta_i \sim s(\cdot)$, $i = 1, \ldots, M_\theta$
3: **for** $i = 1, \ldots, M_\theta$ **do**
4: $\quad \boldsymbol{y}_i \leftarrow M(\theta_i)$ ▷ Model $M(\theta)$ as data simulator
5: $\quad \boldsymbol{\alpha}_i \leftarrow h(\boldsymbol{y}_i)$ ▷ Data compression step
6: **end for**
7: Construct non-linear features $\phi(\boldsymbol{\alpha}_i)$, where $\phi\colon \mathbb{R}^n \rightarrow \mathbb{R}^m$, where $n < m \ll N$
8: Define the soft-max loss as in (12)
9: Train LightGBM with $\{(\phi(\boldsymbol{\alpha}_i), \theta_i)\}_{i=1}^{M_\theta}$ as the training set and (12) as loss function
10: **Output**: g ▷ Function learned by LightGBM
11: **Final estimator:** $\delta_{\text{TSGBM}} := g \circ h$.

V. Simulation Study

In this section we demonstrate the performance of TSGBM via numerical simulations that include both i.i.d. and non-i.i.d. data generating mechanisms. A detailed description of these mechanisms is provided in Sections V-A, V-B and V-C.

Algorithm 2 MSE for TSGBM

1: **Input**: δ_{TSGBM}, $\tilde{\theta}_{\text{test}}$, MC
2: Sum $\leftarrow 0$
3: **for** $m = 1, \ldots,$ MC **do**
4: $\quad \tilde{\boldsymbol{y}}_{\text{test}} \leftarrow M(\tilde{\theta}_{\text{test}})$ ▷ Model $M(\theta)$ as data simulator
5: $\quad \tilde{\boldsymbol{\alpha}}_{\text{test}} \leftarrow h(\tilde{\boldsymbol{y}}_{\text{test}})$ ▷ Data compression step
6: $\quad$ Construct non-linear features $\phi(\tilde{\boldsymbol{\alpha}}_{\text{test}})$
7: $\quad \widetilde{\theta}_{\text{test}} = \delta_{\text{TSGBM}}(\tilde{\boldsymbol{y}}_{\text{test}})$
8: $\quad$ Sum $\leftarrow (\widetilde{\theta}_{\text{test}} - \tilde{\theta}_{\text{test}})^2$
9: **end for**
10: MSE $\leftarrow \frac{\text{Sum}}{\text{MC}}$
11: **Output**: MSE.

For the numerical simulations, the following setup is considered: The TSGBM estimator(δ_{TSGBM}) is implemented using Algorithm 1, with $M_\theta = M_{\text{train}}$ and $K = 10^3$. For each θ_i in Algorithm 1, we generate observations $\boldsymbol{y}_i$ of dimension $N \times 1$. Unless otherwise stated, we do not construct non-linear features $\phi(\boldsymbol{\alpha}_i)$ in Step 7 of Algorithm 1, in which case training set in Step 9 of Algorithm 1 will be $\{(\boldsymbol{\alpha}_i, \theta_i)\}_{i=1}^{M_\theta}$. For TSGBM, the function approximator g is defined via a regression tree. To generate scatter plots for the estimated vs. true values of the parameter, we draw fresh samples of θ from the proposal distribution s, *i.e.*, $\tilde{\theta}_i \sim s(\theta)$, and collect them in the set $\{\tilde{\theta}_i\}_{i=1}^{M_{\text{test}}}$. We compute the MSE according to Algorithm 2, where the values of $\tilde{\theta}_{\text{test}}$ used are listed in Tables I-IV as "True Values".

A. Weibull Distribution

We first demonstrate the performance of minimax TSGBM on an i.i.d. data generating mechanism. Specifically, we consider a Weibull distribution whose probability density function is parameterized by two parameters, namely, *scale* (η) and *shape* (γ). It has been shown in [10] that the minimax TS approach gives reliable estimate for η. However, estimates for the shape parameter γ when its true value is small are not reliable, in the sense that its MSE is significantly larger when compared to the asymptotic Cramér-Rao lower bound (CRLB).

