

# Online Learning for Time Series Prediction

**Oren Anava**

**Elad Hazan**

**Shie Mannor**

*Technion, Haifa, Israel*

**Ohad Shamir**

*Microsoft Research and the Weizmann Institute of Science*

SOANAVA@TX.TECHNION.AC.IL

EHAZAN@IE.TECHNION.AC.IL

SHIE@EE.TECHNION.AC.IL

OHAD.SHAMIR@WEIZMANN.AC.IL

## Abstract

We address the problem of predicting a time series using the ARMA (autoregressive moving average) model, under minimal assumptions on the noise terms. Using regret minimization techniques, we develop effective online learning algorithms for the prediction problem, *without* assuming that the noise terms are Gaussian, identically distributed or even independent. Furthermore, we show that our algorithm’s performances asymptotically approaches the performance of the best ARMA model in hindsight.

**Keywords:** Time series analysis, online learning, regret minimization.

## 1. Introduction

A time series is a sequence of real-valued signals that are measured at successive time intervals. Autoregressive (AR), moving average (MA), and autoregressive moving average (ARMA) models are often used for the purpose of time-series modeling, analysis and prediction. These models have been successfully used in a wide range of applications such as speech analysis, noise cancelation, and stock market analysis (Hamilton (1994); Box et al. (1994); Shumway and Stoffer (2005); Brockwell and Davis (2009)). Roughly speaking, they are based on the assumption that each new signal is a noisy linear combination of the last few signals and independent noise terms.

A great deal of work has been done on parameter identification and signal prediction using these models, mainly in the “proper learning” setting, in which the fitted model tries to mimic the assumed underlying model. Most of this work relied on strong assumptions regarding the noise terms, such as independence and identical Gaussian distribution. These assumptions are quite strict in general and the following statement from Thomson (1994) is sometimes quoted:

*Experience with real-world data, however, soon convinces one that both stationarity and Gaussianity are fairy tales invented for the amusement of undergraduates.*

In this paper we argue that these assumptions can be relaxed into less strict assumptions on the noise terms. Moreover, we offer a novel approach for time series analysis and prediction — an *online learning approach* that allows the noise to be arbitrarily or even (to some extent) adversarially generated. The goal of this paper is to show that the new

approach is more general, and is capable of coping with a wider range of time series and loss functions (rather than only the squared loss).

### 1.1. Summary of results

We present and analyze two online algorithms for the prediction problem, one designed for general convex loss functions and the other for exp-concave ones. Each of these algorithms attains sublinear regret bound against the best ARMA prediction in hindsight, under weak assumptions on the noise terms. We apply our results to the most commonly used loss function in time series analysis, the squared loss, and achieve a regret bound of  $O(\log^2(T))$  against the best ARMA prediction in hindsight. Finally, we present an empirical study that verifies our theoretical results.

### 1.2. Related work

In standard time series analysis, the squared loss is usually considered and the noise terms are assumed to be independent with bounded variance and zero-mean. In this specific setting, one can assume without loss of generality that the noise terms have identical Gaussian distribution (see [Hamilton \(1994\)](#); [Box et al. \(1994\)](#); [Brockwell and Davis \(2009\)](#) for more information). This allows the use of statistical methods, such as least squares and maximum likelihood based methods, for the tasks of analysis and prediction. However, when different loss functions are considered these assumptions do not hold in general, and the aforementioned methods are not applicable. We are not aware of a previous approach that tries to relax these assumptions for general convex loss functions. We note that there has been previous work which tries to relax such assumptions for the squared loss, usually under additional modelling assumptions such as  $t$ -distribution of the noise (e.g., [Damsleth and El-Shaarawi \(1989\)](#); [Tiku et al. \(2000\)](#)). We emphasize that the independence assumption is rather strict and previous works that relax this assumption usually offer specific dependency model, e.g., as proposed by [Engle \(1982\)](#) for the ARCH model.

Furthermore, an online approach that relies on regret minimization techniques was never considered for ARMA prediction, and hence regret bounds of the type we are interested simply do not exist. Yet, results on the convergence rate of the coefficient vectors do exist, and regret bounds can be derived from these results. The most recent bounds we are aware of are by [Ding et al. \(2006\)](#), from which a regret bound of  $O(\log^{2+\epsilon}(T))$  for any  $\epsilon > 0$  can be derived specifically for the squared loss. We are not familiar with regret bounds for other loss functions.

