Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

Article

Adaptive Online Learning for Time Series Prediction

Weijia Shao [1,*], Lukas Radke [1], Fikret Sivrikaya [2], and Sahin Albayrak [1,2]

[1] Technische Universität Berlin; Ernst-Reuter-Platz 7, 10587 Berlin, Germany
[2] GT-ARC gemeinnützige GmbH; Ernst-Reuter-Platz 7, 10587 Berlin, Germany
* Correspondence: weijia.shao@campus.tu-berlin.de

Abstract: We study the problem of predicting time series data using the autoregressive integrated moving average (ARIMA) model in an online manner. Existing algorithms require model selection, which is time consuming and inapt for the setting of online learning. Using adaptive online learning techniques, we develop algorithms for fitting ARIMA models with fewest possible hyperparameters. We analyse the regret bound of the proposed algorithms and examine their performance using experiments on both synthetic and real world datasets.

Keywords: Time Series Analysis; Online Optimisation; Online Model Selection

1. Introduction

An *Autoregressive Integrated Moving Average* (ARIMA) model, which is important for time series analysis [1–5], specifies that the values of a time series depend linearly on their previous values and error terms. In recent years, *online learning methods* have been applied to estimating the ARIMA models for their efficiency and scalability [6–9]. These methods are based on the fact that any ARIMA model can be approximated by a finite dimensional *Autoregressive* (AR) model, which can be fitted incrementally using online convex optimisation algorithms. However, to guarantee accurate predictions, these methods require a proper configuration of hyperparameters, such as the diameter of the decision set, the learning rate, the order of differencing, and the lag of the AR model. These hyperparameters need either to be set according to prior knowledge about the dataset, which is difficult to obtain in practice, or to be tuned using a collected dataset, which is notoriously expensive and inapt for the online setting.

Given a new problem of predicting time series values, it appears that tuning the hyperparameters of the online algorithms can negate the benefits of the online setting. To avoid this, we propose new algorithms for learning an ARIMA model online with fewest possible hyperparameters, while their performance can still be guaranteed in both theory and practice. We first add more "flavor of online learning" by considering an adversarial setting for multivariate time series, where the values of a time series are taken from a vector space, and the error terms are generated arbitrarily. We show that all ARIMA models with fixed order of differencing can be approximated using an AR model of the same order for a large enough lag. Then we propose new algorithms learning the AR model adaptively without requiring any prior knowledge about the model parameters. For Lipschitz-continuous loss functions, we apply a new algorithm based on the adaptive *follow the regularised leader* (**FTRL**) framework [10] and show that our algorithm achieves a sublinear regret bound depending on the data sequence and the Lipschitz constant. We provide some special treatment on the commonly used *squared error* due to its non-Lipschitz continuity. To obtain a data-dependent regret bound, we combine polynomial regulariser [11] with the adaptive **FTRL** framework. Finally, to find the proper order and lag of the AR model in an online manner, we simultaneously maintain multiple AR models and apply an adaptive hedge algorithm to aggregate their predictions. In the previous attempts [12,13], the *exponentiated gradient* algorithm has been directly applied to aggregating the predictions, which not only requires tuning of the learning rate, but also yields a regret bound depending on the loss incurred by the worst model. Our adaptive hedge algorithm is parameter-free and guarantees a regret bound depending on the time series sequence. In

Citation: Shao, W.; Radke, L.; Sivrikaya, F.; Albayrak S. Title. *Preprints* **2021**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

 doi:10.20944/preprints202104.0601.v1

addition to the theoretical analysis, we also demonstrate the performance of the proposed algorithm using both synthetic and real world datasets.

The rest of the paper is organised as follows. In Section 2, we review the existing work on the subject. Then we introduce the notation, learning model and formal description of the problem in Section 3. Next, we present and analyse our algorithms in Section 4. Section 5 demonstrates the empirical performance of the proposed methods. Finally, we conclude our work with some future research directions in Section 6.

2. Related Work

An ARIMA model can be fitted using statistical methods such as *Recursive Least Square* and *Maximum Likelihood Estimation*, which are not only based on strong assumptions on the noise terms and data generation [14–16], but also require solution of non-convex optimisation problems [17]. Although these assumptions can be relaxed [17–20], the pre-trained models can still not deal with *concept drift* [7]. Moreover, retraining is time consuming and memory intensive, especially for large-scale datasets. The idea of applying regret minimisation techniques to *Autoregressive Moving Average* (ARMA) prediction was first introduced in [6]. The authors propose online algorithms incrementally producing predictions close to the values generated by the best ARMA model. This idea has been extended to ARIMA(p, q, d) models in [7], by learning the AR(m) model of the higher order differencing of the time series. Further extensions to multiple time series can be found in [8,9], while the problem of predicting time series with missing data has been addressed in [21].

In order to obtain accurate predictions, the lag of the AR model and the order of differencing have to be tuned, which has been well studied in the offline setting. In some textbooks [15,22,23], *Akaike's Information Criterion* (**AIC**) and the *Bayesian Information Criterion* (**BIC**) are recommended for this task. Both of them require prior knowledge and strong assumptions about the variance of the noise [15], and are time and space consuming as they require numerical simulation, such as cross-validation on previously collected datasets. Nevertheless, given a properly selected lag m and order d, online convex optimisation techniques such as *Online Newton Step* (**ONS**) or *Online Gradient Descent* (**OGD**) can be applied to fitting the model in the regret minimisation framework [6–9]. However, both algorithms introduce additional hyperparameters for controlling the learning rate and the numerical stability.

The idea of selecting hyperparameters for online time series prediction has been proposed in [12,13]. Regarding the online AR predictor with different lags as experts, the authors aggregate over predictors by applying multiplicative weights algorithm for prediction with expert advice. The proposed algorithm is not optimal for time series prediction, since the regret bound of the chosen algorithm depends on the largest loss incurred by the experts [24]. Furthermore, each individual expert still requires that the parameters are taken from a compact decision set, the diameter of which needs to be tuned in practice. A series of recent works on parameter-free online learning [25–28] have provided possibilities of achieving sublinear regret without prior information on the decision set. Given an unconstrained online convex optimisation problem, these algorithms usually relax the problem by assuming a bound on the gradient and guarantee a regret bound depending on it. Unfortunately, a bound on the gradient is an unrealistic assumption for unbounded time series and the squared error loss.

Our idea is based on the combination of the adaptive **FTRL** framework [10] and the idea of handling relative Lipschitz continuous functions [11], which allows us to devise an online algorithm with a data-dependent regret upper bound. To aggregate the results, we propose an adaptive optimistic algorithm such that the overall regret depends on the data sequence instead of worst case loss.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

3. Preliminary and Learning Model

Let X_t denote the value observed at time t of a time series. We assume that X_t is taken from a finite dimensional real vector space $\mathbb{X}$ with norm $\|\cdot\|$. We denote by $\mathcal{L}(\mathbb{X},\mathbb{X})$ the vector space of bounded linear operators from $\mathbb{X}$ to $\mathbb{X}$ and $\|\alpha\|_{\mathrm{op}} = \sup_{x\in\mathbb{X},x\neq 0}\frac{\|\alpha x\|}{\|x\|}$ the corresponding operator norm. An AR(p) model is given by

$$X_t = \sum_{i=1}^{p} \alpha_i X_{t-i} + \epsilon_t,$$

where $\alpha_i \in \mathcal{L}(\mathbb{X},\mathbb{X})$ is a linear operator and $\epsilon_t \in \mathbb{X}$ is some error term. The ARMA(p,q) model extends the AR(p) model by adding a *moving average* (MA) component as follows

$$X_t = \sum_{i=1}^{p} \alpha_i X_{t-i} + \sum_{i=1}^{q} \beta_i \epsilon_{t-i} + \epsilon_t,$$

where $\epsilon_t \in \mathbb{X}$ is the error term and $\beta_i \in \mathcal{L}(\mathbb{X},\mathbb{X})$. We define the d-th order differencing of the time series as $\nabla^d X_t = \nabla^{d-1}X_t - \nabla^{d-1}X_{t-1}$ for $d \geq 1$ and $\nabla^0 X_t = X_t$. The ARIMA(p,q,d) model assumes that the d-th order differencing of the time series follows an ARMA(p,q) model. In this section, this general setting suffices for introducing the learning model. In the following sections, we fix the basis of $\mathbb{X}$ to obtain implementable algorithms, for which different kinds of norms and inner products for vectors and matrices are needed. We provide a table of required notation in Appendix C.

In this paper, we consider the setting of online learning, which can be described as an iterative game between a player and an adversary. In each round t of the game, the player makes a prediction $\tilde{X}_t$. Next, the adversary chooses some X_t and reveals it to the player, who then suffers the loss $l(X_t,\tilde{X}_t)$ for some convex loss function $l : \mathbb{X}\times\mathbb{X} \to \mathbb{R}$. The ultimate goal is to design a strategy for the player to minimise the cumulative loss $\sum_{t=1}^{T} l(X_t,\tilde{X}_t)$ of T rounds. For simplicity, we define

$$l_t : \mathbb{X} \to \mathbb{R}, X \mapsto l(X_t, X).$$

In classical textbooks about time series analysis, the signal is assumed to be generated by a model, based on which the predictions are made. In this paper, we make no assumptions on the data generation. Therefore, minimising the cumulative loss is in general impossible. An achievable objective is to keep a possibly small regret of not having chosen some ARIMA(p,q,d) model to generate the prediction $\tilde{X}_t$. Formally, we denote by $\tilde{X}_t(\alpha,\beta)$ the prediction using the ARIMA(p,q,d) model parameterised by α and β, given by [1]

$$\tilde{X}_t(\alpha,\beta) = \sum_{i=1}^{p} \alpha_i \nabla^d X_{t-i} + \sum_{i=1}^{q} \beta_i \epsilon_{t-i} + \sum_{i=0}^{d-1} \nabla^i X_{t-1}. \tag{1}$$

The cumulative regret of T rounds is then given by

$$\mathrm{R_T}(\alpha,\beta) = \sum_{t=1}^{T} l_t(\tilde{X}_t) - \sum_{t=1}^{T} l_t(\tilde{X}_t(\alpha,\beta)).$$

The goal of this paper is to design a strategy for the player such that the cumulative regret grows sublinearly in T. In the ideal case, in which the data are actually generated by an ARIMA process, the prediction generated by the player yields a small loss. Otherwise, the predictions are always close to these produced by the best ARIMA model independent of the data generation. Following the adversarial setting in [6], we allow the sequences $\{X_t\}$, $\{\epsilon_t\}$ and the parameter α, β to be selected by the adversary. Without any restrictions on

[1] In the paper, we do not directly address the problem of the cointegration, where the third term should be applied to a low rank linear operator.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

the model, this is nothing different than the impossible task of minimising the cumulative loss, since ϵ_{t-1} can always be selected such that $X_t = \tilde{X}_t(\alpha, \beta)$ holds for all t. Therefore, we make the following assumptions throughout this paper:

Assumption 1. *$X_t = \epsilon_t + \tilde{X}_t(\alpha, \beta)$ and there is some $R > 0$ such that $\|\epsilon_t\| \leq R$ for all $t = 1, \dots T$.*

Assumption 2. *The coefficients β_i satisfy $\sum_{i=1}^{q} \|\beta_i\|_{\text{op}} \leq 1 - \epsilon$ for some $\epsilon > 0$.*

Since we are interested in competing against predictions generated by ARIMA models, we assume that ϵ_t is selected in the way as if X_t is generated by the ARIMA process. Furthermore, we assume the norm $\|\epsilon_t\|$ is upper bounded within T iterations. Assumption 2 is a sufficient condition for the MA component to be invertible, which prevents it from going to infinity as $t \to \infty$ [23].

Our work is based on the fact that we can compete against an ARIMA(p, q, d) model by taking predictions from AR(m) model of the d-th order differencing for large enough m, which is shown in the following lemma, the proof of which can be found in Appendix A.

Lemma 1. *Let $\{X_t\}$, $\{\epsilon_t\}$, α and β be as assumed in Assumption 1-2. Then there is some $\gamma \in \mathcal{L}(\mathbb{X}, \mathbb{X})^m$ with $m \geq \frac{q \log T}{\log \frac{1}{1-\epsilon}} + p$ such that*

$$\|\nabla^d \tilde{X}_t(\gamma) - \nabla^d \tilde{X}_t(\alpha, \beta)\| \leq (1-\epsilon)^{\frac{t}{q}} R + \frac{2R}{T},$$

holds for all $t = 1 \dots T$, where we define $\nabla^d \tilde{X}_t(\gamma) = \sum_{i=1}^{m} \gamma_i \nabla^d X_{t-i}$.

As can be seen from the lemma, a prediction $\tilde{X}_t(\gamma)$ generated by the process

$$\tilde{X}_t(\gamma) = \sum_{i=1}^{m} \gamma_i \nabla^d X_{t-i} + \sum_{i=0}^{d-1} \nabla^i X_{t-1},$$

is close to the prediction $\tilde{X}_t(\alpha, \beta)$ generated by the ARIMA process. In the previous works [6,7], the loss function l_t is assumed to be Lipschitz continuous to control the difference of loss incurred by the approximation. In general, this does not hold for *squared error*. However, from Assumption 1 and Lemma 1, it follows that both $\tilde{X}_t(\alpha, \beta)$ and $\tilde{X}_t(\gamma)$ lie in a compact set around X_t with a bounded diameter. Given the convexity of l, which is local Lipschitz continuous in the compact convex domain, we obtain a similar property

$$l(X_t, \tilde{X}_t(\gamma)) - l(X_t, \tilde{X}_t(\alpha, \beta)) \leq L(X_t) \|\nabla^d \tilde{X}_t(\gamma) - \nabla^d \tilde{X}_t(\alpha, \beta)\|,$$

where $L(X_t)$ is some constant depending on X_t. For *squared error*, it is easy to verify that the Lipschitz constant depends on $\|\nabla^d X_t\|$, the boundedness of which can be reasonably assumed. To avoid extraneous details, we simply add the third assumption:

Assumption 3. *Define set $\mathcal{X}_t = \{X \in \mathbb{X} | \|X - X_t\| \leq 4R\}$. There is a compact convex set $\mathcal{X} \supseteq \bigcup_{t=1}^{T} \mathcal{X}_t$, such that l_t is L-Lipschitz continuous in $\mathcal{X}$ for $t = 1, \dots T$.*

The next corollary shows that the losses incurred by the ARIMA and its approximation are close, which allows us to take predictions from the approximation.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

Corollary 1. *Let $\{X_t\}$, $\{\epsilon_t\}$, α, β and l be as assumed in Assumption 1-3. Then there is some $\gamma \in \mathcal{L}(\mathbb{X}, \mathbb{X})^m$ with $m \geq \frac{q \log T}{\log \frac{1}{1-\epsilon}} + p$, such that*

$$\sum_{t=1}^{T} l_t(\tilde{X}_t(\gamma)) - l_t(\tilde{X}_t(\alpha, \beta)) \leq LR(\frac{1}{1-(1-\epsilon)^{\frac{1}{q}}} + 2)$$

holds for all $t = 1 \dots T$.