However, using TSGBM, *i.e.*, with LightGBM in the second stage, shows significant improvement in the MSE for γ, and also for the scale parameter η the MSE is comparable to that outlined in [10]. This has been highlighted in Table I. Also, the training time of minimax TSGBM is considerably smaller compared to the minimax TS algorithm developed in [10], which runs CVXPY [23] for each training sample of θ (and the number of such training samples required for the original minimax TS method to be reliable was fairly large). Minimax TSGBM, on the other hand, does not suffer from this issue, since it directly uses a soft-max maximization objective as the custom loss during its training and minimizes this loss for the entire training sample, *i.e.*, it performs batch training.

For the numerical simulations, the following choices are made: (i) h computes quantiles from order statistics of the observations ($n = 5$), for both the original TS and TSGBM-based estimators. To compute non-linear features, ϕ was taken as described in [10] for η and γ, (ii) The functional approximator g was taken to be linear regression for the original TS estimator, while for TSGBM, we use LightGBM with parameters: l_r= 0.1, itr = 10^3, B_max= 5, l_num = 16, bg_fr = 1, d_l_min = 20, and λ = 10^{-4}, (iii) For both TS estimators, the proposal distribution for η and γ is an uninformative prior [10] over $[1, 20]$. Also, MC= 1000 for MSE computations and (iv) For the original minimax TS estimator, $M_{\text{train}} = 1000$ samples are considered, while for TSGBM, $M_{\text{train}} = 10^4$ since it is more computationally efficient and it gives reliable estimates compared to the original approach.

The scatter plots of estimated vs. true parameter shown in

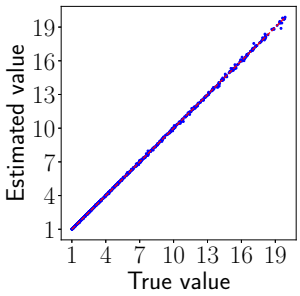

Fig. 2: Scatter plot of estimated vs. true values of scale parameter (η); see Section V-A.

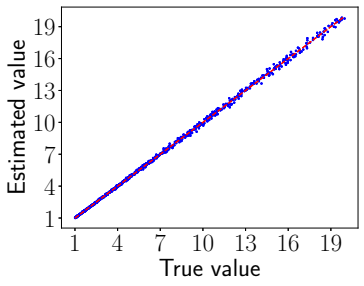

Fig. 3: Scatter plot of estimated vs. true values of shape parameter (γ); see Section V-A.

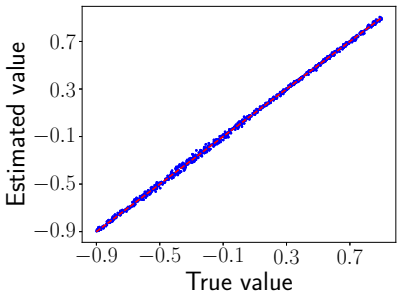

Fig. 4: Scatter plot of estimated vs. true values of a; see Section V-B.

Figures 2, 3 complement the results given in table I, demonstrating that minimax TSGBM yields reliable estimates.

B. Single Parameter Dynamical System

We now consider a single parameter discrete-time state-space model from [8] that is described as

$$\begin{aligned} x_1(k+1) &= a x_1(k) + v_{11}(k), \\ x_2(k+1) &= x_1(k) + a^2 x_2(k) + v_{12}(k), \\ y(k) &= a x_1(k) + x_2(k) + v_2(k), \end{aligned} \tag{13}$$

where $x_1(k)$ and $x_2(k)$ are hidden states and $y(k)$ is the output at time k. $v_{11}(k)$ and $v_{12}(k)$ are the additive process noises and $v_2(k)$ is the observation noise at time k. $v_{11} \sim \mathcal{N}(0,1)$, $v_{12} \sim \mathcal{N}(0,1)$ and $v_2 \sim \mathcal{N}(0, 0.01)$ are mutually uncorrelated Gaussian noises.

True Values (a)	MSE
0.1	1.08×10^{-4}
0.2	1.01×10^{-4}
0.3	8.12×10^{-5}
0.4	7.24×10^{-5}
0.5	1.02×10^{-4}

TABLE II: MSE of TSGBM for different values of a; see Section V-B.

For the above state-space model (13), the parameter of interest is a and we use minimax TSGBM to estimate it. For this purpose, we set $M_{\text{train}} = 10^4$, $N = 10^4$, h as the coefficients of an estimated AR(5) model, ϕ as the collection of all monomials of degree at most 2, $M_{\text{test}} = 10^3$, s as a uniform distribution over $[-1, 1]$, and $\theta = a$. LightGBM is used with parameters: l_r = 0.05, itr = 10^3, B_max= 4, l_num= 8, bg_fr= 0.9, d_l_min = 30, and λ = 10^{-3}.