## 2. Preliminaries and model

### 2.1. Time series modelling

A *time series* is a sequence of signals, measured at successive times, which are assumed to be spaced at uniform intervals. We denote by  $X_t$  the signal measured at time  $t$ , and by  $\epsilon_t$  the noise term at time  $t$ . The AR( $k$ ) (short for autoregressive) model, parameterized by a horizon  $k$  and a coefficient vector  $\alpha \in \mathbb{R}^k$ , assumes that the time series is generated

according to the following model, where  $\epsilon_t$  is a zero-mean random noise term:

$$X_t = \sum_{i=1}^k \alpha_i X_{t-i} + \epsilon_t. \quad (1)$$

In words, the model assumes that each  $X_t$  is a noisy linear combination of the previous  $k$  signals. A more sophisticated model is the ARMA( $k, q$ ) (short for autoregressive moving average) model, which is parameterized by two horizon terms  $k, q$  and coefficient vectors  $\alpha \in \mathbb{R}^k$  and  $\beta \in \mathbb{R}^q$ . This model assumes that  $X_t$  is generated via the formula:

$$X_t = \sum_{i=1}^k \alpha_i X_{t-i} + \sum_{i=1}^q \beta_i \epsilon_{t-i} + \epsilon_t, \quad (2)$$

where again  $\epsilon_t$  are zero-mean noise terms. Sometimes, an additional constant bias term is added to the equation (to indicate constant drift), but we will ignore this for simplicity. Notice that this does not increase the complexity of the problem, since we can simply raise the dimension of the vector  $\alpha$  by one and assign the value 1 to corresponding signal. Note that the AR( $k$ ) model is a special case of the ARMA( $k, q$ ) model, where the  $\beta_i$  coefficients are all zero.

## 2.2. The online setting for ARMA prediction

Online learning is usually defined in a game-theoretic framework, where the data, rather than being chosen stochastically, is chosen arbitrarily, possibly by an all-powerful adversary with full knowledge of our learning algorithm (see for instance [Cesa-Bianchi and Lugosi \(2006\)](#)). In our context, we will describe the setting as follows: first, some coefficient vectors  $(\alpha, \beta)$  are fixed by the adversary. At each time point  $t$ , the adversary chooses  $\epsilon_t$  and generates the resulting signal  $X_t$  using the formula in Equation 2. We emphasize that  $(\alpha, \beta)$  and the noise terms are not revealed to us at any time point.

At iteration  $t$ , we need to make a prediction  $\tilde{X}_t$  for the signal, after which the real signal  $X_t$  is revealed, and we suffer a *loss* denoted by  $\ell_t(X_t, \tilde{X}_t)$ . Our goal is to minimize the sum of losses over a predefined number of iterations  $T$ . A reasonable benchmark is to try to be not much worse than the best possible ARMA model. More precisely, we let

$$f_t(\alpha, \beta) = \ell_t(X_t, \tilde{X}_t(\alpha, \beta)) = \ell_t\left(X_t, \left(\sum_{i=1}^k \alpha_i X_{t-i} + \sum_{i=1}^q \beta_i \epsilon_{t-i}\right)\right) \quad (3)$$

denote the loss at time  $t$  of the (conditionally expected) prediction given by an ARMA model with some coefficients  $(\alpha, \beta)$ . We then define the *regret* as

$$R_T = \sum_{t=1}^T \ell_t(X_t, \tilde{X}_t) - \min_{\alpha, \beta} \sum_{t=1}^T \ell_t(X_t, \tilde{X}_t(\alpha, \beta)). \quad (4)$$

We wish to obtain efficient algorithms, whose regret grows sublinearly in  $T$ , corresponding to an average per-round regret going to zero as  $T$  increases. <sup>1</sup>

---

1. The iterations in which  $t \leq k$  are usually ignored since we assume that the loss per iteration is bounded by a constant, this adds at most a constant to the final regret bound.

A major challenge in our setting is that the noise terms  $\{\epsilon_t\}_{t=1}^T$  are *unknown*. As a result, we cannot use existing online convex optimization algorithms over the space of coefficient vectors  $(\alpha, \beta)$ . Moreover, even if we are given some  $(\alpha, \beta)$ , we cannot generate a prediction  $\tilde{X}_t$  using the ARMA model. This lack of information makes it also hard to compute the best coefficient vectors in hindsight, and hence competing against the best ARMA model is ill-defined in this case.

### 2.3. Our assumptions

Throughout Section 3 we assume the following:

1. The noise terms are stochastically and independently generated, each from a zero-mean distribution which might be chosen adversarially (up to the assumptions below). In Section 4 we show how to relax this assumption to adversarial noise. Also, we assume that  $\mathbb{E}[|\epsilon_t|] < M_{\max} < \infty$  and  $\mathbb{E}[\ell_t(X_t, X_t - \epsilon_t)] < \infty$  for all  $t$ .
2. The loss function  $\ell_t$  is Lipschitz continuous for some Lipschitz constant  $L > 0$ . This is a standard assumption and it holds in particular for the squared loss, as well as for other convex loss functions, with compact domain.
3. The coefficients  $\alpha_i$  satisfy  $|\alpha_i| < c$  for some  $c \in \mathbb{R}$ . This assumption is also standard, and needed in general for the decision set (defined in Subsection 3.1) to be bounded.
4. The coefficients  $\beta_i$  satisfy  $\sum_{i=1}^q |\beta_i| < 1 - \varepsilon$ , for some  $\varepsilon > 0$ .
5. The signal is bounded (by constant which is independent of  $T$ ). This assumption can be easily removed using a doubling trick, but we assume it in order to simplify the analysis. Without loss of generality we assume that  $|X_t| < X_{\max} \in \mathbb{R}$  for all  $t$ .

