Online Time Series Prediction with Missing Data

Oren Anava OANAVA@TX.TECHNION.AC.IL
Technion, Haifa, Israel

Elad Hazan EHAZAN@CS.PRINCETON.EDU
Princeton University, NY, USA

Assaf Zeevi ASSAF@GSB.COLUMBIA.EDU
Columbia University, NY, USA

Abstract

We consider the problem of time series prediction in the presence of missing data. We cast the problem as an online learning problem in which the goal of the learner is to minimize prediction error. We then devise an efficient algorithm for the problem, which is based on autoregressive model, and *does not* assume any structure on the missing data nor on the mechanism that generates the time series. We show that our algorithm's performance asymptotically approaches the performance of the best AR predictor in hindsight, and corroborate the theoretic results with an empirical study on synthetic data.

1. Introduction

A time series is a sequence of signal observations, typically measured at uniform time intervals. Perhaps one of the most well-studied models for time series analysis and prediction is the autoregressive (AR) model. Roughly speaking, the AR model is based on the assumption that each observation can be represented as a (noisy) linear combination of some previous observations. This model has been successfully used in many real-world applications such as DNA microarray data analysis, stock market prediction, and noise cancelation, to name but a few.

Recently there has been growing interest in the problem of time series prediction in the presence of missing data, mainly in the "proper learning" setting, in which an underlying model is assumed and the goal is to recover its parameters. Most of the current work relies on statistical assumptions on the error terms, such as independence and Gaussian distribution. These assumptions allow the use of Maximum Likelihood (ML) techniques to recover consistent (and sometimes optimal) estimators for the model parameters. However, these assumptions are many times not met in practice, causing the resulting estimators to be no longer consistent. Occasionally, additional assumptions on the structure of the missing data are added, and the statistical modeling becomes even more distant from the data.

In this paper we argue that assumptions on the model generating the time series and the structure of its missing data can be relaxed in a substantial manner while still supporting the development of efficient methods to solve the problem. Our main contribution is a novel *online learning approach* for time series prediction with missing data, that allows the observations (along with the missing data) to be arbitrarily or even adversarially generated. The goal of this paper is to show that the new approach is theoretically more robust, and is thus capable of coping with a wider range of time series and missing data structures.

1.1. Informal Statement of Our Results

We cast the problem of AR prediction in the presence of missing data as an online learning problem. A major modeling challenge arises as AR prediction is not well defined when some of the data is missing. To overcome this issue, we define a new family of AR predictors; each such predictor makes use of its own past predictions to "fill in" the missing data, and then provides an AR prediction using the completed data. To be slightly more formal, a predictor in this family has the following recursive form:

$$\tilde{X}_t^{\mathrm{REC}}(\alpha) = \sum_{k=1}^{p} \alpha_k X_{t-k} \mathbf{1}\{X_{t-k} \text{ is revealed}\} + \sum_{k=1}^{p} \alpha_k \tilde{X}_{t-k}^{\mathrm{REC}}(\alpha) \mathbf{1}\{X_{t-k} \text{ is not revealed}\},$$

Proceedings of the 31^{st} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

where X_t is the signal measured at time point t, and $\alpha \in \mathbb{R}^p$ is the vector of AR coefficients. Now, let $\ell_t(X_t, \tilde{X}_t)$ denote the loss suffered by predicting $\tilde{X}_t$ at time point t, and $\mathcal{R}_T$ be the corresponding regret term. Then, our main theorem is the following:

Theorem 3.1. *Our algorithm generates an online sequence of predictions $\{\tilde{X}_t\}_{t=1}^T$, for which it holds that:*

$$\mathcal{R}_T = \sum_{t=1}^{T} \ell_t\big(X_t, \tilde{X}_t\big)\mathbf{1}\{X_t \text{ is revealed}\} - \sum_{t=1}^{T} \ell_t\big(X_t, \tilde{X}_t^{\text{REC}}(\alpha^*)\big)\mathbf{1}\{X_t \text{ is revealed}\} = \mathcal{O}(\sqrt{F}),$$

where F denotes the number of time points in which our algorithm received feedback, and α^* is the minimizer of $\sum_{t=1}^{T} \ell_t(X_t, \tilde{X}_t^{REC}(\alpha))\mathbf{1}\{X_t \text{ is revealed}\}$.