Figure 4 shows a scatter plot of estimated vs. true values of a. We see that the estimates are quite reliable, in the sense that the relation between the true and estimated values is almost linear.

Next, we compute the MSE of TSGBM by evaluating the trained estimator for 5 different values of a over 1000 Monte Carlo runs in Table II.

Based on the results of this section, we can conclude that the minimax framework of the TS approach is indeed applicable for dynamical systems, where the data generating process is not i.i.d, and TSGBM gives very reliable estimates.

C. Stochastic Volatility Model

In this subsection, we consider a discrete-time stochastic volatility model from [24] that is described by the equations

$$\begin{aligned} x_{k+1} &= a + b x_k + v_k, \\ y_k &= \sqrt{e^{x_k}} r_k, \end{aligned} \tag{14}$$

where $v_k \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$, $r_k \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ and v_k, r_k are mutually uncorrelated. In this model, a, b are the unknown parameters that we need to estimate. Let $\theta = (a\ b)^T$ be the parameter vector of model (14). As one may notice, model (14) has multiplicative noise in its observation. Hence, using an estimated AR(n) process as the compression stage of the TS approach is not a good idea, since an AR(n) model assumes that noise is added to the output. In order to obtain an additive noise structured for (14), we transform the observations by first squaring y_k and then applying the logarithm. This leads to the transformed observations

$$\bar{y}_k = x_k + \bar{r}_k,$$

where $\bar{r}_k = \log(r_k^2)$.

Now, $\bar{y}_k$ has an additive noise structure, and hence an AR(n) model can be used in the first stage of TS. Therefore, the final state-space model equivalent to (14) is

$$\begin{aligned} x_{k+1} &= a + b x_k + v_k \\ \bar{y}_k &= x_k + \bar{r}_k. \end{aligned} \tag{15}$$

We will use this model, instead of (14), to estimate $\theta = (a\ b)^T$ using the TS approach.

For estimating a and b, we consider ranges of a and b to be (i) $a \in [-0.5, 0.5]$ and $b \in [0.45, 0.9]$ and (ii) $a \in [0.1, 0.5]$ and $b \in [0.1, 0.9]$. Figures 5-8 show scatter plots of estimated vs. true values of the unknown parameters a and b of the stochastic volatility model. Tables III, IV list MSE of TSGBM for both a and b. For these figures and tables, we have set $M_{\text{train}} = 10^4$, $N = 3 \times 10^4$, $M_{\text{test}} = 10^3$, s as a uniform distribution over the ranges of a, b considered (see below), and h as the coefficients of an estimated AR(5) model. Also, we have considered a non-linear feature map ϕ consisting of all monomials of degree at most 2, and the following parameters for LightGBM: l_r = 0.05, itr = 10^4, B_max = 6, l_num = 8, bg_fr = 0.9, d_l_min= 30, and λ = 10^{-4}. From Figures 5-8 and Tables III, IV, we see that TSGBM provides fairly reliable estimates of a and b

True Values		MSE	
a	b	$\hat{a}$	$\hat{b}$
-0.5	0.45	6.13×10^{-4}	3.83×10^{-4}
-0.3	0.55	2.59×10^{-4}	2.89×10^{-4}
-0.1	0.65	1.19×10^{-4}	1.13×10^{-4}
0.1	0.75	7.07×10^{-5}	4.75×10^{-5}
0.3	0.85	1.89×10^{-4}	3.57×10^{-5}

TABLE III: MSE of TSGBM for different values of $a \in [-0.5, 0.5]$ and $b \in [0.45, 0.9]$; see Section V-C.

True Values		MSE	
a	b	$\hat{a}$	$\hat{b}$
0.1	0.5	1.74×10^{-4}	4.05×10^{-4}
0.2	0.6	1.22×10^{-4}	1.6×10^{-4}
0.3	0.7	2.12×10^{-4}	8.68×10^{-5}
0.4	0.8	2.21×10^{-4}	4.78×10^{-5}
0.5	0.9	1.27×10^{-3}	1.42×10^{-5}

TABLE IV: MSE of TSGBM for different values of $a \in [0.1, 0.5]$ and $b \in [0.1, 0.9]$; see Section V-C.