## 3. Online time series prediction

As said before, we cannot use existing online convex optimization algorithms over the space of coefficient vectors  $(\alpha, \beta)$  since the noise terms are unknown to us at any stage. Instead, we use an *improper learning* approach, where our predictions at each time point will not come from an ARMA model that tries to mimic the underlying model. More specifically, we fix some  $m \in \mathbb{N}$ , and at each time point  $t$ , we choose an  $(m+k)$ -dimensional coefficient vector  $\gamma \in \mathbb{R}^{m+k}$  and predict by  $\tilde{X}_t(\gamma) = \sum_{i=1}^{m+k} \gamma_i X_{t-i}$ . It follows that our loss at iteration  $t$  is determined by the loss function

$$\ell_t^m(\gamma^t) = \ell_t(X_t, \tilde{X}_t(\gamma^t)) = \ell_t\left(X_t, \left(\sum_{i=1}^{m+k} \gamma_i^t X_{t-i}\right)\right). \quad (5)$$

This can be seen as an AR model with horizon  $(m+k)$ . This leads to one of our key results: we can learn ARMA( $k, q$ ) model using AR( $m+k$ ) model, for a properly chosen value of  $m$ . We quantify this result in Theorem 1 in terms of regret.

### 3.1. Algorithm parameters definition and calculation

Before presenting the algorithm and stating our main theorem, we need to define the following parameters. The decision set  $\mathcal{K}$  is the set of candidates  $((m+k)$ -dimensional coefficient vectors) we can choose from at each iteration; it is defined as

$$\mathcal{K} = \left\{ \gamma \in \mathbb{R}^{m+k}, |\gamma_j| \leq c, j = 1, \dots, m \right\}.$$

Intuitively, the structure of  $\mathcal{K}$  follows from Assumptions 3-4 on  $\alpha$  and  $\beta$ , which restrict our improper learning variable  $\gamma$ . We denote by  $D$  the diameter of  $\mathcal{K}$ , and bound:

$$D = \sup_{\gamma_1, \gamma_2 \in \mathcal{K}} \|\gamma_1 - \gamma_2\|_2 \leq 2 \sup_{\gamma \in \mathcal{K}} \|\gamma\|_2 = 2c \cdot \sqrt{(m+k)}. \quad (6)$$

Next, we denote by  $G$  the upper-bound of  $\|\nabla \ell_t^m(\gamma)\|$  for all  $t$  and  $\gamma \in \mathcal{K}$ . This parameter depends on the loss function considered, and its computation is done accordingly. E.g., for the squared loss we get that  $G = 2c \cdot \sqrt{(m+k)} \cdot (X_{\max})^2$ , relying on Assumption 5. Finally, we denote by  $\lambda$  the exp-concavity parameter of the loss functions  $\{\ell_t^m\}_{t=1}^T$ , i.e., it holds that  $e^{-\lambda \cdot \ell_t^m(\gamma)}$  is concave for all  $t$ .<sup>2</sup> This parameter is relevant only for exp-concave loss functions, and its computation is also done according to the loss function considered. It can be shown that  $\lambda = \frac{1}{m+k}$  when the squared loss is considered.

### 3.2. ARMA Online Newton Step (ARMA-ONS)

Algorithm 1 shows how to choose  $\gamma^t$  in each iteration, when the loss functions  $\{\ell_t^m\}_{t=1}^T$  are assumed to be  $\lambda$ -exp-concave in  $\gamma$ . The notation  $\Pi_{\mathcal{K}}^{A_t}$  refers to the projection onto  $\mathcal{K}$  in the norm induced by  $A_t$ , i.e.,  $\Pi_{\mathcal{K}}^{A_t}(y) = \arg \min_{x \in \mathcal{K}} (y-x)^\top A_t (y-x)$ .

---

#### Algorithm 1 ARMA-ONS(k,q)

---

- 1: Input: ARMA order k,q; learning rate  $\eta$ ; an initial  $(m+k) \times (m+k)$  matrix  $A_0$ .
  - 2: Set  $m = q \cdot \log_{1-\varepsilon} \left( (TLM_{\max})^{-1} \right)$ .
  - 3: Choose  $\gamma^1 \in \mathcal{K}$  arbitrarily.
  - 4: **for**  $t = 1$  to  $(T-1)$  **do**
  - 5:   Predict  $\tilde{X}_t(\gamma^t) = \sum_{i=1}^{m+k} \gamma_i^t X_{t-i}$ .
  - 6:   Observe  $X_t$  and suffer loss  $\ell_t^m(\gamma^t)$ .
  - 7:   Let  $\nabla_t = \nabla \ell_t^m(\gamma^t)$ , update  $A_t \leftarrow A_{t-1} + \nabla_t \nabla_t^\top$
  - 8:   Set  $\gamma^{t+1} \leftarrow \Pi_{\mathcal{K}}^{A_t} \left( \gamma^t - \frac{1}{\eta} A_t^{-1} \nabla_t \right)$
  - 9: **end for**
- 

Note that  $A_t^{-1}$  can be efficiently re-computed after each update using the Sherman-Morrison formula.