Proof. It follows from Assumption 1 and Lemma 1 that $\tilde{X}_t(\gamma), \tilde{X}_t(\alpha, \beta) \in \mathcal{X}$ holds for all $t = 1, \dots T$. Together with Assumption 3, we obtain

$$\sum_{t=1}^{T} (l_t(\tilde{X}_t(\gamma)) - l_t(\tilde{X}_t(\alpha, \beta))) \leq L \sum_{t=1}^{T} \|\tilde{X}_t(\gamma) - \tilde{X}_t(\alpha, \beta)\|.$$

Applying Lemma 1, we obtain the claimed result. □

4. Algorithms and Analysis

From Corollary 1, it follows clearly that an ARIMA(p, q, d) model can be approximated by an integrated AR model with large enough m. However, neither the order of differencing d nor the lag m is known. To circumvent tuning them using a previously collected dataset, we propose a framework with a two-level hierarchical construction, which is described in Algorithm 1.

The idea is to maintain a master algorithm $\mathcal{M}$ and a set of slave algorithms $\{\mathcal{A}_m | m = 1, \dots, K\}$. At each step t, the master algorithm receives predictions $\tilde{X}_t^k$ from $\mathcal{A}_k$ for $k = 1, \dots, K$. Then it comes up with a convex combination $\tilde{X}_t = \sum_{i=1}^{K} w_t^i \tilde{X}_t^i$ for some $w_t \in \Delta$ in the simplex. Next, it observes X_t and computes the loss $l_t(X_t^k(\gamma))$ for each slave $\mathcal{A}_k$, which is then used to update $\mathcal{A}_k$ and w_{t+1}. Let $\{\tilde{X}_t^k\}$ be the sequence generated by some slave k. We define the regret of not having chosen the prediction generated by slave k as

$$R_T(k) = \sum_{t=1}^{T} l_t(\sum_{i=1}^{K} w_t^i \tilde{X}_t^i) - \sum_{t=1}^{T} l_t(\tilde{X}_t^k),$$

and the regret of the slave k

$$R_T(\mathcal{A}_k) = \sum_{t=1}^{T} l_t(\tilde{X}_t^k) - \sum_{t=1}^{T} l_t(\tilde{X}_t(\gamma_k)),$$

where $\tilde{X}_t(\gamma_k)$ is the prediction generated by an integrated AR model parameterised by γ_k. Let $\mathcal{A}_k$ be some slave. Then the regret of this two-level framework can obviously be decomposed as

$$R_T(\alpha, \beta) = R_T(k) + R_T(\mathcal{A}_k) + \underbrace{\sum_{t=1}^{T} l_t(\tilde{X}_t(\gamma_k)) - \sum_{t=1}^{T} l_t(\tilde{X}_t(\alpha, \beta))}_{\text{Corollary 1}}.$$

For γ_k, α and β satisfying the condition in Corollary 1 [2], the marked term above is upper bounded by a constant, i.e.

$$\sum_{t=1}^{T} l_t(\tilde{X}_t(\gamma_k)) - \sum_{t=1}^{T} l_t(\tilde{X}_t(\alpha, \beta)) \in \mathcal{O}(1).$$

[2] This is not a condition of having a correct algorithm. With more salves, there are more α, β satisfying the condition. We increase the freedom of the model by increasing the number of slaves.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

If the regret of the master and the slaves grow sublinearly in T, we can achieve an overall

Algorithm 1 Two-level framework

Input: K instances of the slave algorithm $\mathcal{A}_1, \ldots, \mathcal{A}_K$. An instance of master algorithm $\mathcal{M}$.
for $t = 1$ to T **do**
  Get $\tilde{X}_t^i$ from each $\mathcal{A}_i$
  Get $w_t \in \Delta^K$ from $\mathcal{M}$ ▷ Δ^K is the standard K-simplex
  Integrate the prediction: $\tilde{X}_t = \sum_{i=1}^K w_t^i \tilde{X}_t^i$
  Observe X_t
  Define $z_t \in \mathbb{R}^K$ with $z_{i,t} = l_t(\tilde{X}_t^i)$
  Update $\mathcal{A}_i$ using $z_{i,t}$ for $i = 1, \ldots, K$
  Update $\mathcal{M}$ using z_t
end for

sublinear regret upper bound, which is formally described in the following corollary.

Corollary 2. *Let $\mathcal{A}_i$ be an online learning algorithm against an* AR(m_i) *model parameterised by γ^i for $i = 1, \ldots, K$. For any* ARIMA *model parameterised by α and β, if there is a $k \in \{1, \ldots, K\}$ such that $\tilde{X}_t(\gamma^k)$, $\tilde{X}_t(\alpha, \beta)$ and $\{X_t\}$ satisfy Assumption 1-3, then running algorithm 1 with $\mathcal{M}$ and $\mathcal{A}_1, \ldots, \mathcal{A}_K$ guarantees*

$$\sum_{t=1}^{T} (l_t(\tilde{X}_t) - l_t(\tilde{X}_t(\alpha, \beta))) \leq \mathcal{R}_T(k) + \mathcal{R}_T(\mathcal{A}_k) + \mathcal{O}(1).$$

Next, we design and analyse parameter-free algorithms for the slaves and the master.

4.1. Parameter-Free Online Learning Algorithms

4.1.1. Algorithms for Lipschitz Loss

Given fixed m and d, an integrated $AR(m)$ model can be treated as an ordinary linear regression model. In each iteration t, we select $\gamma_t = (\gamma_{1,t}, \ldots, \gamma_{m,t}) \in \mathcal{L}(\mathbb{X}, \mathbb{X})^m$ and make prediction

$$\tilde{X}_t(\gamma_t) = \sum_{i=1}^{m} \gamma_{i,t} \nabla^d X_{t-i} + \sum_{i=0}^{d-1} \nabla^i X_{t-1}.$$

Since l_t is convex, there is some subdifferential $g_t \in \partial l_t(\tilde{X}_t(\gamma_t))$ such that

$$l_t(\tilde{X}_t(\gamma_t)) - l_t(\tilde{X}_t(\gamma)) \leq g_t(\sum_{i=1}^{m} (\gamma_{i,t} - \gamma_i) \nabla^d X_{t-i}),$$

for all $\gamma \in \mathcal{L}(\mathbb{X}, \mathbb{X})^m$. Define $g_{i,t} : \mathcal{L}(\mathbb{X}, \mathbb{X}) \to \mathbb{R}, v \mapsto g_t(v \nabla^d X_{t-i})$. The regret can be further upper bounded by

$$\sum_{t=1}^{T} l_t(\tilde{X}_t(\gamma_t)) - l_t(\tilde{X}_t(\gamma)) \leq \sum_{t=1}^{T} \sum_{i=1}^{m} g_{i,t}(\gamma_{i,t} - \gamma_i). \tag{2}$$

Thus, we can cast the online linear regression problem to an online linear optimisation problem. Unlike the previous work, we focus on the unconstrained setting, where γ_t is not picked from a compact decision set. In this setting, we can apply an **FTRL** algorithm with an adaptive regulariser. To obtain an efficient implementation, we fix a basis for both $\mathbb{X}$ and $\mathbb{X}_*$. Now we can assume $\mathbb{X} = \mathbb{X}_* = \mathbb{R}^n$ and work with the matrix representation of $\gamma \in \mathcal{L}(\mathbb{X}, \mathbb{X})$. It is easy to verify that (2) can be rewritten as

$$\sum_{t=1}^{T} l_t(\tilde{X}_t(\gamma_t)) - l_t(\tilde{X}_t(\gamma)) \leq \sum_{t=1}^{T} \sum_{i=1}^{m} \langle g_t \nabla^d X_{t-i}^\top, \gamma_{i,t} - \gamma_i \rangle_F,$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

where $\langle A, B\rangle_F = \mathrm{tr}(A^\top B)$ is the *Frobenius inner product*. It is well known that the *Frobenius inner product* can be considered as a dot product of vectorised matrices, with which we obtain a simple first order[3] algorithm described in Algorithm 2. The cumulative regret of

Algorithm 2 ARIMA-AdaFTRL

Input: $L_1 > 0$
Initialise $\theta_{1,i}$ arbitrarily, $\eta_{1,i} = 0$, $G_{i,0} = 0$ for $i = 1, \ldots, m$
for $t = 1$ to T **do**
  for $i = 1$ to m **do**
    $G_{i,t} = \max\{G_{i,t-1}, \|\nabla^d X_{t-i}\|_2\}$
    $\eta_{i,t} = \|\theta_{i,1}\|_F + \sqrt{\sum_{s=1}^{t-1} \|g_{i,s}\|_F^2 + (L_t G_{i,t})^2}$
    if $\eta_{i,t} \neq 0$ **then**
      $\gamma_{i,t} = \frac{\theta_{i,t}}{\eta_{i,t}}$
    else
      $\gamma_{i,t} = 0$
    end if
  end for
  Play $\tilde{X}_t(\gamma_t)$
  Observe X_t and $h_t \in \partial l_t(\tilde{X}_t(\gamma_t))$
  $L_{t+1} = \max\{L_t, \|g_t\|_2\}$
  for $i = 1$ to m **do**
    $g_{i,t} = g_t \nabla^d X_{t-i}^\top$
    $\theta_{i,t+1} = \theta_{i,t} - g_{i,t}$
  end for
end for

Algorithm 2 can be upper bounded using the following theorem.

Theorem 1. *Let $\{X_t\}$ be any sequence of vectors taken from $\mathbb{X}$. Algorithm 2 guarantees*

$$\begin{aligned}
&\sum_{t=1}^{T} l_t(\tilde{X}_t(\gamma_t)) - l_t(\tilde{X}_t(\gamma)) \\
\leq &\sum_{i=1}^{m} \left(\frac{\|\gamma_i\|_F^2 L_{T+1}}{2} + L_{T+1} + \frac{L_{T+1}^2}{L_1}\right) \sqrt{\sum_{t=1}^{T} \|\nabla^d X_{t-i}\|_2^2} \\
&+ \sum_{i=1}^{m} \frac{(L_{T+1} G_{i,T+1} + \|\theta_{i,1}\|_F)\|\gamma_i\|_F^2 + \|\theta_{i,1}\|_F}{2}.
\end{aligned}$$

For an L-Lipschitz loss function l_t, in which L_{T+1} is upper bounded by L, we obtain a sublinear regret upper bound depending on the sequence of d-th order differencing $\{\nabla^d X_t\}$. In case L is known, we can set $L_0 = L$, otherwise picking L_0 arbitrarily from a reasonable range, e.g. $L_0 = 1$, would not make a devastating impact on the performance of the algorithms.

4.1.2. Algorithms for *Squared Errors*

For the commonly used *squared error* given by

$$l_t(\tilde{X}_t(\gamma_t)) = \frac{1}{2}\|\tilde{X}_t(\gamma_t) - X_t\|_2^2,$$

[3] The computational complexity per iteration depends linearly on the dimension of the parameter, i.e. $\mathcal{O}(n^2 m)$.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

it can be verified that g_t can be represented as a vector

$$g_t = \sum_{i=1}^{m} \gamma_{i,t} \nabla^d X_{t-i} - \nabla^d X_t$$

for all t. Algorithm 2 could fail due to the dependency on $\|g_t\|_2$, which could be set arbitrarily large due to the adversarially selected X_t. To design a parameter-free algorithm for the *squared error*, we equip **FTRL** with a time-varying polynomial regulariser described in Algorithm 3.