This result is somewhat surprising, as even the problem of finding the best AR predictor in hindsight is non-convex due to the recursive structure of the predictors we consider. The key observation that enables an efficient solution in this scheme relies on non-proper learning: it is possible to learn coefficients in a much larger class and compete against the best AR predictor. The complexity of this class is in fact exponential in the parameters of our problem, yet we prove that the learning of the new coefficients can be done efficiently due to some special characteristics. This idea was successfully applied also in the work of (Hazan et al., 2015), who considered the problem of low-rank classification with missing data.

1.2. Related Work

Several different approaches for time series prediction in the presence of missing data exist; we overview the major ones. Perhaps the earliest approach, originated in the control theory community, goes back to the work of (Kalman, 1960). In this work, the concept of state-space modeling is presented to deal with streams of noisy input data. Although the original work is not aimed at coping with missing data, it initialized a solid line of works that use state-space modeling to impute missing data (Shumway & Stoffer, 1982; Sinopoli et al., 2004; Liu & Goldsmith, 2004; Wang et al., 2005). We refer the reader to (Durbin & Koopman, 2012) for a complete overview of time series analysis using state-space models.

Another increasingly common approach builds upon the concept of multiple imputation (Honaker & King, 2010). Roughly speaking, multiple imputation aims at inferring relevant information from the observed data (using known statistical methods), and use it to impute multiple values for each missing data point. The resulted multiple data sets are now treated as completed, which allows the use of standard statistical methods for the analysis task. Results from different data sets are then combined using various simple procedures.

In the statistical literature, missing data are usually imputed using maximum likelihood estimators corresponding to a specific underlying model. Very often, these estimators are not efficiently computed, which motivates the use of Expectation Maximization (EM) algorithms. This approach was proposed by (Dempster et al., 1977), and is currently the most popular for dealing with missing data in time series. Essentially, the EM algorithm avoids the separate treatment of each of the exponentially many missing data patterns by using the following two-step procedure: in the E-step, missing observations are filled in with their conditional expectations given the observed data and the current estimate of the model parameters; and in the M-step, a new estimate of the model parameters is computed from the current version of the completed data. We note that the vast majority of the state-space modeling literature relies on EM techniques as well (for instance, see (Shumway & Stoffer, 1982; Sinopoli et al., 2004)).

One particular time series model that has received a great attention in the statistical literature is the AR model. In the context of missing data, there are many works that assume an underlying AR model on the data, and differ in the assumptions on the missing data patterns: in (Dunsmuir & Robinson, 1981), a stochastic mechanism is assumed to generate the missing data; (Ding et al., 2010) consider a scarce pattern of the observed signal; and (Choong et al., 2009) rely on local similarity structures in the data. The existence of distributional assumptions on the time series or on the patterns of the missing data is in common to all of these works.

To date, we are not aware of an approach that allows the signal (along with its missing data) to be generated arbitrary, let alone adversarially. The only approach that allows for adversarially generated signals we are aware of is the recent result of (Anava et al., 2013), which does not account for missing data. Our work can be seen as an extension of the latter to the missing data setting.

2. Preliminaries and Model

A *time series* is a sequence of signal observations, measured at successive time points (usually spaced at uniform intervals). Let X_t denote the signal measured at time t. The traditional AR model, parameterized by lag p and coefficient vector $\alpha \in \mathbb{R}^p$, assumes that each observation complies with the formula

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

where $\{\epsilon_t\}_{t\in\mathbb{Z}}$ is assumed to be white noise. In words, the model assumes that X_t is a noisy linear combination of the previous p observations. Sometimes, an additional additive term α_0 is included to indicate 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 α by one and assign the value 1 to the corresponding observation.