For Algorithm 1 we can prove the following:

---

2. It is easy to show that every exp-concave function is convex, the converse does not hold.

**Theorem 1** Let  $k, q \geq 1$ , and set  $A_0 = \epsilon I_{m+k}$ ,  $\epsilon = \frac{1}{\eta^2 D^2}$ ,  $\eta = \frac{1}{2} \min\{4GD, \lambda\}$ . Then, for any data sequence  $\{X_t\}_{t=1}^T$  that satisfies the assumptions from Section 2.3, Algorithm 1 generates an online sequence  $\{\gamma^t\}_{t=1}^T$ , for which the following holds:

$$\sum_{t=1}^T \ell_t^m(\gamma^t) - \min_{\alpha, \beta} \sum_{t=1}^T \mathbb{E}[f_t(\alpha, \beta)] = O\left(\left(GD + \frac{1}{\lambda}\right) \log(T)\right). \quad (7)$$

**Remark:** The expectation is necessary since the noise terms  $\epsilon_t$  are unknown random variables, even in hindsight. Also, obtaining a high probability bound on the regret is possible but requires additional assumptions on the noise process such as boundedness or light tail.

**Proof** Intuitively, Theorem 1 states that we can have a regret as low as the best ARMA( $k, q$ ) model, using only an AR( $m+k$ ) model. The proof consists of two steps. In the first step we bound the regret suffered by an AR( $m+k$ ) prediction using familiar techniques of online convex optimization. In the second step we bound the distance between the AR( $m+k$ ) loss function and the ARMA( $k, q$ ) loss function, using a chain of bounds and inequalities. Integrating both steps yields the requested regret bound for the ARMA( $k, q$ ) loss function.

**Step 1:** Relying on the fact that the loss functions  $\{\ell_t^m\}_{t=1}^T$  are  $\lambda$ -exp-concave, we can guarantee that

$$\sum_{t=1}^T \ell_t^m(\gamma^t) - \min_{\gamma} \sum_{t=1}^T \ell_t^m(\gamma) = O\left(\left(GD + \frac{1}{\lambda}\right) \log(T)\right),$$

using the *Online Newton Step* (ONS) algorithm, presented in Hazan et al. (2007).

**Step 2:** Define recursively

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

with initial condition  $X_1^\infty(\alpha, \beta) = X_1$ . We then denote by

$$f_t^\infty(\alpha, \beta) = \ell_t(X_t, X_t^\infty(\alpha, \beta)) \quad (8)$$

the loss suffered by the prediction  $X_t^\infty(\alpha, \beta)$  at iteration  $t$ . From this definition it follows that  $X_t^\infty(\alpha, \beta)$  is of the form  $X_t^\infty(\alpha, \beta) = \sum_{i=1}^{t-1} c_i(\alpha, \beta) X_{t-i}$  for some appropriate coefficients  $c_i(\alpha, \beta)$ . The motivation behind the definition of  $f_t^\infty$  follows from the need to replace  $f_t$  with a loss function that fits the full information online optimization model (no unknown parameters). We set  $m \in \mathbb{N}$ , and define

$$X_t^m(\alpha, \beta) = \sum_{i=1}^k \alpha_i X_{t-i} + \sum_{i=1}^q \beta_i (X_{t-i} - X_{t-i}^{m-i}(\alpha, \beta)),$$

with initial condition  $X_t^m(\alpha, \beta) = X_t$  for all  $t$  and  $m \leq 0$ . We denote by

$$f_t^m(\alpha, \beta) = \ell_t(X_t, X_t^m(\alpha, \beta)) \quad (9)$$

the loss suffered by the prediction  $X_t^m(\alpha, \beta)$  at iteration  $t$ . The motivation here is simple: it is easier to generate predictions using only the last  $(m + k)$  signals, and the distance between the loss function is relatively small. Now, let

$$(\alpha^*, \beta^*) = \arg \min_{\alpha, \beta} \sum_{t=1}^T \mathbb{E}[f_t(\alpha, \beta)] \quad (10)$$

denote the best ARMA coefficient in hindsight for predicting the signal  $\{X_t\}_{t=1}^T$ . Then, from Lemma 2, stated below, we have that

$$\min_{\gamma} \sum_{t=1}^T \ell_t^m(\gamma) \leq \sum_{t=1}^T f_t^m(\alpha^*, \beta^*),$$

and it follows that

$$\sum_{t=1}^T \ell_t^m(\gamma^t) - \sum_{t=1}^T f_t^m(\alpha^*, \beta^*) = O\left(\left(GD + \frac{1}{\lambda}\right) \log(T)\right).$$