Algorithm 3 ARIMA-AdaFTRL-Poly

Input: $G_0 > 0$
Initialise θ_1 arbitrarily, $G_1 = \max\{G_0, \|\nabla^d X_0\|_2, \ldots, \|\nabla^d X_{-m+1}\|_2\}$
for $t = 1$ to T **do**
  $\eta_t = \|\theta_1\|_F + \sqrt{\sum_{s=1}^{t-1} \|\nabla^d X_s x_s^\top\|_F^2 + (G_t \|x_t\|_2)^2}$
  $\lambda_t = \sqrt{\sum_{s=1}^{t} \|x_s\|_2^4}$
  if $\|\theta_t\|_F \neq 0$ **then**
    Select $c \geq 0$ satisfying $\lambda_t c^3 + \eta_t c = \|\theta_t\|_F$
    $\gamma_t = \frac{c\theta_t}{\|\theta_t\|_F}$
  else
    $\gamma_t = 0$
  end if
  Play $\tilde{X}_t(\gamma_t)$
  Observe X_t and $g_t = \gamma_t x_t - \nabla^d X_t$
  $G_{t+1} = \max\{G_t, \|\nabla^d X_t\|_2\}$
  $\theta_{t+1} = \theta_t - g_t x_t^\top$
end for

Define

$$x_t = \begin{pmatrix} \nabla^d X_{t-1} \\ \vdots \\ \nabla^d X_{t-m} \end{pmatrix}$$

and consider the matrix representation $\gamma_t = \begin{pmatrix} \gamma_{1,t} & \cdots & \gamma_{m,t} \end{pmatrix}$. Then we have $g_t = \gamma_t x_t - \nabla^d X_t$, and the upper bound of the regret can be rewritten as

$$\sum_{t=1}^{T} l_t(\tilde{X}_t(\gamma_t)) - l_t(\tilde{X}_t(\gamma)) \leq \sum_{t=1}^{T} \langle (\gamma_t x_t - \nabla^d X_t) x_t^\top, \gamma_t - \gamma \rangle_F.$$

The idea of Algorithm 3 is to run the **FTRL** algorithm with a polynomial regulariser

$$\frac{\lambda_t}{4} \|\gamma\|_F^4 + \frac{\eta_t}{2} \|\gamma\|_F^2,$$

for increasing sequences $\{\lambda_t\}$ and $\{\eta_t\}$, which leads to updating rule given by

$$\gamma_t = \arg \max_{\gamma \in \mathcal{L}(\mathbb{X}, \mathbb{X})^m} \langle \theta_t, \gamma \rangle_F - \frac{\lambda_t}{4} \|\gamma\|_F^4 - \frac{\eta_t}{2} \|\gamma\|_F^2 = \frac{c\theta_t}{\|\theta_t\|_F},$$

for c satisfying $\lambda_t c^3 + \eta_t c = \|\theta_t\|_F$. Since we have $\lambda_t \geq 0$ and $\eta_t > 0$ for $\theta_1 \neq 0$, c exists and has a closed form expression. The computational complexity per iteration has a linear dependency on the dimension of $\mathcal{L}(\mathbb{X}, \mathbb{X})^m$. The following theorem provides a regret upper bound of Algorithm 3.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

Theorem 2. *Let $\{X_t\}$ be any sequence of vectors taken from $\mathbb{X}$ and*

$$l_t(\tilde{X}_t(\gamma)) = \frac{1}{2}\|X_t - \tilde{X}_t(\gamma)\|_2^2 = \frac{1}{2}\|\nabla^d X_t - \nabla^d \tilde{X}_t(\gamma)\|_2^2$$

be the squared error. We define $x_t = \begin{pmatrix}\nabla^d X_{t-1} & \cdots & \nabla^d X_{t-m}\end{pmatrix}^\top$ and $\gamma = \begin{pmatrix}\gamma_1 & \cdots & \gamma_m\end{pmatrix}$, the matrix representation of $\gamma_1, \ldots \gamma_m \in \mathcal{L}(\mathbb{X}, \mathbb{X})$. Then, Algorithm 3 guarantees

$$\begin{aligned}\sum_{t=1}^{T}(l_t(\tilde{X}_t(\gamma_t)) - l_t(\tilde{X}_t(\gamma))) \leq & \frac{(\sqrt{m}G_{T+1}^2 + \|\theta_1\|_F)\|\gamma\|_F^2}{2} \\ & + \|\theta_1\|_F + (1 + \frac{\|\gamma\|_F^4}{4})\sqrt{\sum_{t=1}^{T}\|x_t\|_2^4} \\ & + (1 + \frac{G_{T+1}}{G_0} + \frac{\|\gamma\|_F^2}{2})\sqrt{\sum_{t=1}^{T}\|\nabla^d X_t x_t^\top\|_F^2}.\end{aligned}$$

for all $\gamma \in \mathcal{L}(\mathbb{X}, \mathbb{X})^m$.

For *squared error*, Algorithm 3 does not require a compact decision set and ensures a sublinear regret bound depends on the data sequence. Similar to Algorithm 2, one can set G_0 according to the prior knowledge about the bounds of the time series. Alternatively, we can simply set $G_0 = 1$ to obtain a reasonable performance.

4.2. Online Model Selection using Master Algorithms

The straightforward choice of the master algorithm would be the *exponentiated gradient* algorithm for prediction with expert advice. However, this algorithm requires tuning of the learning rate and losses bounded by a small quantity, which can not be assumed for our case. The **AdaHedge** [29] solves these problems. However, it yields a worst-case regret bound depending on the largest loss observed, which could be much worse compared to a data dependent regret bound.

Our idea is based on the *adaptive optimistic follow the regularized leader* (**AO-FTRL**) framework [10]. Given a sequence of hints $\{h_t\}$ and loss vectors $\{z_t\}$, **AO-FTRL** guarantees a regret bound related to $\sum_{t=1}^{T}\|z_t - h_t\|_t^2$ for some time varying norm $\|\cdot\|_t$. In our case, where the loss incurred by a slave is given by $l(X_t, \tilde{X}_t^k)$ at iteration t, we simply choose $h_{k,t} = l(\sum_{i=0}^{d-1}\nabla^i X_{t-1}, \tilde{X}_t^k)$. If l is L-Lipschitz in its first argument, then we have $|z_{k,t} - h_{k,t}| \leq L\|\nabla^d X_t\|$, which leads to a data dependent regret. The obtained algorithm is described in Algorithm 4. Its regret is upper bounded by the following theorem, the proof of which is provided in Appendix B.

Theorem 3. *Let $\{\tilde{X}_t\}$, $\{\tilde{X}_t^k\}$, $\{z_t\}$, $\{h_t\}$, and $\{w_t\}$ be as generated in Algorithm 4. Assume l is L-Lipschitz in its first argument and convex in its second argument. Then for any sequence $\{X_t\}$ and slave algorithm $\mathcal{A}_k$, we have*

$$\mathcal{R}_T(k) \leq (\sqrt{2\log K} + \sqrt{\frac{8}{\log K}})\sqrt{\sum_{t=1}^{T} L^2\|\nabla^d X_t\|_2^2}.$$

By Corollary 2, combining Algorithm 4 with Algorithm 2 or 3 guarantees a data-dependent regret upper bound sublinear in T. Note that there is an input parameter d for Algorithm 4, which can be adjusted according to the prior knowledge of the dataset such that $\|\nabla^d X_t\|_2^2$ can be bounded by a small quantity. In case no prior knowledge can be obtained, we can set d to the maximal order of differencing used in the slave algorithms. Arguably, the Lipschitz-continuity is not a reasonable assumption for *squared error* with

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

Algorithm 4 ARIMA-AO-Hedge

Input: predictor $\mathcal{A}_1, \ldots, \mathcal{A}_K$, d
Initialise $\theta_{k,1} = 0$, $\eta_1 = 0$ for $i = 1, \ldots, K$
for $t = 1$ to T **do**
  Get prediction $\tilde{X}_t^i$ from $\mathcal{A}_k$ for $i = 1, \ldots, K$
  Set $Y_t = \sum_{i=0}^{d-1} \nabla^i X_{t-1}$
  Set $h_{i,t} = l(Y_t, \tilde{X}_t^i)$ for $i = 1, \ldots, K$
  if $\eta_1 = 0$ **then**
    Set $w_{i,t} = 1$ for some $i \in \arg\max_{j \in \{1,\ldots,K\}} h_{j,t}$
  else
    Set $w_{i,t} = \frac{\exp(\eta_t^{-1}(\theta_{i,t} - h_{i,t}))}{\sum_{i=1}^{K} \exp(\eta_t^{-1}(\theta_{i,t} - h_{i,t}))}$ for $i = 1, \ldots, K$
  end if
  Predict $\tilde{X}_t = \sum_{i=1}^{K} w_{i,t} \tilde{X}_t^i$
  Observe X_t, update $\mathcal{A}_i$ and set $z_{i,t} = l(X_t, \tilde{X}_t^i)$ for $i = 1, \ldots, K$
  $\theta_{t+1} = \theta_t - z_t$
  $\eta_{t+1} = \sqrt{\frac{1}{2 \log K} \sum_{s=1}^{t} \|h_t - z_t\|_\infty^2}$
end for

unbounded domain. With a bounded $\|\nabla^d X_t\|_2^2$, we can assume that the loss function is locally-Lipschitz, however, with a Lipschitz constant depending on the prediction. In the next section, we show the performance of Algorithm 4 in combination with Algorithms 2 and 3 in different experimental settings.

5. Experiments and Results

In this section, we carry out experiments on both synthetic and real-world data to show that the proposed algorithms can generate promising predictions without tuning hyperparameters.

5.1. Experiment Settings

The synthetic data is generated randomly. We run 20 trials for each synthetic experiment and average the results. For numerical stability, we scale the real world data down so that the values are between 0 and 10. Note that the range of the data are not assumed or used in the algorithms.

Setting 1: Sanity Check

For a sanity check, we generated a stationary 10-dimensional ARIMA$(5, 2, 1)$ process using randomly drawn coefficients.

Setting 2: Time Varying Parameters

Aimed at demonstrating the effectiveness of the proposed algorithm in the non-stationary case, we generated the non-stationary 10-dimensional ARIMA$(5, 2, 1)$ process using time varying parameters. We draw α_1, α_2 and β_1, β_2 randomly and independent, and generate data at iteration t with the ARIMA$(5, 2, 1)$ model parameterised by $\alpha_t = \frac{t}{10^4}\alpha_1 + (1 - \frac{t}{10^4})\alpha_2$ and $\beta_t = \frac{t}{10^4}\beta_1 + (1 - \frac{t}{10^4})\beta_2$.

Setting 3: Time Varying Models

To get a more adversarially selected time series values, we generate the first half of the values using a stationary 10-dimensional ARIMA$(5, 2, 1)$ model and the second half of the values using a stationary 10-dimensional ARIMA$(5, 2, 0)$ model. The model parameters are drawn randomly.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

Stock Data: Time Series with Trend

Following the experiments in [8], we collected the daily stock prices of seven technology companies from Yahoo Finance together with the **S&P 500** index for over twenty years, which has an obvious increasing trend and is believed to exhibit integration.

Google Flu Data: Time Series with Seasonality

We collect estimates of influenza activity of the northern hemisphere countries, which has an obvious seasonal pattern. In the experiment, we examine the performance of the algorithms for handling regular and predictable changes that occur over a fixed period

Electricity Demand: Trend and Seasonality

In this setting, we collect monthly load, gross electricity production, net electricity consumption and gross demand in Turkey from 1976 to 2010. The dataset contains both trend and seasonality.

5.2. Experiments for the Slave Algorithms

We first fix $d = 1$ and $m = 16$ and compare our slave algorithms with **ONS**, **OGD** from [9] for squared error $l_t(\tilde{X}_t) = \frac{1}{2}\|X_t - \tilde{X}_t\|_2^2$ and Euclidean distance $l_t(\tilde{X}_t) = \|X_t - \tilde{X}_t\|_2$. **ONS**, **OGD** stack and vectorise the parameter matrices, and incrementally update the vectorised parameter using the following rule

$$w_{t+1} = \Pi_{\mathcal{W}}(w_t - \eta(\sum_{s=1}^{t} g_t g_t^\top + \lambda I)^{-1} g_t)$$

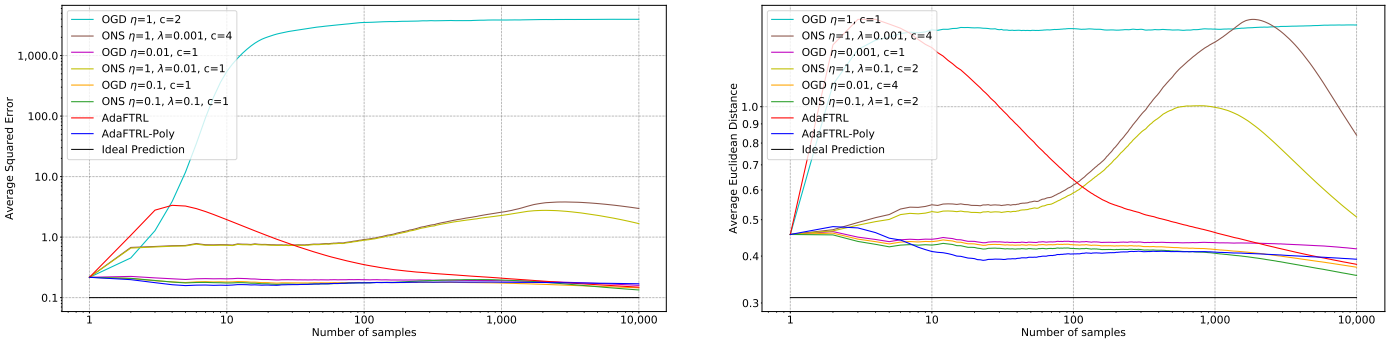
and

$$w_{t+1} = \Pi_{\mathcal{W}}(w_t - \eta g_t),$$

respectively, where g_t is the vectorised gradient at step t, $\mathcal{W}$ is the decision set satisfying $\sup_{u \in \mathcal{W}} \|u\|_2 \leq c$ and the operator $\Pi_{\mathcal{W}}(v)$ projects v into $\mathcal{W}$. We select a list of candidate values for each hyperparameter, evaluate their performance on the whole dataset, and select the configuration with the best performance for comparison. Since the synthetic data are generated randomly, we average the results over 20 trials for stability. The corresponding results are shown in Figures 1-6. To show the impact of the hyperparameters on the performance of the baseline algorithms, we also plot their performance using sub-optimal configurations. Note that, since the error term ϵ_t cannot be predicted, an ideal predictor would suffer an average error rate of at least $\|\epsilon_t\|_2^2$ and $\|\epsilon_t\|_2$ for the two kinds of loss function. This is known for the synthetic datasets and plotted in the figures.