The motivation to use $\mathrm{AR}(p)$ models for signal prediction goes back to Wold's decomposition theorem. According to this theorem, a stationary signal $\{X_t\}_{t\in\mathbb{Z}}$ can be represented as an $\mathrm{MA}(\infty)$ process. That is,

$$X_t = \sum_{i=1}^{\infty} \beta_i \epsilon_{t-i} + \epsilon_t,$$

where $\sum_{i=1}^{\infty} \beta_i^2 < \infty$, and $\{\epsilon_t\}_{t\in\mathbb{Z}}$ have zero-mean and equal variance. If, in addition, $\{X_t\}_{t=1}^{T}$ is assumed to be invertible, we can represent it as an $\mathrm{AR}(\infty)$ process. That is,

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

where $\{\alpha_i\}_{i=1}^{\infty}$ are uniquely defined. This representation, accompanied with the natural assumption that α_i decays fast enough as a function of i, motivates the use of $\mathrm{AR}(p)$ models for the task of signal prediction.

2.1. The Online Setting for AR Prediction

After motivating the use of AR models, arises the question of misspecification: what happens if we employ a model that does not comply with our data? Standard statistical methods (e.g., maximum likelihood) for estimating the AR coefficients are based on the assumption that the observations come from an AR model, and thus are not suitable when the model is misspecified. This drives the use of online learning based techniques to circumvent this issue.

Online learning is a well established learning paradigm which has both theoretical and practical appeals. The goal in this paradigm is to make a sequential prediction, where the data, rather than being generated stochastically, is assumed to be chosen by an adversary that has full knowledge of our learning algorithm (see for instance (Cesa-Bianchi & Lugosi, 2006)). Specifically, the following setting is usually assumed in the context of time series prediction: at time point t, we need to make a prediction $\tilde{X}_t$, after which the true value of the signal X_t is revealed, and we suffer a *loss* denoted by $\ell_t(X_t, \tilde{X}_t)$. Usually, ℓ_t is assumed to be convex with Lipschitz gradients.

When considering an $\mathrm{AR}(p)$ prediction, we must define in advance the decision set $\mathcal{K} \subset \mathbb{R}^p$, which stands for the class of AR coefficients against which we want to compete. We henceforth let $\mathcal{K} = [-1, 1]^p$. Our prediction at time point t then takes the form:

$$\tilde{X}_t^{\mathrm{AR}}(\alpha^t) = \sum_{k=1}^{p} \alpha_k^t X_{t-k}, \tag{1}$$

where $\alpha^t \in \mathcal{K}$ is generated by our online algorithm. Here comes the punch of the online setting: our goal is to design an algorithm that generates a sequence $\{\alpha^t\}_{t=1}^{T}$ which is almost as good as the best (in hindsight) AR coefficients in $\mathcal{K}$. More formally, we define the *regret* to be

$$\mathcal{R}_T = \sum_{t=1}^{T} \ell_t\big(X_t, \tilde{X}_t^{\mathrm{AR}}(\alpha^t)\big) - \min_{\alpha\in\mathcal{K}} \sum_{t=1}^{T} \ell_t\big(X_t, \tilde{X}_t^{\mathrm{AR}}(\alpha)\big),$$

and wish to design efficient algorithms, whose regret grows sublinearly in T. Thus, even if the model is misspecified (meaning the best AR coefficients in $\mathcal{K}$ have unsatisfactory predictive power), minimizing the regret term is still meaningful. Remains the question of *how can we compete against the best* AR *coefficients in the presence of missing data?* The latter is the main question we try to answer in this work.

2.2. Problem Definition

Throughout this work we consider the following setting (which accounts for missing data): at time point t, we need to make a prediction $\tilde{X}_t$, after which feedback in the form of the real signal X_t is not necessarily revealed, and we suffer *loss* denoted by $\ell_t(X_t, \tilde{X}_t)\mathbf{1}\{X_t \text{ is revealed}\}$. That is, we suffer loss only if we receive feedback. Here also, ℓ_t is assumed to be convex with Lipschitz gradients.