From Lemma 3 below we know that

$$\left| \sum_{t=1}^T \mathbb{E}[f_t^\infty(\alpha^*, \beta^*)] - \sum_{t=1}^T \mathbb{E}[f_t^m(\alpha^*, \beta^*)] \right| = O(1),$$

for  $m = q \cdot \log_{1-\epsilon} \left( (TLM_{\max})^{-1} \right)$ , which implies that

$$\sum_{t=1}^T \ell_t^m(\gamma^t) - \sum_{t=1}^T \mathbb{E}[f_t^\infty(\alpha^*, \beta^*)] = O\left(\left(GD + \frac{1}{\lambda}\right) \log(T)\right).$$

Finally, from Lemma 4 below we know that

$$\left| \sum_{t=1}^T \mathbb{E}[f_t^\infty(\alpha^*, \beta^*)] - \sum_{t=1}^T \mathbb{E}[f_t(\alpha^*, \beta^*)] \right| = O(1),$$

and thus

$$\sum_{t=1}^T \ell_t^m(\gamma^t) - \min_{\alpha, \beta} \sum_{t=1}^T \mathbb{E}[f_t(\alpha, \beta)] = O\left(\left(GD + \frac{1}{\lambda}\right) \log(T)\right).$$

■

Next, we state the lemmas we used. Due to space constraints, the full proofs are omitted from this version and appear in Anava et al. (2013).

Lemmas 3 and 4 below are the main technical innovation in our paper, we briefly sketch their proof. Recall that our goal is to compete against the best ARMA model in hindsight, which depends on noise terms that were generated by the adversary and are unknown to us at any stage. To bypass this issue, we define recursively a new loss function that replaces the unknown noise terms with their estimators. Then, we use Lemma 4 to prove that the

distance between the new loss function and the original one is small in expectation. This solution arises a new problem: the “order” of each prediction is now the whole signal, meaning we have to consider  $t - 1$  observations to satisfy a prediction  $\tilde{X}_t$ . Lemma 3 solves this issue by cutting the memory parameter and offering a loss function which is close in expectation to the original one. This lemma also shows that the length of the memory needed in this case is of order  $k + q \log(T)$ .

**Lemma 2** *Let  $\ell_t^m(\gamma)$ ,  $f_t^m(\alpha, \beta)$  and  $(\alpha^*, \beta^*)$  be as denoted in Equations 5, 9 and 10. Then, for all  $m \in \mathbb{N}$  and data sequence  $\{X_t\}_{t=1}^T$  that satisfies the assumptions from Section 2.3, it holds that*

$$\min_{\gamma} \sum_{t=1}^T \ell_t^m(\gamma) \leq \sum_{t=1}^T f_t^m(\alpha^*, \beta^*).$$

**Lemma 3** *Let  $f_t^\infty(\alpha, \beta)$ ,  $f_t^m(\alpha, \beta)$  and  $(\alpha^*, \beta^*)$  be as denoted in Equations 8, 9 and 10. Then, for any data sequence  $\{X_t\}_{t=1}^T$  that satisfies the assumptions from Section 2.3, it holds that*

$$\left| \sum_{t=1}^T \mathbb{E}[f_t^\infty(\alpha^*, \beta^*)] - \sum_{t=1}^T \mathbb{E}[f_t^m(\alpha^*, \beta^*)] \right| = O(1),$$

if we choose  $m = q \cdot \log_{1-\varepsilon} \left( (TLM_{\max})^{-1} \right)$ .

**Lemma 4** *Let  $f_t(\alpha, \beta)$ ,  $f_t^\infty(\alpha, \beta)$  and  $(\alpha^*, \beta^*)$  be as denoted in Equations 3, 8 and 10. Then, for any data sequence  $\{X_t\}_{t=1}^T$  that satisfies the assumptions from Section 2.3, it holds that*

$$\left| \sum_{t=1}^T \mathbb{E}[f_t^\infty(\alpha^*, \beta^*)] - \sum_{t=1}^T \mathbb{E}[f_t(\alpha^*, \beta^*)] \right| = O(1).$$

### 3.3. ARMA Online Gradient Descent (ARMA-OGD)

We now turn to present a different algorithm for choosing  $\gamma^t$  at each time point. This algorithm is applicable to general convex loss functions, as well as to exp-concave ones. It is computationally simpler but has a somewhat worse theoretical (and empirical) performance compared to the previous one, when considering an exp-concave loss function. The notation  $\Pi_{\mathcal{K}}$  refers to the Euclidean projection onto  $\mathcal{K}$ , i.e.,  $\Pi_{\mathcal{K}}(y) = \arg \min_{x \in \mathcal{K}} \|y - x\|_2$ .

For Algorithm 2 we can prove the following:

**Theorem 5** *Let  $k, q \geq 1$ , and set  $\eta = \frac{D}{G\sqrt{T}}$ . Then, for any data sequence  $\{X_t\}_{t=1}^T$  that satisfies the assumptions from Section 2.3, Algorithm 2 generates an online sequence  $\{\gamma^t\}_{t=1}^T$ , for which the following holds:*

$$\sum_{t=1}^T \ell_t^m(\gamma^t) - \min_{\alpha, \beta} \sum_{t=1}^T \mathbb{E}[f_t(\alpha, \beta)] = O\left(GD\sqrt{T}\right). \quad (11)$$

The proof of this theorem is very similar to the proof of Theorem 1, albeit plugging into our framework the Online Gradient Descent (OGD) algorithm of Zinkevich (2003) rather than the Online Newton Step algorithm.