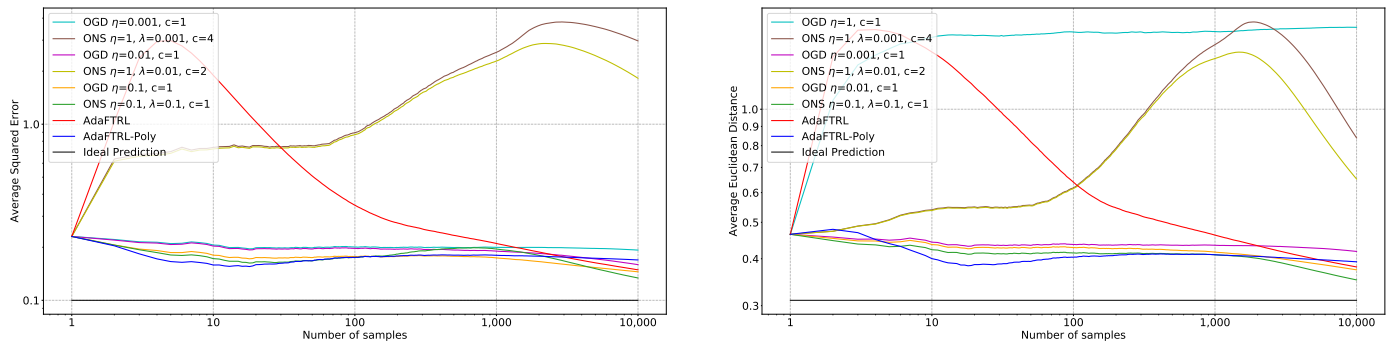
In all settings, both **AdaFTRL** and **AdaFTRL-Poly** have a performance on par with well-tuned **OGD** and **ONS**, which can have extremely bad performance using sub-optimal hyperparameter configurations. In the experiments using synthetic datasets, **AdaFTRL** suffers large loss at the beginning while generating accurate predictions after 2000 iterations. **AdaFTRL-Poly** has more stable performance compared to **AdaFTRL**. In the experiment with Google Flu data, all algorithms suffer huge losses around the iteration 300 due to an abrupt change in the dataset. **OGD** and **ONS** with sub-optimal hyperparameter configurations, despite good performance for the first half of the data, generate very inaccurate predictions after the abrupt change in the dataset. This could lead to a catastrophic failure in practice, when certain patterns do not appear in the dataset collected for hyperparameter tuning. Our algorithms are more robust against this change and perform similarly to **OGD** and **ONS** with optimal hyperparameter configurations.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1



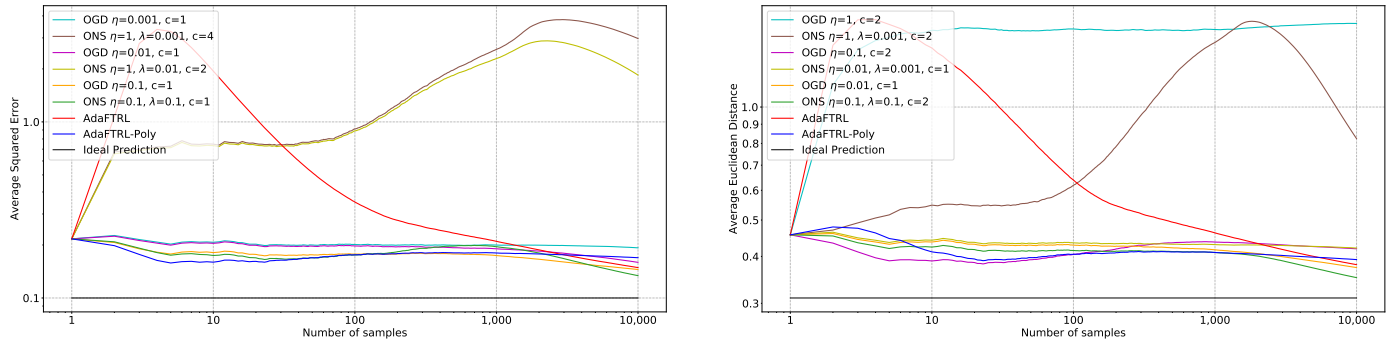
(a) Squared Error **(b)** Euclidean Distance

Figure 1. Results for Setting 1 (Sanity Check), using a stationary ARIMA(5,2,1) model



(a) Squared Error **(b)** Euclidean Distance

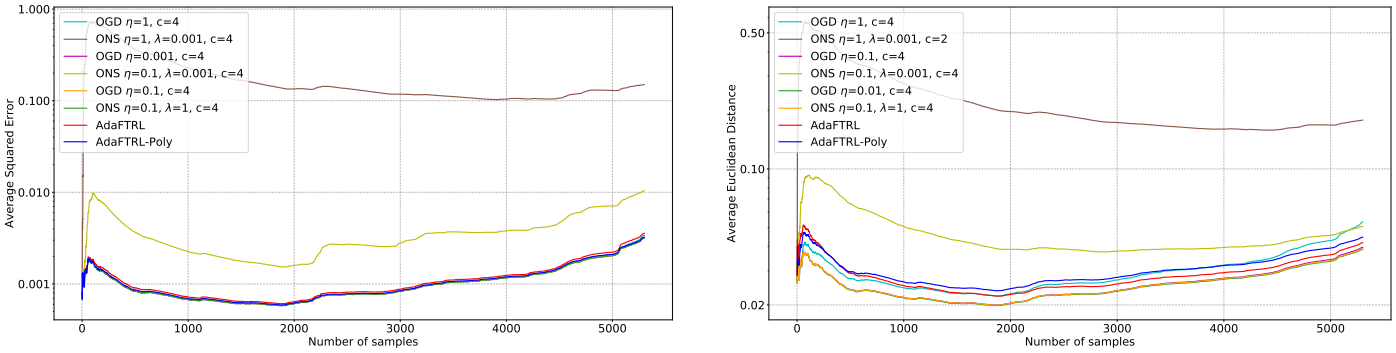
Figure 2. Results for Setting 2 (Time Varying Parameters), using a non-stationary ARIMA(5,2,1) model



(a) Squared Error **(b)** Euclidean Distance

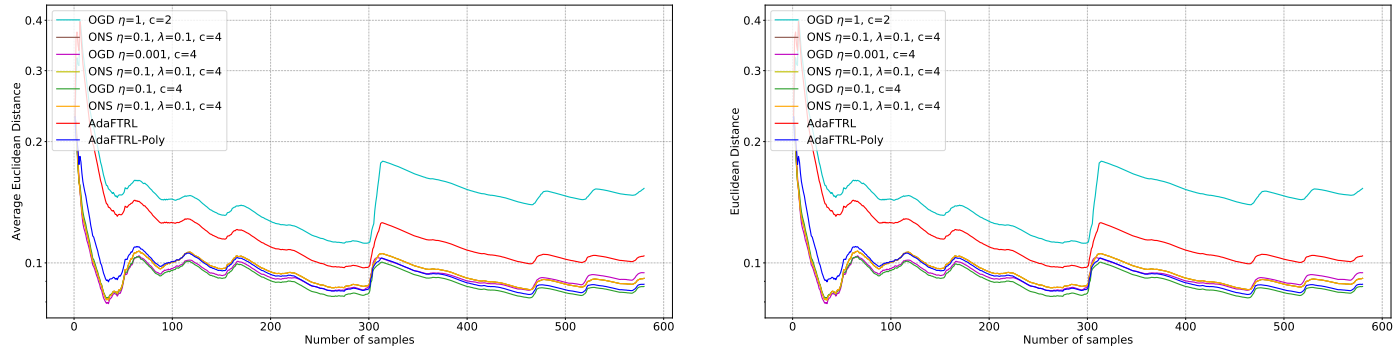
Figure 3. Results for Setting 3 (Time Varying Models), using a combination of stationary ARIMA(5,2,1) and ARIMA(5,2,0) models

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1



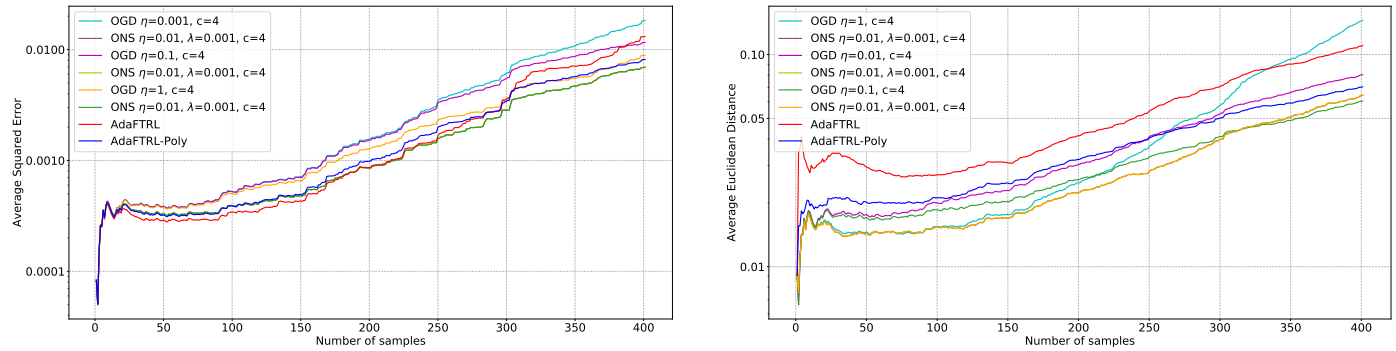
(a) Squared Error **(b)** Euclidean Distance

Figure 4. Results for Stock Data



(a) Squared Error **(b)** Euclidean Distance

Figure 5. Results for Google Flu Data



(a) Squared Error **(b)** Euclidean Distance

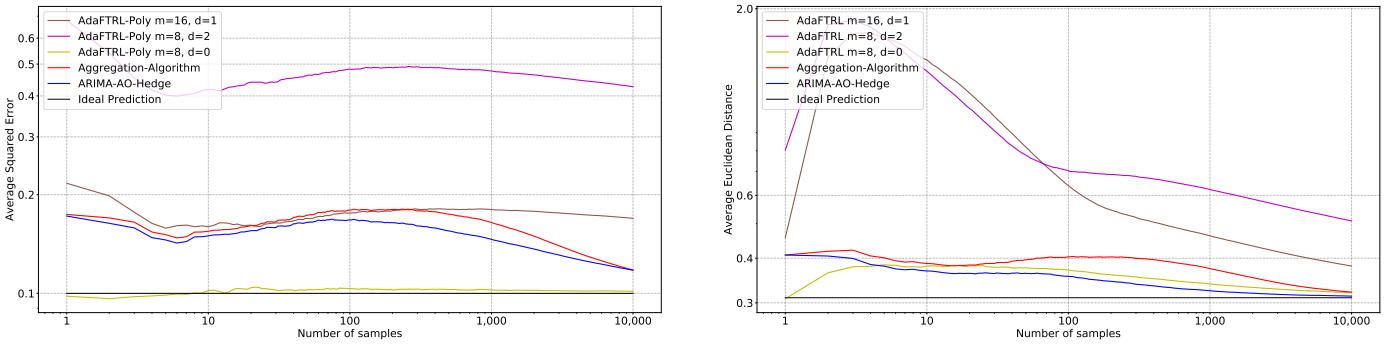
Figure 6. Results for Electricity Demand Data

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

5.3. Experiments for Online Model Selection

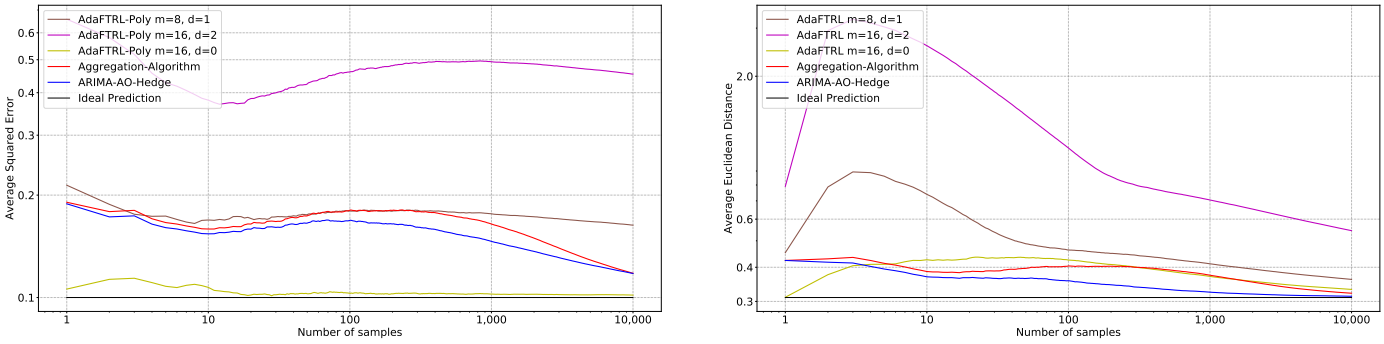
The performance of the two-level framework and Algorithm 4 for online model selection is demonstrated in Figures 7-12. We simultaneously maintain 96 AR(m) models of d-th order differencing, for $m = 1, \dots 32$ and $d = 0, \dots 2$, which are updated by Algorithm 2 and 3 for squared error and Euclidean distance respectively. The predictions generated by the AR models are aggregated using Algorithm 4 and the Aggregation Algorithm **AA** introduced in [13] with learning rate set to $\sqrt{T}$. We compare the average losses incurred by the aggregated predictions with these incurred by the best AR model. To show the impact of m and d, we also plot the average loss of some other sub-optimal AR models.

In all settings, **AO-Hedge** outperforms **AA**, although the differences are tiny in some of the experiments. We would like to stress again that the choice of the hyperparameters has a great impact on the performance of the AR model. In setting 1-3, the AR model with 0-th order differencing has the best performance although the data are generated using $d = 1$, which suggests that the prior knowledge about the data generation may not be helpful for the model selection in all cases. The experimental results also show that **AO-Hedge** has a performance similar to the best AR model.