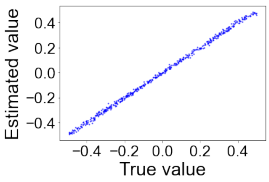

Fig. 5: Scatter plot of estimated vs. true values for $a \in [-0.5, 0.5]$; see Section V-C.

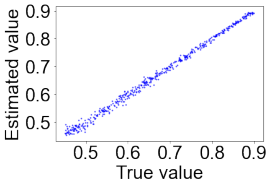

Fig. 6: Scatter plot of estimated vs. true values for $b \in [0.45, 0.9]$; see Section V-C.

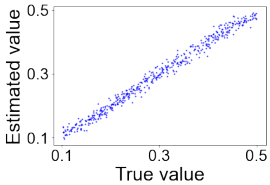

Fig. 7: Scatter plot of estimated vs. true values for $a \in [0.1, 0.5]$; see Section V-C.

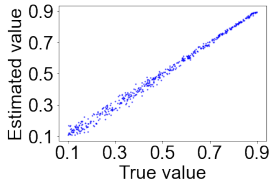

Fig. 8: Scatter plot of estimated vs. true values for $b \in [0.1, 0.9]$; see Section V-C.

with minimal training effort, as it might not be the case with DNNs or recurrent neural networks.

VI. Conclusion

In this paper we have developed a computationally efficient minimax implementation of the TS approach, and we have numerically demonstrated that it is applicable for dynamical systems with non-i.i.d. observations. In particular, the new algorithm, called TSGBM, uses gradient boosting as the supervised learning algorithm in the second stage of TS, and it provides reliable estimates of unknown parameters with minimal training effort, in contrast to using, say, deep neural networks.

It should be remarked that choosing the first stage of TS to appropriately capture the temporal and probabilistic structure of the observations is very important, but its careful design, in the non-i.i.d. case, is left for future work.

References

[1] G. Casella and R. Berger, *Statistical Inference, 2nd Ed.* Duxbury, 2002.

[2] E. Lehmann and G. Casella, *Theory of Point Estimation, 2nd Ed.* Springer-Verlag, 1998.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society (series B)*, vol. 39, no. 1, pp. 1–22, 1977.

[4] T. Söderström and P. Stoica, *Instrumental Variable Methods for System Identification*. Springer-Verlag, 1983.

[5] L. Ljung, "On the consistency of prediction error identification methods," in *System Identification Advances and Case Studies* (R. K. Mehra and D. G. Lainiotis, eds.), pp. 121–164, Elsevier, 1976.

[6] C. Gourieroux, A. Monfort, and E. Renault, "Indirect inference," *Journal of Applied Econometrics*, vol. 8, no. S, pp. S85–118, 1993.

[7] C. Gouriéroux and A. Monfort, *Simulation-Based Econometric Methods*. Oxford University Press, 1997.

[8] S. Garatti and S. Bittanti, "Estimation of white-box model parameters via artificial data generation: a two-stage approach," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 11409–11414, 2008.

[9] S. Garatti and S. Bittanti, "A new paradigm for parameter estimation in system modeling," *International Journal of Adaptive Control and Signal Processing*, vol. 27, no. 8, pp. 667–687, 2013.

[10] B. Lakshminarayanan and C. R. Rojas, "A statistical decision-theoretical perspective on the two-stage approach to parameter estimation," in *61st IEEE Conference on Decision and Control (CDC)*, pp. 5369–5374, 2022.

[11] R. E. Schapire, "The boosting approach to machine learning: An overview," *Nonlinear estimation and classification*, pp. 149–171, 2003.

[12] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, pp. 1189–1232, 2001.

[13] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[14] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis, 2nd Ed.* Springer, 1985.

[15] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2012.

[16] G. Calafiore and M. Campi, "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51(5), pp. 742–753, 2006.

[17] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[18] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," KDD'16, p. 785–794, 2016.

[19] A. N. Dorogush, V. Ershov, and A. Gulin, "Catboost: gradient boosting with categorical features support," *arXiv preprint arXiv:1810.11363*, 2018.

[20] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *NeurIPS*, 2017.

[21] J. D. Cook, "Soft maximum." `https://www.johndcook.com/blog/2010/01/13/soft-maximum/`, January 2010.

[22] T. Söderström and P. Stoica, *System Identification*. Prentice Hall, 1989.

[23] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[24] J. Courts, A. G. Wills, T. B. Schön, and B. Ninness, "Variational system identification for nonlinear state-space models," *Automatica*, vol. 147, p. 110687, 2023.