---

**Algorithm 2** ARMA-OGD(k,q)

---

- 1: Input: ARMA order k,q. Learning rate  $\eta$ .
  - 2: Set  $m = q \cdot \log_{1-\varepsilon} \left( (TLM_{\max})^{-1} \right)$ .
  - 3: Choose  $\gamma^1 \in \mathcal{K}$  arbitrarily.
  - 4: **for**  $t = 1$  to  $(T - 1)$  **do**
  - 5:   Predict  $\tilde{X}_t(\gamma^t) = \sum_{i=1}^{m+k} \gamma_i^t X_{t-i}$ .
  - 6:   Observe  $X_t$  and suffer loss  $\ell_t^m(\gamma_t)$ .
  - 7:   Let  $\nabla_t = \nabla \ell_t^m(\gamma^t)$
  - 8:   Set  $\gamma^{t+1} \leftarrow \Pi_{\mathcal{K}} \left( \gamma^t - \frac{1}{\eta} \nabla_t \right)$
  - 9: **end for**
- 

#### 4. Additional results

The results presented in Theorems 1 and 5 rely on the assumptions that the noise terms are independent and zero-mean. Under these assumptions, the best coefficient vectors in hindsight are those that have generated the signal. However, if we allow the noise terms to be adversarially generated (the adversary chooses  $\epsilon_t$  at time  $t$  with no limitations), the best coefficient vectors in hindsight are not necessarily the ones used for generating the signal. For this case we have the following theorem:

**Theorem 6** *Denote by  $(\alpha', \beta')$  the coefficient vectors that have generated the signal, and assume that  $\{X_t\}_{t=1}^T$  satisfies Assumptions 2-5 from Section 2.3, when the noise terms are allowed to be chosen adversarially. Then, for exp-concave loss functions Algorithm 1 generates an online sequence  $\{\gamma^t\}_{t=1}^T$ , for which the following holds:*

$$\sum_{t=1}^T \ell_t^m(\gamma^t) - \sum_{t=1}^T f_t(\alpha', \beta') = O \left( \left( GD + \frac{1}{\lambda} \right) \log(T) \right),$$

*and for convex loss functions, Algorithm 2 generates an online sequence  $\{\gamma^t\}_{t=1}^T$ , for which the following holds:*

$$\sum_{t=1}^T \ell_t^m(\gamma^t) - \sum_{t=1}^T f_t(\alpha', \beta') = O \left( GD\sqrt{T} \right).$$

In this setting,  $M_{\max}$  is an upper-bound on  $|\epsilon_t|$  for all  $t = 1, \dots, T$ . Notice that we compare here the total loss suffered by our algorithms to the expected loss suffered by ARMA prediction with the coefficient vectors that have generated the signal, and not to the expected loss of the best ARMA prediction in hindsight. Nevertheless, this theorem captures interesting cases (e.g., correlated noise), in which traditional approaches fail to perform properly. The proof of this theorem resembles the proof of Theorem 1, with the modification of plugging  $(\alpha', \beta')$  into Lemmas 3 and 4, instead of  $(\alpha^*, \beta^*)$ .

## 5. Experiments

### 5.1. Application of Theorem 1 to squared loss

As already mentioned, the squared loss is the most commonly used loss function in time series analysis. It is defined as  $\ell_t(X_t, \tilde{X}_t) = (X_t - \tilde{X}_t)^2$  for prediction  $\tilde{X}_t$  and signal  $X_t$ . In our case, the predictions come from an AR model with horizon  $(m + k)$ , and hence our loss at time  $t$  is  $(X_t - \sum_{i=1}^{m+k} \gamma_i^t X_{t-i})^2$ , when  $\{\gamma^t\}_{t=1}^T$  are generated using Algorithm 1. Substituting the values of  $G$ ,  $D$  and  $\lambda$ , as defined and computed in Subsection 3.1 for the squared loss, yields the following result:

$$\sum_{t=1}^T \ell_t^m(\gamma^t) - \min_{\alpha, \beta} \sum_{t=1}^T \mathbb{E}[f_t(\alpha, \beta)] = O(k \log(T) + q \log^2(T)). \quad (12)$$

This result implies that the average loss suffered by Algorithm 1 converges asymptotically to the average loss suffered by the best ARMA prediction in hindsight, under the assumptions of Section 2.3.

### 5.2. Experiments with artificial data

The following experiments demonstrate the prediction effectiveness of the proposed algorithms, under some different settings. We compare the performance to the ARMA-RLS algorithm, which was presented in Ding et al. (2006). In a few words, the ARMA-RLS is a “proper learning” algorithm — it tries to mimic the underlying model. It estimates the noise terms using a recursive least squares based method, and satisfies a prediction using these estimations and the previous signals. The ARMA-RLS does not assume noise stationarity or ergodicity. We also benchmark the standard Yule-Walker estimation method.<sup>3</sup> The results are displayed in the figures below. In all cases, the  $x$ -axis is time (number of samples), and the  $y$ -axis is the average squared loss. Also, we have averaged the results over 20 runs for stability, and chose the order of our AR prediction to be  $m + k = 10$  in all settings.