(a) Squared Error **(b)** Euclidean Distance

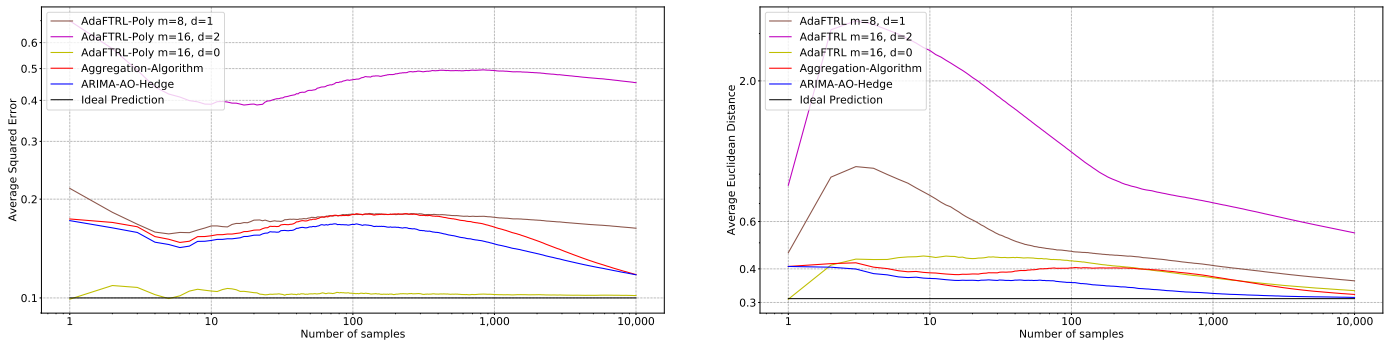
Figure 7. Model Selection in Setting 1



(a) Squared Error **(b)** Euclidean Distance

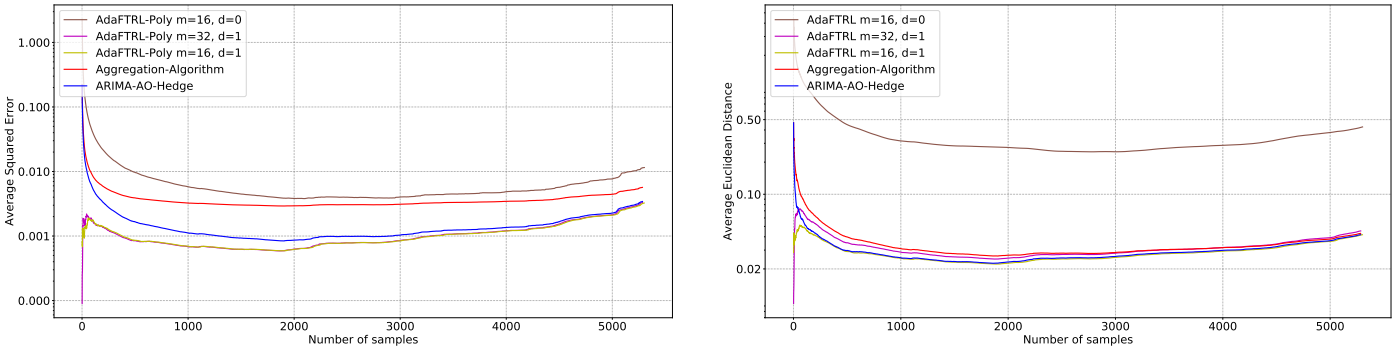
Figure 8. Model Selection in Setting 2

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1



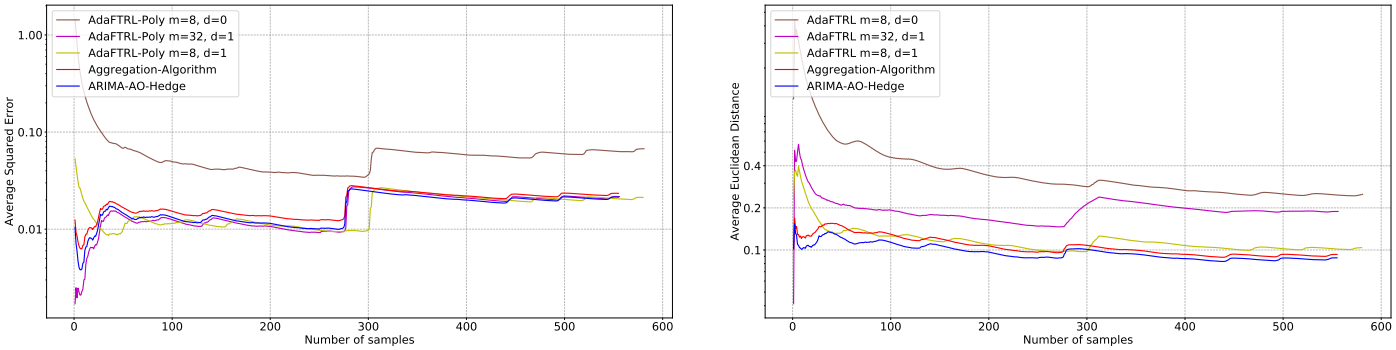
(a) Squared Error **(b)** Euclidean Distance

Figure 9. Model Selection in Setting 3



(a) Squared Error **(b)** Euclidean Distance

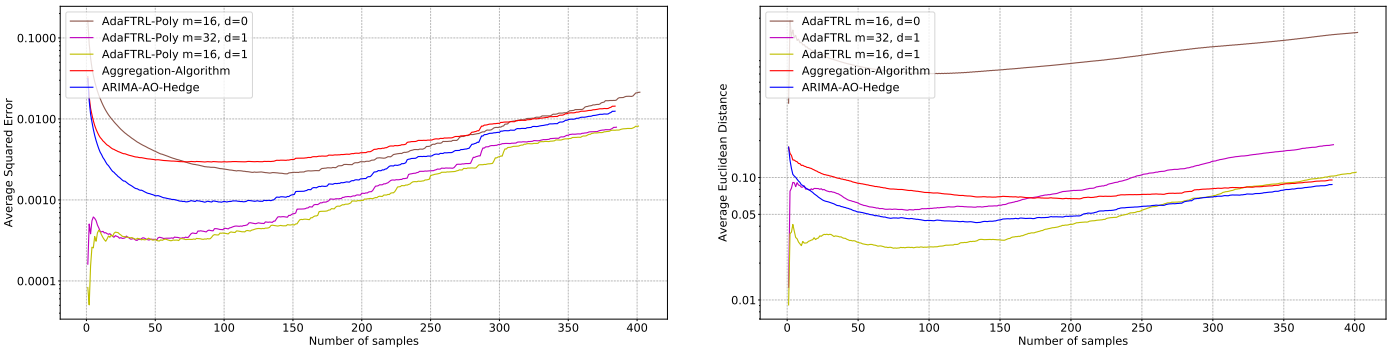
Figure 10. Model Selection for Stock Data



(a) Squared Error **(b)** Euclidean Distance

Figure 11. Model Selection for Google Flu

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1



(a) Squared Error **(b)** Euclidean Distance

Figure 12. Model Selection for Electricity Demand

6. Conclusion

We have proposed algorithms for fitting ARIMA models in an online manner without requiring prior knowledge or tuning hyperparameters. We have shown that the cumulative regret of our method grows sublinearly in the number of iterations and depends on the values of the time series. The comparison study on both synthetic and real-world datasets have suggested that the proposed algorithms have the performance on par with the well tuned state of the art algorithms.

There are still several remaining issues that we want to address in future research. First of all, it would be interesting to also develop a parameter-free algorithm for the cointegrated vector ARMA model. Secondly, we believe that the strong assumption on the β coefficient can be relaxed for multi-dimensional time series by generalising Lemma 2 in [7]. Furthermore, We are also interested in applying online learning to other time series models such as the (generalised) ARCH model [30]. Finally, the proposed algorithms need to be empirically analysed using more real world datasets and loss functions.

Author Contributions: Conceptualization, W.S.; methodology, W.S. and L.R.; validation, W.S., L.R. and F.S.; formal analysis, W.S.; investigation, W.S. and L.R; writing—original draft preparation, W.S. and L.R.; writing—review and editing, W.S., L.R., F.S. and S.A.; visualization, L.R.; supervision, F.S. and S.A.. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge support by the German Research Foundation and the Open Access Publication Fund of TU Berlin.

Data Availability Statement: The source code for generating synthetic data set, the implementation of the algorithms and the detailed information about our experiments are available on GitHub: https://github.com/OnlinePredictorTS/AOLForTimeSeries. The stock data are collected from https://finance.yahoo.com/. The google flu data are available in https://github.com/datalit/googleflutrends/. The detailed information about the electricity demand can be found in [31].

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

We prove lemma 1 in this section. Consider the ARIMA model given by

$$\nabla^d X_t(\alpha,\beta) = \sum_{i=1}^{p} \alpha_i \nabla^d X_{t-i} + \sum_{i=1}^{q} \beta_i \epsilon_{t-i} + \epsilon_t$$

with $\nabla^d X_t(\alpha,\beta) = \nabla^d X_t$ for $t \leq 0$. Let

$$X_t(\alpha,\beta) = \nabla^d X_t(\alpha,\beta) + \sum_{i=0}^{d-1} \nabla^i X_{t-1}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

be the t-th value generated by the ARIMA process. To prove Lemma 1, we generalised the proof provided in [6]. To remove the MA component, we first recursively define a growing process of the d-th order differencing

$$\nabla^d X_t^\infty(\alpha,\beta) = \sum_{i=1}^{p} \alpha_i \nabla^d X_{t-i} + \sum_{i=1}^{q} \beta_i (\nabla^d X_{t-i} - \nabla^d X_{t-i}^\infty(\alpha,\beta))$$

with $\nabla^d X_t^\infty(\alpha,\beta) = \nabla^d X_t$ for $t \le 0$. Let

$$X_t^\infty(\alpha,\beta) = \nabla^d X_t^\infty(\alpha,\beta) + \sum_{i=0}^{d-1} \nabla^i X_{t-1}$$

be the t-th value generated by this process.
The next lemma shows that it approximates an ARIMA(p,q,d) process.

Lemma A1. *For any α, β and $\{\epsilon_t\}$ satisfying **A1-A2**, we have, for $t = 1,\dots,T$,*

$$\|X_t^\infty(\alpha,\beta) - \tilde{X}_t(\alpha,\beta)\| \le (1-\epsilon)^{\frac{t}{q}} R.$$

Proof. First of all, we have

$$\begin{aligned} X_t^\infty(\alpha,\beta) - \tilde{X}_t(\alpha,\beta) =& \nabla^d X_t^\infty(\alpha,\beta) - \nabla^d \tilde{X}_t(\alpha,\beta) \\ =& \sum_{i=1}^{q} \beta_i (\nabla^d X_{t-i} - \nabla^d X_{t-i}^\infty(\alpha,\beta) - \epsilon_{t-i}), \end{aligned}$$

for $t \ge 0$. Define $Y_t = \nabla^d X_t - \nabla^d X_t^\infty(\alpha,\beta) - \epsilon_t$. W.l.o.g. we can assume $\|\epsilon_t\| \le R$ for $t \le 0$. Next, we prove by induction on t that $\|Y_\tau\| \le (1-\epsilon)^{\frac{\tau}{q}} R$ holds for all $\tau \le t$. For the induction basis, we have

$$\|Y_\tau\| = \|-\epsilon_t\| \le R$$

for all $\tau \le 0$. We assume the claim holds for some t, then we have

$$\begin{aligned} \|Y_{t+1}\| =& \|\nabla^d X_{t+1} - \nabla^d X_{t+1}^\infty(\alpha,\beta) - \epsilon_{t+1}\| \\ =& \|\nabla^d X_{t+1} - \sum_{i=1}^{p} \alpha_i \nabla^d X_{t+1-i} - \sum_{i=1}^{q} \beta_i \epsilon_{t+1-i} - \epsilon_{t+1}\| + \|\sum_{i=1}^{q} \beta_i Y_{t+1-i}\| \\ =& \sum_{i=1}^{q} \|Y_{t+1-i}\| \|\beta_i\|_{\text{op}} \\ \le& (1-\epsilon)^{\frac{t+1-q}{q}} R \sum_{i=1}^{q} \|\beta_i\|_{\text{op}} \\ \le& (1-\epsilon)^{\frac{t+1}{q}} R \end{aligned}$$

which concludes the induction. Finally, we have

$$\begin{aligned} \|X_t^\infty(\alpha,\beta) - \tilde{X}_t(\alpha,\beta)\| =& \|\sum_{i=1}^{q} \beta_i (\nabla^d X_{t-i}(\alpha,\beta) - \nabla^d X_{t-i}^\infty(\alpha,\beta) - \epsilon_{t-i})\| \\ \le& \sum_{i=1}^{q} \|\beta_i\|_{\text{op}} \|Y_{t-i}\| \\ \le& (1-\epsilon)(1-\epsilon)^{\frac{t-q}{q}} R \\ =& (1-\epsilon)^{\frac{t}{q}} R \end{aligned}$$

which is the claimed result. □

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

Next, we recursively define the following process

$$\nabla^d X_t^m(\alpha,\beta) = \sum_{i=1}^{p} \alpha_i \nabla^d X_{t-i} + \sum_{i=1}^{q} \beta_i (\nabla^d X_{t-i} - \nabla^d X_{t-i}^{m-i}(\alpha,\beta)), \tag{A1}$$

where $\nabla^d X_t^m(\alpha,\beta) = \nabla^d X_t$ for $m \le 0$. Let $\{X_t^m(\alpha,\beta)\}$ be the sequence generated as follows

$$X_t^m(\alpha,\beta) = \nabla^d X_t^m(\alpha,\beta) + \sum_{i=0}^{d-1} \nabla^i X_{t-1}. \tag{A2}$$

We show in the next lemma that it is close to $\{X_t^\infty(\alpha,\beta)\}$

Lemma A2. *For any α, β, $\{l_t\}$ and $\{\epsilon_t\}$ satisfying **A1-A2**, we have*

$$\|X_t^m(\alpha,\beta) - X_t^\infty(\alpha,\beta)\| \le \frac{2R}{T},$$

for $m = \frac{q \log T}{\log \frac{1}{1-\epsilon}}$.

Proof. Define $Z_t^m = \nabla^d X_t^m(\alpha,\beta) - \nabla^d X_t^\infty(\alpha,\beta)$. We prove by induction on m that

$$\|Z_t^{\tilde{m}}\| \le (1-\epsilon)^{\frac{\tilde{m}}{q}} 2R$$

holds for all $t = 1,\dots,T$ and $0 \le \tilde{m} \le m$. For $m = 0$, we have for $t = 1,\dots,T$

$$\begin{aligned}\|Z_t^0\| =& \|\nabla^d X_t^0(\alpha,\beta) - \nabla^d X_t^\infty(\alpha,\beta)\| \\ =& \|\nabla^d X_t - \nabla^d X_t^\infty(\alpha,\beta)\|\end{aligned}$$

By the definition of the stochastic process $\{\nabla^d X^\infty(\alpha,\beta)\}$, we have

$$\begin{aligned}&-\nabla^d X_t + \nabla^d X_t^\infty(\alpha,\beta) \\ =&-\nabla^d X_t + \sum_{i=1}^{p} \alpha_i \nabla^d X_{t-i} + \sum_{i=1}^{q} \beta_i (\nabla^d X_{t-i}(\alpha,\beta) - \nabla^d X_{t-i}^\infty(\alpha,\beta)) \\ =&-\nabla^d X_t + \sum_{i=1}^{p} \alpha_i \nabla^d X_{t-i} + \sum_{i=1}^{q} \beta_i \epsilon_{t-i} + \sum_{i=1}^{q} \beta_i (\nabla^d X_{t-i}(\alpha,\beta) - \nabla^d X_{t-i}^\infty(\alpha,\beta) - \epsilon_{t-i}) \\ =&\nabla^d \tilde{X}_t(\alpha,\beta) - \nabla^d X_t + \sum_{i=1}^{q} \beta_i (\nabla^d X_{t-i}(\alpha,\beta) - \nabla^d X_{t-i}^\infty(\alpha,\beta) - \epsilon_{t-i}) \\ =&\nabla^d \tilde{X}_t(\alpha,\beta) - \nabla^d X_t + \sum_{i=1}^{q} \beta_i Y_{t-i},\end{aligned}$$

where Y_{t-i} is defined as in the proof of Lemma A1. From the assumption, we have $\|\nabla^d \tilde{X}_t(\alpha,\beta) - \nabla^d X_t\| = \|\epsilon_t\| \le R$, and, as we have proved in Lemma A1, $\|Y_t\| \le R$ holds.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

Therefore, we obtain $\|Z_t^0\| \leq 2R$, which is the induction basis. Next, assume the claim holds for all $0, \ldots, m-1$. Then we have

$$\begin{aligned}
\|Z_t^m\| =& \|\sum_{i=1}^{q} \beta^i (\nabla^d X_{t-i} - \nabla^d X_{t-i}^{m-i}(\alpha,\beta) - \nabla^d X_{t-i} + \nabla^d X_{t-i}^{\infty}(\alpha,\beta))\| \\
\leq& \|\sum_{i=1}^{q} \beta_i (\nabla^d X_{t-i}^{\infty}(\alpha,\beta) - \nabla^d X_{t-i}^{m-i}(\alpha,\beta))\| \\
\leq& \sum_{i=1}^{m} \|\beta_i (\nabla^d X_{t-i}^{\infty}(\alpha,\beta) - \nabla^d X_{t-i}^{m-i}(\alpha,\beta))\| \\
&+ \sum_{i=m+1}^{q} \|\beta_i (\nabla^d X_{t-i}^{\infty}(\alpha,\beta) - \nabla^d X_{t-i})\|
\end{aligned}$$

From the induction hypothesis, we have

$$\|\nabla^d X_{t-i}^{\infty}(\alpha,\beta) - \nabla^d X_{t-i}^{m-i}(\alpha,\beta)\| \leq (1-\epsilon)^{\frac{m-i}{q}} 2R.$$

From the proof of the induction basis, we have

$$\sum_{i=m+1}^{q} \|\beta_i (\nabla^d X_{t-i}^{\infty}(\alpha,\beta) - \nabla^d X_{t-i})\| \leq 2R \sum_{i=m+1}^{q} \|\beta_i\|_{\text{op}}.$$

Therefore, $\|Z_t^m\|$ can be further bounded using

$$\begin{aligned}
\|Z_t^m\| \leq& 2R \sum_{i=1}^{m} \|\beta^i\|_{\text{op}} (1-\epsilon)^{\frac{m-i}{q}} + 2R \sum_{i=m+1}^{q} \|\beta^i\|_{\text{op}} \\
\leq& 2R \sum_{i=1}^{m} \|\beta^i\|_{\text{op}} (1-\epsilon)^{\frac{m-i}{q}} + 2R \sum_{i=m+1}^{q} \|\beta^i\|_{\text{op}} (1-\epsilon)^{\frac{m-i}{q}} \\
\leq& (1-\epsilon)^{\frac{m-q}{q}} 2R \sum_{i=1}^{q} \|\beta^i\|_{\text{op}} \\
\leq& (1-\epsilon)^{\frac{m}{q}} 2R,
\end{aligned}$$

Choosing $m \geq \frac{q \log T}{\log \frac{1}{1-\epsilon}} = q \log_{1-\epsilon}(T)^{-1}$, we have

$$\|X_t^m(\alpha,\beta) - X_t^{\infty}(\alpha,\beta)\| \leq \frac{2R}{T},$$

which is the claimed result. □

This process of the d-th order differencing is actually an integrated AR$(m+p)$ process with order d, which is shown in the following lemma.

Lemma A3. *For any data sequence $\{X_t^m(\alpha,\beta)\}$ generated by a process of the d-th order differencing given by A1 and A2 there is a $\gamma \in \mathcal{L}(\mathbb{X},\mathbb{X})^{m+p}$ such that*

$$\sum_{i=1}^{m+p} \gamma_i \nabla^d X_{t-i} + \sum_{i=0}^{d-1} \nabla^i X_{t-1} = X_t^m(\alpha,\beta)$$

holds for all t.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

Proof. Let $\{\nabla^d X_t^m(\alpha,\beta)\}$ be the sequence generated by A1. We prove by induction on m that, for all $\tilde{m} \leq m$, there is a $\gamma \in \mathcal{L}(\mathbb{X},\mathbb{X})^{\tilde{m}+p}$, such that

$$\nabla^d X_t^{\tilde{m}}(\alpha,\beta) = \sum_{i=1}^{\tilde{m}+p} \gamma_i \nabla^d X_{t-i},$$

holds for all α and β. The induction basis follows directly from the definition that

$$\nabla^d X_t^0(\alpha,\beta) = \sum_{i=1}^{p} \alpha_i \nabla^d X_{t-i}.$$

Assume that the claim holds for some m. Let α_i be the zero linear functional for $i > p$ and β_i be the zero linear functional for $i > q$. Then we have

$$\begin{aligned}
&\nabla^d X_t^{m+1}(\alpha,\beta)\\
=&\sum_{i=1}^{p} \alpha_i \nabla^d X_{t-i} + \sum_{i=1}^{q} \beta_i (\nabla^d X_{t-i} - \nabla^d X_{t-i}^{m+1-i}(\alpha,\beta))\\
=&\sum_{i=1}^{p} \alpha_i \nabla^d X_{t-i} + \sum_{i=1}^{m+1} \beta_i \nabla^d X_{t-i} - \sum_{i=1}^{m+1} \beta_i \nabla^d X_{t-i}^{m+1-i}(\alpha,\beta)\\
=&\sum_{i=1}^{p} \alpha_i \nabla^d X_{t-i} + \sum_{i=1}^{m+1} \beta_i \nabla^d X_{t-i} - \sum_{i=1}^{m+1} \beta_i \sum_{j=1}^{m+1-i+p} \gamma_j^{m+1-i} \nabla^d X_{t-i-j}\\
=&\sum_{i=1}^{p} \alpha_i \nabla^d X_{t-i} + \sum_{i=1}^{m+1} \beta_i \nabla^d X_{t-i} - \sum_{i=1}^{m+p+1} (\sum_{j=1}^{m+1} \beta_j \sum_{k=1}^{i-j} \gamma_k^{m+1-j}) \nabla^d X_{t-i},
\end{aligned}$$

where the second equality follows from the fact that $\beta_i(\nabla^d X_{t-i} - \nabla^d X_{t-i}^{m+1-i}(\alpha,\beta)) = 0$ for $i > m+1$, the third line uses the induction hypothesis and the last line is obtained by rearranging and setting $\sum_{i=m}^{n} a_i = 0$ for $m > n$. The induction step is obtained by setting

$$\gamma_i^{m+1} = \alpha_i + \beta_i - \sum_{j=1}^{m+1} \beta_j \sum_{k=1}^{i-j} \gamma_k^{m+1-j}$$

for $i = 1,\ldots,m+p+1$, and the claimed result follows. □

Finally, we prove Lemma 1 by combining the results.

Proof of Lemma 1. From Lemma A1, A2 and A3, there is some $\gamma \in \mathcal{L}(\mathbb{X},\mathbb{X})^m$ with $m \geq \frac{q \log T}{\log \frac{1}{1-\epsilon}} + p$ such that

$$\begin{aligned}
&\|\nabla^d X_t(\gamma) - \nabla^d \tilde{X}_t(\alpha,\beta)\|\\
=&\|\nabla^d X_t^m(\gamma) - \nabla^d \tilde{X}_t(\alpha,\beta)\|\\
\leq&\|\nabla^d X_t^m(\gamma) - \nabla^d X_t^{\infty}(\alpha,\beta)\| + \|\nabla^d X_t^{\infty}(\gamma) - \nabla^d \tilde{X}_t(\alpha,\beta)\|\\
\leq&(1-\epsilon)^{\frac{t}{q}} R + \frac{2R}{T},
\end{aligned}$$

which is the claimed result. □

Appendix B

In this section, we prove the theorems in Section 4. The required notation are summarised in Appendix C. We have applied some important properties of convex functions and their convex conjugate defined on a general vector space, which can be found in [28].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

The proposed algorithms are instances of the *Adaptive Optimistic Follow The Regularised Leader* (**AO-FTRL**) [10], which is described in Algorithm 5.

Algorithm 5 AO-FTRL

Input: closed convex set $\mathcal{W} \subseteq \mathbb{X}$
Initialise: θ_1 arbitrary
for $t = 1$ to T **do**
  Get hint h_t
  $w_t = \nabla \psi_t^*(\theta_t - h_t)$
  Observe $g_t \in \mathbb{X}_*$
  $\theta_{t+1} = \theta_t - g_t$
end for

Lemma A4. *We run **AO-FTRL** with closed convex regularisers $\psi_1, \dots, \psi_T$ defined on $\mathcal{W} \subseteq \mathbb{X}$ satisfying $\psi_t(w) \leq \psi_{t+1}(w)$s for all $w \in \mathcal{W}$ and $t = 1, \dots, T$. Then, for all $u \in \mathcal{W}$, we have*

$$\sum_{t=1}^{T} g_t(w_t - u) \leq \psi_{T+1}(u) + \psi_1^*(\theta_1) + \sum_{t=1}^{T} \mathcal{B}_{\psi_t^*}(\theta_{t+1}, \theta_t - h_t),$$

where $\mathcal{B}_{\psi_t^*}(\theta_{t+1}, \theta_t - h_t)$ is the Bregman divergence associated with ψ_t^*.

Proof. W.l.o.g. we assume $h_{T+1} = 0$, since it is not involved in the algorithm. Then we have

$$\begin{aligned}
&\sum_{t=1}^{T} (\psi_{t+1}^*(\theta_{t+1} - h_{t+1}) - \psi_t^*(\theta_t - h_t)) \\
=&\psi_{T+1}^*(\theta_{T+1} - h_{T+1}) - (\theta_1 - h_1)w_1 + \psi_1(w_1) \\
\geq&(\theta_{T+1} - h_{T+1})u - \psi_{T+1}(u) + h_1 w_1 - \theta_1 w_1 + \psi_1(w_1) \\
\geq&\theta_{T+1}u - \psi_{T+1}(u) + h_1 w_1 - \sup_{w \in \mathcal{W}} (\theta_1 w_1 - \psi_1(w_1)) \\
=&-\sum_{t=1}^{T} g_t u - \psi_{T+1}(u) + h_1 w_1 - \psi_1^*(\theta_1).
\end{aligned}$$

Furthermore, we have

$$\begin{aligned}
&\psi_{t+1}^*(\theta_{t+1} - h_{t+1}) - \psi_t^*(\theta_t - h_t) \\
=&\psi_{t+1}^*(\theta_{t+1} - h_{t+1}) - \psi_t^*(\theta_{t+1}) + \psi_t^*(\theta_{t+1}) - \psi_t^*(\theta_t - h_t) \\
\leq&(\theta_{t+1} - h_{t+1})w_{t+1} - \psi_{t+1}(w_{t+1}) - \theta_{t+1}w_{t+1} + \psi_t(w_{t+1}) + \psi_t^*(\theta_{t+1}) - \psi_t^*(\theta_t - h_t) \\
\leq&\psi_t^*(\theta_{t+1}) - \psi_t^*(\theta_t - h_t) - h_{t+1}w_{t+1}
\end{aligned}$$

Combining the inequalities above, rearranging and adding $\sum_{t=1}^{T} \langle g_t, w_t \rangle$ to both sides, we obtain

$$\begin{aligned}
&\sum_{t=1}^{T} g_t(w_t - u) \\
\leq&\psi_{T+1}(u) + \psi_1^*(\theta_1) + \sum_{t=1}^{T} (\psi_t^*(\theta_{t+1}) - \psi_t^*(\theta_t - h_t) + g_t w_t - h_t w_t) \\
=&\psi_{T+1}(u) + \psi_1^*(\theta_1) + \sum_{t=1}^{T} (\psi_t^*(\theta_{t+1}) - \psi_t^*(\theta_t - h_t) - (\theta_{t+1} - \theta_t + h_t)\nabla \psi_t^*(\theta_t - h_t)) \\
=&\psi_{T+1}(u) + \psi_1^*(\theta_1) + \sum_{t=1}^{T} \mathcal{B}_{\psi_t^*}(\theta_{t+1}, \theta_t - h_t),
\end{aligned}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

which is the claimed result. □

Proof of Theorem 1. First of all, since we have

$$\sum_{t=1}^{T} l_t(\tilde{X}_t(\gamma_t)) - l_t(\tilde{X}_t(\gamma)) \leq \sum_{t=1}^{T}\sum_{i=1}^{m} g_{i,t}(\gamma_{i,t} - \gamma_i) = \sum_{i=1}^{m}(\sum_{t=1}^{T} g_{i,t}(\gamma_{i,t} - \gamma_i)),$$

the overall regret can be considered as the sum of the regrets $\sum_{t=1}^{T} g_{i,t}(\gamma_{i,t} - \gamma_i)$. Next, we analyse the regret of each $i = 1, \dots m$. Define $\psi_{i,t}(\gamma_i) = \frac{\eta_{i,t}}{2}\|\gamma_i\|_F^2$. It is easy to verify $\gamma_{i,t} \in \partial\psi_{i,t}^*(\theta_{i,t})$ for $t = 1, \dots, T$. Applying lemma A4 with $h_t = 0$, we obtain

$$\sum_{t=1}^{T} g_{i,t}(\gamma_{i,t} - \gamma_i) \leq \psi_{i,T+1}(\gamma_i) + \psi_{i,1}^*(\theta_{i,1}) + \sum_{t=1}^{T} \mathcal{B}_{\psi_{i,t}^*}(\theta_{i,t+1}, \theta_{i,t}).$$