**Setting 1.** We started with a simple sanity check using Gaussian noise. We generated a stationary ARMA process using the coefficient vectors  $\alpha = [0.6, -0.5, 0.4, -0.4, 0.3]$  and  $\beta = [0.3, -0.2]$ , when the noise terms are uncorrelated and normally distributed as  $\mathcal{N}(0, 0.3^2)$ . Note that since predicting the noise is impossible, a perfect predictor will suffer an average error rate of at least the variance of the noise — 0.09 in this setting. As can be seen in Figure 1(a) the ARMA-ONS algorithm outperforms the other online algorithms due to its lower regret in this setting of exp-concave loss functions, and quickly approaches the performance of the perfect predictor.

**Setting 2.** We generated the non-stationary ARMA process using the coefficient vectors  $\beta = [0.32, -0.2]$  and

$$\alpha(t) = [-0.4, -0.5, 0.4, 0.4, 0.1] * \left(\frac{t}{10^4}\right) + [0.6, -0.4, 0.4, -0.5, 0.4] * \left(1 - \frac{t}{10^4}\right),$$

---

3. Yule-Walker estimation method is offline. We use it as an online prediction method by a simple adaptation — we let it predict the signal at time  $t$  with the knowledge of the signal at times  $1, \dots, t - 1$ .

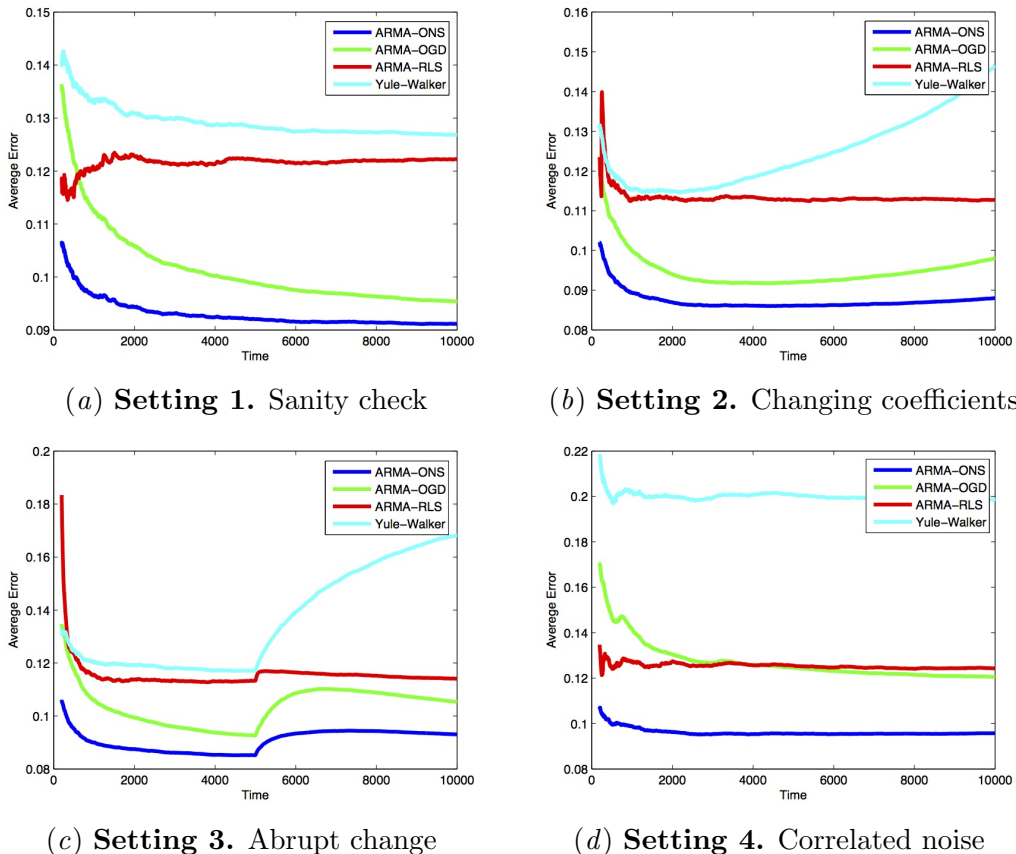


Figure 1: Experimental results for artificial data, all averaged over 20 runs.

i.e., the coefficient vectors change slowly in time. The noise terms are uncorrelated and distributed uniformly on  $[-0.5, 0.5]$  (denoted as  $Uni[-0.5, 0.5]$ ). In this setting, a perfect predictor will suffer average error rate of at least 0.0833, due to the variance of the noise. The motivation behind this setting is to demonstrate the effectiveness of the online algorithms in the non-stationary case, in which the coefficients change in time. This is especially important when dealing with real data time series, since the stationarity assumption is rather strict. In Figure 1(b) we can see the clear advantage of our online algorithms. Here again, ARMA-ONS is superior to the other algorithms, despite it being less adaptive — as the theoretical bounds predict; see Hazan and Seshadhri (2009) for discussion of adaptivity of OGD vs. ONS.