From the updating rule of $G_{i,t}$, we have $g_{i,t} = 0$ for $G_{i,t} = 0$. Let t_0 be the smallest index such that $G_{i,t_0} > 0$. Then we have

$$\sum_{t=1}^{T} \mathcal{B}_{\psi_{i,t}^*}(\theta_{i,t+1}, \theta_{i,t}) = \sum_{t=t_0}^{T} \mathcal{B}_{\psi_{i,t}^*}(\theta_{i,t+1}, \theta_{i,t}).$$

For $G_{i,t} > 0$, $\psi_{i,t}$ is $\eta_{i,t}$-strongly convex with respect to $\|\cdot\|_F$. From the duality of strong convexity and strong smoothness (see Proposition 2 in [28]), we have

$$\sum_{t=t_0}^{T} \mathcal{B}_{\psi_{i,t}^*}(\theta_{i,t+1}, \theta_{i,t}) \leq \sum_{t=t_0}^{T} \frac{1}{2\eta_{i,t}}\|g_{i,t}\|_F^2 = \sum_{t=t_0}^{T} \frac{\|g_{i,t}\|_F^2}{2\sqrt{\sum_{s=1}^{t-1}\|g_{i,s}\|_F^2 + (L_t G_{i,t})^2}}.$$

From the definition of *Frobenius norm*, we have

$$\|g_{i,t}\|_F^2 = \|h_t \nabla^d X_{t-i}^\top\|_F^2 = \|h_t\|_2^2 \|\nabla^d X_{t-i}\|_2^2 \leq \frac{\|h_t\|_2^2}{L_t^2} L_t^2 G_{i,t}^2.$$

Then, we obtain

$$\begin{aligned}
\sum_{t=t_0}^{T} \frac{\|g_{i,t}\|_F^2}{2\sqrt{\sum_{s=1}^{t-1}\|g_{i,s}\|_F^2 + (L_t G_{i,t})^2}} &\leq \sum_{t=t_0}^{T} \frac{\max\{1, \frac{\|h_t\|_2}{L_t}\}\|g_{i,t}\|_F^2}{2\sqrt{\sum_{s=1}^{t}\|g_{i,s}\|_F^2}} \\
&\leq \max\{1, \frac{\|h_1\|_2}{L_1}, \dots, \frac{\|h_T\|_2}{L_T}\}\sqrt{\sum_{t=1}^{T}\|g_{i,t}\|_F^2} \\
&\leq (1 + \frac{L_{T+1}}{L_1})\sqrt{\sum_{t=1}^{T}\|g_{i,t}\|_F^2} \\
&\leq (L_{T+1} + \frac{L_{T+1}^2}{L_1})\sqrt{\sum_{t=1}^{T}\|\nabla^d X_{t-i}\|_2^2},
\end{aligned}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

where the second inequality uses the lemma 4 in [28] and the last inequality follows from the fact that $\|g_{i,t}\|_F \leq L_t \|\nabla^d X_{t-i}\|_2 \leq L_{T+1}\|\nabla^d X_{t-i}\|_2$. Furthermore, we have

$$\begin{aligned}\psi_{i,T+1}(\gamma_i) \leq & \frac{\|\gamma_i\|_F^2}{2}\sqrt{\sum_{t=1}^{T}\|g_{i,t}\|_F^2} + \frac{L_{T+1}G_{i,T+1}\|\gamma_i\|_F^2}{2} \\ \leq & \frac{\|\gamma_i\|_F^2 L_{T+1}}{2}\sqrt{\sum_{t=1}^{T}\|\nabla^d X_{t-i}\|_2^2} + \frac{L_{T+1}G_{i,T+1}\|\gamma_i\|_F^2}{2},\end{aligned}$$

and $\psi_{i,1}^*(\theta_{i,1}) \leq \frac{\|\theta_{i,1}\|_F}{2}$. Adding up from 1 to m, we have

$$\begin{aligned}&\sum_{t=1}^{T} l_t(\tilde{X}_t(\gamma_t)) - l_t(\tilde{X}_t(\gamma)) \\ \leq & \sum_{i=1}^{m}\left(\frac{\|\gamma_i\|_F^2 L_{T+1}}{2} + L_{T+1} + \frac{L_{T+1}^2}{L_1}\right)\sqrt{\sum_{t=1}^{T}\|\nabla^d X_{t-i}\|_2^2} \\ & + \sum_{i=1}^{m}\frac{L_{T+1}G_{i,T+1}\|\gamma_i\|_F^2 + \|\theta_{i,1}\|_F}{2}\end{aligned}$$

□

Proof of Theorem 2. Define $\psi_t(\gamma) = \frac{\lambda_t\|\gamma\|^4}{4} + \frac{\lambda_t\|\gamma\|^2}{2}$. First of all, it is easy to verify that $\gamma_t \in \partial\psi_t^*(\theta_t)$. Applying lemma A4 with $h_t = 0$, we have

$$\sum_{t=1}^{T}\langle g_t x_t^\top, \gamma_t - \gamma\rangle_F \leq \psi_{T+1}(\gamma) + \psi_1^*(\theta_1) + \sum_{t=1}^{T}\mathcal{B}_{\psi_t^*}(\theta_{t+1},\theta_t). \tag{A3}$$

Define $v_t \in \partial\psi_{t+1}^*(\theta_t)$. Then we have

$$\begin{aligned}\mathcal{B}_{\psi_t^*}(\theta_{t+1},\theta_t) =& \psi_t^*(\theta_{t+1}) - \psi_t^*(\theta_t) - \langle\gamma_t, \theta_{t+1}-\theta_t\rangle_F \\ =& \langle\theta_{t+1}, v_t\rangle_F - \psi_t(v_t) - \langle\theta_t,\gamma_t\rangle_F + \psi_t(\gamma_t) - \langle\gamma_t,\theta_{t+1}-\theta_t\rangle_F \\ =& \langle\theta_{t+1}, v_t\rangle_F - \psi_t(v_t) + \psi_t(\gamma_t) - \langle\gamma_t,\theta_{t+1}\rangle_F \\ =& \langle\theta_{t+1}, v_t-\gamma_t\rangle_F - \psi_t(v_t) + \psi_t(\gamma_t) \\ =& \langle g_t x_t^\top, \gamma_t - v_t\rangle_F - \psi_t(v_t) + \psi_t(\gamma_t) + \langle\theta_t, v_t-\gamma_t\rangle_F \\ =& \langle g_t x_t^\top, \gamma_t - v_t\rangle_F - \mathcal{B}_{\psi_t}(v_t,\gamma_t) \\ =& \langle\gamma_t x_t x_t^\top, \gamma_t - v_t\rangle_F + \langle -\nabla^d X_t x_t^\top, \gamma_t - v_t\rangle_F - \mathcal{B}_{\psi_t}(v_t,\gamma_t) \\ =& \langle\gamma_t x_t x_t^\top, \gamma_t - v_t\rangle_F - \mathcal{B}_{\tilde{\psi}_t}(v_t,\gamma_t) \\ & + \langle -\nabla^d X_t x_t^\top, \gamma_t - v_t\rangle_F - \mathcal{B}_{\bar{\psi}_t}(v_t,\gamma_t),\end{aligned} \tag{A4}$$

where we define $\tilde{\psi}_t(\gamma) = \frac{\lambda_t}{4}\|\gamma\|_F^4$ and $\bar{\psi}_t(\gamma) = \frac{\eta_t}{2}\|\gamma\|_F^2$. From the properties of the Frobenius norm, we have

$$\begin{aligned}\langle\gamma_t x_t x_t^\top, \gamma_t - v_t\rangle_F \leq & \|\gamma_t x_t x_t^\top\|_F\|\gamma_t - v_t\|_F \\ \leq & \|x_t\|_2^2\|\gamma_t\|_F\|\gamma_t - v_t\|_F\end{aligned}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

Following the idea of [32], we can upper bound $\|\gamma_t\|_F^2\|\gamma_t - v_t\|_F^2$ as follows

$$\begin{aligned}
&\frac{\lambda_t}{2}\|\gamma_t\|_F^2\|\gamma_t - v_t\|_F^2 \\
=&\frac{\lambda_t}{2}\|\gamma_t\|_F^2(\|\gamma_t\|_F^2 + \|v_t\|_F^2 - 2\langle\gamma_t, v_t\rangle_F) \\
\leq&\frac{\lambda_t}{4}(\|\gamma_t\|_F^4 + \|v_t\|_F^4 - 2\|\gamma_t\|_F^2\|v_t\|_F^2) + \frac{\lambda_t}{2}\|\gamma_t\|_F^2(\|\gamma_t\|_F^2 + \|v_t\|_F^2 - 2\langle\gamma_t, v_t\rangle_F) \\
=&\frac{\lambda_t}{4}\|v_t\|_F^4 + \frac{3\lambda_t}{4}\|\gamma_t\|_F^4 - \lambda_t\|\gamma_t\|_F^2\langle\gamma_t, v_t\rangle_F \\
=&\frac{\lambda_t}{4}\|v_t\|_F^4 - \frac{\lambda_t}{4}\|\gamma_t\|_F^4 + \lambda_t\|\gamma_t\|_F^2\langle\gamma_t, \gamma_t\rangle_F - \lambda_t\|\gamma_t\|_F^2\langle\gamma_t, v_t\rangle_F \\
=&\frac{\lambda_t}{4}\|v_t\|_F^4 - \frac{\lambda_t}{4}\|\gamma_t\|_F^4 - \lambda_t\|\gamma_t\|_F^2\langle\gamma_t, v_t - \gamma_t\rangle_F \\
=&\mathcal{B}_{\tilde{\psi}_t}(v_t, \gamma_t)
\end{aligned}$$

Thus, for $\lambda_t \neq 0$, we have

$$\begin{aligned}
\langle\gamma_t x_t x_t^\top, \gamma_t - v_t\rangle_F - \mathcal{B}_{\tilde{\psi}_t}(v_t, \gamma_t) \leq& 2\sqrt{\frac{\|x_t\|_2^4}{2\lambda_t}\mathcal{B}_{\tilde{\psi}_t}(v_t, \gamma_t)} - \mathcal{B}_{\tilde{\psi}_t}(v_t, \gamma_t) \\
\leq& \frac{\|x_t\|_2^4}{2\lambda_t},
\end{aligned}$$

where, the second inequality uses the fact that $2ab - b^2 \leq a^2$. Let t_0 be the smallest index such that $\lambda_{t_0} > 0$. Then we have

$$\begin{aligned}
&\sum_{t=1}^{T}(\langle\gamma_t x_t x_t^\top, \gamma_t - v_t\rangle_F - \mathcal{B}_{\tilde{\psi}_t}(v_t, \gamma_t)) \\
\leq&\sum_{t=t_0}^{T}\frac{\|x_t\|_2^4}{2\lambda_t} \\
=&\sum_{t=t_0}^{T}\frac{\|x_t\|_2^4}{2\sqrt{\sum_{s=1}^{t}\|x_t\|_2^4}} \\
\leq&\sqrt{\sum_{t=1}^{T}\|x_t\|_2^4},
\end{aligned} \tag{A5}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

where the last inequality uses lemma 4 in [28]. Similarly, let t_1 be the smallest index such that $\eta_{t_0} > 0$. Then we obtain the upper bound

$$\begin{aligned}
&\sum_{t=1}^{T}(\langle -\nabla^d X_t x_t^\top, \gamma_t - v_t\rangle_F - \mathcal{B}_{\bar{\psi}_t}(v_t, \gamma_t)) \\
\leq &\sum_{t=1}^{T}(\|\nabla^d X_t x_t^\top\|_F\|\gamma_t - v_t\|_F - \mathcal{B}_{\bar{\psi}_t}(v_t, \gamma_t)) \\
\leq &\sum_{t=t_1}^{T}(\sqrt{\frac{2\|\nabla^d X_t x_t^\top\|_F^2}{\eta_t}\mathcal{B}_{\bar{\psi}_t}(v_t, \gamma_t)} - \mathcal{B}_{\bar{\psi}_t}(v_t, \gamma_t)) \\
\leq &\sum_{t=t_1}^{T}(2\sqrt{\frac{\|\nabla^d X_t x_t^\top\|_F^2}{2\eta_t}\mathcal{B}_{\bar{\psi}_t}(v_t, \gamma_t)} - \mathcal{B}_{\bar{\psi}_t}(v_t, \gamma_t)) \\
\leq &\sum_{t=t_1}^{T}\frac{\|\nabla^d X_t x_t^\top\|_F^2}{2\eta_t} \\
= &\sum_{t=t_1}^{T}\frac{\|\nabla^d X_t x_t^\top\|_F^2}{2\sqrt{\sum_{s=1}^{t-1}\|\nabla^d X_s x_s^\top\|_F^2 + L_t^2\|x_t\|_2^2}} \\
\leq &\max\{1, \frac{\|\nabla^d X_1 x_1^\top\|_F}{G_1}, \ldots, \frac{\|\nabla^d X_T x_T^\top\|_F}{G_T}\}\sum_{t=t_1}^{T}\frac{\|\nabla^d X_t x_t^\top\|_F^2}{2\sqrt{\sum_{s=1}^{t}\|\nabla^d X_s x_s^\top\|_F^2}} \\
\leq &\max\{1, \frac{\|\nabla^d X_1 x_1^\top\|_F}{G_1}, \ldots, \frac{\|\nabla^d X_T x_T^\top\|_F}{G_T}\}\sqrt{\sum_{t=1}^{T}\|\nabla^d X_t x_t^\top\|_F^2} \\
\leq &(1 + \frac{G_{T+1}}{G_1})\sqrt{\sum_{t=1}^{T}\|\nabla^d X_t x_t^\top\|_F^2}
\end{aligned} \tag{A6}$$

Combining A3, A4, A5 and A6, we obtain

$$\begin{aligned}
\sum_{t=1}^{T}\langle g_t x_t^\top, \gamma_t - \gamma\rangle_F \leq &\frac{(\sqrt{m}G_{T+1}^2 + \|\theta_1\|_F)\|\gamma\|_F^2}{2} + \psi_1^*(\theta_1) + (1 + \frac{\|\gamma\|_F^4}{4})\sqrt{\sum_{t=1}^{T}\|x_t\|_2^4} \\
&+ (1 + \frac{G_{T+1}}{G_1} + \frac{\|\gamma\|_F^2}{2})\sqrt{\sum_{t=1}^{T}\|\nabla^d X_t x_t^\top\|_F^2}.
\end{aligned}$$

For $\theta_1 \neq 0$, it is easy to verify that $\psi_1^*(\theta_1) \leq \langle w_1, \theta_1\rangle_F \leq \frac{\|\theta_1\|_F^2}{\eta_1} \leq \|\theta_1\|_F$. By putting this in the inequality above, we obtain the claimed result. □

Appendix B.1 Proof of Theorem 3

Proof. Define

$$\psi_t : \Delta \to \mathbb{R}, w \mapsto \eta_t \sum_{k\in I_w}^{K} w_k \log w_k + \eta_t \log K,$$

where $I_w = \{i = 1, \ldots, k | w_i \neq 0\}$. It can be verified that $w_t \in \partial\psi_t^*(\theta_t)$. Applying Lemma A4, we obtain

$$\sum_{t=1}^{T} z_t^\top(w_t - u) \leq \psi_{T+1}(u) + \psi_1^*(\theta_1) + \sum_{t=1}^{T}\mathcal{B}_{\psi_t^*}(\theta_{t+1}, \theta_t - h_t).$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

From the definition of ψ_t, it follows that $\psi_{T+1}(u) \leq \sqrt{\frac{\log K}{2}\sum_{t=1}^{T}\|z_t - h_t\|_\infty^2}$ and $\psi_1^*(\theta_1) = 0$ hold. Define $v_t \in \partial\psi_t^*(\theta_{t+1})$. Next, we bound the third term as follows

$$
\begin{aligned}
&\mathcal{B}_{\psi_t^*}(\theta_{t+1}, \theta_t - h_t) \\
=&\psi_t^*(\theta_{t+1}) - \psi_t^*(\theta_t - h_t) - (h_t - z_t)^\top w_t \\
=&\theta_{t+1}^\top v_t - \psi_t(v_t) - (\theta_t - h_t)^\top w_t + \psi_t(w_t) - (h_t - z_t)^\top w_t \\
=&(h_t - z_t)^\top (v_t - w_t) - (\psi_t(v_t) - \psi_t(w_t) - (\theta_t - h_t)^\top (v_t - w_t)) \\
=&(h_t - z_t)^\top (v_t - w_t) - \mathcal{B}_{\psi_t}(v_t, w_t) \\
=&(h_t - z_t)^\top (v_t - w_t) - \eta_{t+1}\|v_t - w_t\|_1^2 + \eta_{t+1}\|v_t - w_t\|_1^2 - \mathcal{B}_{\psi_t}(v_t, w_t) \\
\leq&(h_t - z_t)^\top (v_t - w_t) - \eta_{t+1}\|v_t - w_t\|_1^2 + (\eta_{t+1} - \eta_t)\|v_t - w_t\|_1^2 \\
\leq&\|h_t - z_t\|_\infty \|v_t - w_t\|_1 - \eta_{t+1}\|v_t - w_t\|_1^2 + 4(\eta_{t+1} - \eta_t) \\
\leq&\frac{\|h_t - z_t\|_\infty^2}{4\eta_{t+1}} + 4(\eta_{t+1} - \eta_t),
\end{aligned}
$$

where the first inequality uses the fact that ψ_t is $2\eta_t$ strongly convex w.r.t. $\|\cdot\|_1$. Adding up from 1 to T, we have

$$
\begin{aligned}
\sum_{t=1}^{T}\mathcal{B}_{\psi_t^*}(\theta_{t+1}, \theta_t - h_t) &\leq \sum_{t=1}^{T}\left(\frac{\|h_t - z_t\|_\infty^2}{4\eta_{t+1}} + 4(\eta_{t+1} - \eta_t)\right) \\
&\leq \sqrt{\frac{\log K}{2}\sum_{t=1}^{T}\|h_t - z_t\|_\infty^2} + 4\eta_{T+1} \\
&\leq \sqrt{\frac{\log K}{2}\sum_{t=1}^{T}\|h_t - z_t\|_\infty^2} + \sqrt{\frac{8}{\log K}\sum_{t=1}^{T}\|h_t - z_t\|_\infty^2}
\end{aligned}
$$

Combining the inequalities, we obtain

$$
\begin{aligned}
&\sum_{t=1}^{T} l(X_t, \sum_{i=1}^{K} w_{i,t}\tilde{X}_t^i) - \sum_{t=1}^{T} l(X_t, \tilde{X}_t^k) \\
\leq&\sum_{t=1}^{T}\sum_{i=1}^{K} w_{i,t} l(X_t, \tilde{X}_t^i) - \sum_{t=1}^{T} l(X_t, \tilde{X}_t^k) \\
=&\sum_{t=1}^{T} w_t^\top z_t - \sum_{t=1}^{T} l(X_t, \tilde{X}_t^k) \\
\leq&(\sqrt{2\log K} + \sqrt{\frac{8}{\log K}})\sqrt{\sum_{t=1}^{T}\|h_t - z_t\|_\infty^2},
\end{aligned}
$$

where the first inequality follows from *Jensen's inequality*. Further more, if l is L-Lipschitz in its first argument, then we have

$$
\|h_t - z_t\|_\infty = \max_{i\in\{1,\ldots,K\}} |z_{i,t} - h_{i,t}| \leq L\|\nabla^d X_t\|_2.
$$

Finally, we obtain the regret upper bound

$$
\sum_{t=1}^{T} l(X_t, \sum_{i=1}^{K} w_{i,t}\tilde{X}_t^i) - \sum_{t=1}^{T} l(X_t, \tilde{X}_t^k) \leq \left(\sqrt{2\log K} + \sqrt{\frac{8}{\log K}}\right)\sqrt{\sum_{t=1}^{T} L^2\|\nabla^d X_t\|_2^2},
$$

which is the claimed result. □

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 April 2021 doi:10.20944/preprints202104.0601.v1

Appendix C

We summarise the main notations used throughout the article in Table 1.

Table 1. Nomenclature

$(\mathbb{X}, \|\cdot\|)$	finite dimensional norm space
$(\mathbb{X}_*, \|\cdot\|_*)$	the dual space with dual norm of $(\mathbb{X}, \|\cdot\|)$
$\mathcal{L}(\mathbb{X}, \mathbb{X})$	vector space of bounded linear operators
$\|\alpha\|_{\text{op}} = \sup_{x \in \mathbb{X}, x \neq 0} \frac{\|\alpha x\|}{\|x\|}$	the operator norm of $\alpha \in \mathcal{L}(\mathbb{X}, \mathbb{X})$
$\|x\|_2 = \sqrt{\sum_{i=1}^d x_i^2}$	2 norm for $x \in \mathbb{R}^d$
$\|x\|_1 = \sum_{i=1}^d \|x_i\|$	1 norm for $x \in \mathbb{R}^d$
$\|x\|_\infty = \max\{\|x_1\|, \ldots, \|x_d\|\}$	max norm for $x \in \mathbb{R}^d$
$\langle A, B \rangle_F = \text{tr}(A^\top B)$	Frobenius inner product
$\|A\|_F = \sqrt{\langle A, A \rangle_F}$	Frobenius norm
$\Delta^d : \{x \in \mathbb{R}^d \| \sum_{i=1}^d x_i = 1, x_i \geq 0\}$	standard d-simplex
$\psi : \mathcal{W} \to \mathbb{R}$	closed convex function
$\partial\psi(w) = \{g \in \mathbb{X}_* \| \forall v \in \mathcal{W}. \psi(v) - \psi(w) \geq g(v - w)\}$	the set of subdifferential of ψ at w
$\psi^* : \mathbb{X}_* \to \mathbb{R}, \theta \mapsto \sup_{w \in \mathcal{W}} \theta w - \psi(w)$	convex conjugate of ψ
$\mathcal{B}_\psi(u, v) = \psi(u) - \psi(v) - g(u - v)$, where $g \in \partial\psi(u)$	the Bregman divergence

References

1. Chujai, P.; Kerdprasop, N.; Kerdprasop, K. Time series analysis of household electric consumption with ARIMA and ARMA models. Proceedings of the International MultiConference of Engineers and Computer Scientists, 2013, Vol. 1, pp. 295–300.
2. Ghofrani, M.; Arabali, A.; Etezadi-Amoli, M.; Fadali, M.S. Smart scheduling and cost-benefit analysis of grid-enabled electric vehicles for wind power integration. *IEEE Transactions on Smart grid* **2014**, *5*, 2306–2313.
3. Rounaghi, M.M.; Zadeh, F.N. Investigation of market efficiency and financial stability between S&P 500 and London stock exchange: Monthly and yearly forecasting of time series stock returns using ARMA model. *Physica A: Statistical Mechanics and its Applications* **2016**, *456*, 10–21.
4. Shumway, R.; Stoffer, D. *Time Series Analysis and Its Applications: With R Examples*; Springer Texts in Statistics, Springer New York, 2010.
5. Zhu, B.; Chevallier, J. Carbon price forecasting with a hybrid Arima and least squares support vector machines methodology. In *Pricing and Forecasting Carbon Markets*; Springer, 2017; pp. 87–107.
6. Anava, O.; Hazan, E.; Mannor, S.; Shamir, O. Online learning for time series prediction. Conference on learning theory, 2013, pp. 172–184.
7. Liu, C.; Hoi, S.C.; Zhao, P.; Sun, J. Online ARIMA algorithms for time series prediction. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 1867–1873.
8. Xie, C.; Bijral, A.; Ferres, J.L. Nonstop: A nonstationary online prediction method for time series. *IEEE Signal Processing Letters* **2018**, *25*, 1545–1549.
9. Yang, H.; Pan, Z.; Tao, Q.; Qiu, J. Online learning for vector autoregressive moving-average time series prediction. *Neurocomputing* **2018**, *315*, 9–17.
10. Joulani, P.; György, A.; Szepesvári, C. A modular analysis of adaptive (non-) convex optimization: Optimism, composite objectives, variance reduction, and variational bounds. *Theoretical Computer Science* **2020**, *808*, 108–138.
11. Zhou, Y.; Sanches Portella, V.; Schmidt, M.; Harvey, N. Regret Bounds without Lipschitz Continuity: Online Learning with Relative-Lipschitz Losses. *Advances in Neural Information Processing Systems* **2020**, *33*.
12. Jamil, W.; Bouchachia, A. Model selection in online learning for times series forecasting. UK Workshop on Computational Intelligence. Springer, 2018, pp. 83–95.
13. Jamil, W.; Kalnishkan, Y.; Bouchachia, H. Aggregation Algorithm vs. Average For Time Series Prediction. Proceedings of the ECML PKDD 2016 Workshop on Large-scale Learning from Data Streams in Evolving Environments, STREAMEVOLV-2016, 2016, pp. 1–14.
14. Box, G.E.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time series analysis: forecasting and control*; John Wiley & Sons, 2015.
15. Brockwell, P.J.; Davis, R.A. *Time Series: Theory and Methods*; Springer Science & Business Media, 2013.
16. Hamilton, J.D. *Time series analysis*; Vol. 2, Princeton New Jersey, 1994.
17. Georgiou, T.T.; Lindquist, A. A convex optimization approach to ARMA modeling. *IEEE transactions on automatic control* **2008**, *53*, 1108–1119.
18. Ding, F.; Shi, Y.; Chen, T. Performance analysis of estimation algorithms of nonstationary ARMA processes. *IEEE Transactions on Signal Processing* **2006**, *54*, 1041–1053.
19. Huang, S.J.; Shih, K.R. Short-term load forecasting via ARMA model identification including non-Gaussian process considerations. *IEEE Transactions on power systems* **2003**, *18*, 673–679.
20. Lii, K.S. Identification and estimation of non-Gaussian ARMA processes. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **1990**, *38*, 1266–1276.
21. Yang, H.; Pan, Z.; Tao, Q. Online Learning for Time Series Prediction of AR Model with Missing Data. *Neural Processing Letters* **2019**, *50*, 2247–2263.
22. Ding, J.; Noshad, M.; Tarokh, V. Order selection of autoregressive processes using bridge criterion. 2015 IEEE International Conference on Data Mining Workshop (ICDMW). IEEE, 2015, pp. 615–622.
23. Lütkepohl, H. *New introduction to multiple time series analysis*; Springer Science & Business Media, 2005.
24. Steinhardt, J.; Liang, P. Adaptivity and optimism: An improved exponentiated gradient algorithm. International Conference on Machine Learning. PMLR, 2014, pp. 1593–1601.
25. Cutkosky, A.; Boahen, K. Online learning without prior information. Conference on Learning Theory. PMLR, 2017, pp. 643–677.

 doi:10.20944/preprints202104.0601.v1

26. Cutkosky, A.; Orabona, F. Black-box reductions for parameter-free online learning in banach spaces. Conference On Learning Theory. PMLR, 2018, pp. 1493–1529.
27. Orabona, F.; Pál, D. Coin betting and parameter-free online learning. Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 577–585.
28. Orabona, F.; Pál, D. Scale-free online learning. *Theoretical Computer Science* **2018**, *716*, 50–69.
29. De Rooij, S.; Van Erven, T.; Grünwald, P.D.; Koolen, W.M. Follow the leader if you can, hedge if you must. *The Journal of Machine Learning Research* **2014**, *15*, 1281–1316.
30. Bollerslev, T. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics* **1986**, *31*, 307–327.
31. Tutun, S.; Chou, C.A.; Canıyılmaz, E. A new forecasting framework for volatile behavior in net electricity consumption: A case study in Turkey. *Energy* **2015**, *93*, 2406–2422.
32. Lu, H. "Relative Continuity" for Non-Lipschitz Nonsmooth Convex Optimization Using Stochastic (or Deterministic) Mirror Descent. *Informs Journal on Optimization* **2019**, *1*, 288–303.