**Setting 3.** Here we consider the non-stationary ARMA process that is generated using two different sets of coefficient vectors. The first set is  $\alpha = [0.6, -0.5, 0.4, -0.4, 0.3]$  and  $\beta = [0.3, -0.2]$ , and it is used for generating the signal at the first half of the iterations. The second set is  $\alpha = [-0.4, -0.5, 0.4, 0.4, 0.1]$  and  $\beta = [-0.3, 0.2]$ , and it is used for generating the signal at the second half of the iterations. The noise terms are uncorrelated and distributed  $Uni[-0.5, 0.5]$ . In Figure 1(c) we demonstrate the effectiveness of online algo-

rithms in a scenario when the coefficients abruptly change. Here again, a perfect predictor will suffer average error rate of at least 0.0833, due to the variance of the noise.

**Setting 4.** Consider an ARMA process that is generated using the coefficient vectors  $\alpha = [0.11, -0.5]$  and  $\beta = [0.41, -0.39, -0.685, 0.1]$ . Each noise term is distributed normally, with expectation that is the value of the previous noise term, and variance  $0.3^2$ . I.e., the noise terms are positively correlated. In Figure 1(d) one can clearly see the robustness of online algorithms to correlated noise. Note that despite the correlativity introduced in this setting, ARMA-ONS achieves an average error rate that converges approximately to the variance of the noise — 0.09.

## 6. Conclusion and discussion

In this paper we developed a new approach for time series analysis — an online learning approach. Our main result in this paper is that one can predict time series as well as the best ARMA model, regardless of the loss function considered, under weak assumptions on the noise terms — zero mean distribution. This result is strengthened in light of the fact that the noise terms in the underlying model are unknown to us at any stage. We overcome this difficulty by using improper learning techniques. Additionally, we present an analytical extension of our approach to adversarially generated noise terms. The main powerful properties of the online approach, as pointed out in our work, are generality, simplicity and efficiency, in comparison to existing methods.

There are three issues that remain for further research. First, in our analysis we assume that  $\sum_{i=1}^q |\beta_i| < 1 - \varepsilon$  for some  $\varepsilon > 0$ , which seems to limit the freedom of the  $\beta$  coefficients. This assumption appears sometimes in the literature (e.g., in GARCH models) and is a sufficient condition for the MA component to be causally invertible, yet not necessary. In our case, we believe that this assumption follows from our proof techniques and the results would still hold for any  $\beta$  coefficients. Second, in Section 4 we present results in which the total loss suffered by our algorithms is compared to the expected loss suffered by ARMA prediction with the coefficient vectors that have generated the signal. Whereas competing against the best ARMA prediction under adversarial noise is impossible because of identifiability issues, it would be interesting to study intermediate setups such as correlated or adversarial noise to some extent. Third, the ARMA model is not appropriate for every time series, as can be quite expected. However, Engle (1982) showed that some finance related time series can be well predicted using the ARCH model and its expansions. Therefore, it would be interesting to generalize our work to other time series models, such as ARCH and ARIMA.

## Acknowledgments

This research was funded in part by the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI). The first two authors were supported in part by ISF grant 810/11 and the third author was supported in part by ISF grant 920/12.

**References**

- Oren Anava, Elad Hazan, Shie Mannor, and Ohad Shamir. Online learning for time series prediction. *CoRR*, abs/1302.6927, 2013.
- G. Box, G. Jenkins, and G. Reinsel. *Time Series Analysis: Forecasting and Control*. Prentice-Hall, 3 edition, 1994.
- P. Brockwell and R. Davis. *Time Series: Theory and Methods*. Springer, 2 edition, 2009.
- N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.
- E. Damsleth and A. H. El-Shaarawi. ARMA models with double-exponentially distributed noise. *Journal of the Royal Statistical Society. Series B*, 51(1):61–69, 1989.
- F. Ding, Y. Shi, and T. Chen. Performance analysis of estimation algorithms of nonstationary ARMA processes. *IEEE Transactions on Signal Processing*, 33(3):1041–1053, 2006.
- R.F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50:987–1007, 1982.
- J. Hamilton. *Time Series Analysis*. Princeton Univ. Press, 1994.
- E. Hazan and C. Seshadhri. Efficient learning algorithms for changing environments. In *ICML*, page 50, 2009.
- E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.
- R. Shumway and D. Stoffer. *Time Series Analysis and Its Applications*. Springer, 2005.
- D. Thomson. Jackknifing multiple-window spectra. In *ICASSP*, volume 6, pages VI/73 – VI/76, April 1994.
- M. Tiku, W. K. Wong, D. Vaughan, and G. Bian. Time series models in non-normal situations: Symmetric innovations. *Journal of Time Series Analysis*, 21(5):571–596, 2000.
- M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.