The problem arising in this setting is two-fold: first, we cannot provide an AR prediction at time t (even given the vector α) if some of the required past observations are missing. Second, the best AR predictor in hindsight is not well-defined. To solve this problem, we define a family of recursive predictors, each of the form:

$$\begin{aligned}\tilde{X}_t^{\mathrm{REC}}(\alpha) = &\sum_{k=1}^{p} \alpha_k X_{t-k} \mathbf{1}\{X_{t-k} \text{ is revealed}\} \\ &+ \sum_{k=1}^{p} \alpha_k \tilde{X}_{t-k}^{\mathrm{REC}}(\alpha) \mathbf{1}\{X_{t-k} \text{ is not revealed}\},\end{aligned} \tag{2}$$

where $\alpha \in \mathcal{K} = [-1, 1]^p$. Essentially, a predictor in this family uses its own (updated) estimations as a proxy for the actual signal, and then provides an $\mathrm{AR}(p)$ prediction as if there is no missing data. The problem at hand then translates into minimizing the corresponding regret term.

2.3. Our Assumptions

Throughout the remainder of this work we assume that the following hold:

(1) $X_t \in [-1,1]$ for all t. Here, the constant 1 can be replaced by any other constant $C < \infty$ (which is known in advanced), but in order to simplify the writing we assume that $C = 1$.

(2) For all t there exist at least p successive time points in $t-d,\ldots,t-1$ for which we received feedback. This assumption makes sure that each prediction $\tilde{X}_t^{\text{REC}}(\alpha)$ does not "look back" more than d time points. This assumption can be completely removed, as discussed in Section 3.4.

Note we do not assume an underlying $\text{AR}(p)$ model that generates the signal, nor a statistical model according to which the missing observations are omitted from us.

3. Our Approach

We briefly outline our approach to the problem at hand. Basically, observe that the prediction at time point t can be written in the following form:

$$\tilde{X}_t^{\text{REC}}(\alpha) = \sum_{k=1}^{d} p_k(\alpha) X_{t-k} \mathbf{1}\{X_{t-k}\},$$

where $p_k(\alpha)$ is a polynomial in α that is determined by the structure of missing data. Each polynomial p_k potentially contains up to 2^{k-1} terms of the form $\alpha_{i_1} \cdot \ldots \cdot \alpha_{i_j}$, such that $\sum_{m=1}^{j} i_m = k$. This means that the prediction at time point t constitutes of up to 2^d terms of the form $\alpha_{i_1} \cdot \ldots \cdot \alpha_{i_j}$, such that for each of them it holds that $\sum_{m=1}^{j} i_m \leq d$. Notice that each such term is less or equal to 1 for α^* that is the best recursive $\text{AR}(p)$ predictor in hindsight, since we consider $\mathcal{K} = [-1,1]^p$.

The latter observation allows the use of non-proper learning techniques in this setting. Essentially, the idea is to learn a vector $w \in \mathbb{R}^{2^d}$ such that each entry in w corresponds to a product of the form $\alpha_{i_1} \cdot \ldots \cdot \alpha_{i_j}$, while ignoring the restrictions imposed on w by α. Obtaining a regret bound w.r.t. to best w would imply a regret bound w.r.t. the best α, yet an efficiency question would remain since the dimension of w is 2^d. In Section 3.3 we prove that the inner products in the space induced by this relaxation can be computed efficiently, which overall gives an efficient algorithm with provable regret bound.

3.1. Notation and Definitions

We denote by $[n]$ the set $\{1,\ldots,n\}$, and use $X_{(t-d:t-1)} \in \{\mathbb{R} \cup \{*\}\}^d$ to denote the vector of values $X_{t-d},\ldots,X_{t-1}$, where missing observations are encoded using $\{*\}$. The notation $\mathbf{1}\{X_t\}$ will be used as the indicator of the event $\{X_t \text{ is revealed}\}$. Finally, denote

$$\tilde{\mathbb{B}}_d = \left\{ w \in \mathbb{R}^{2^d-1} \ : \ \|w\|_2^2 \leq 2^d \right\}.$$

We say that the vector $b = (b_1,\ldots,b_d) \in \{0,1\}^d$ is the *binary representation* of a number n if n is represented as $\sum_{k=1}^{d} b_k 2^{k-1}$. We denote by $b(n)$ the unique binary representation of n. For a given number n and its binary representation $b(n) = (b_1,\ldots,b_d)$, we define $m(n)$ to be the maximal index k such that $b_k = 1$. That is,

$$m(n) = \max\left\{k \ : \ 2^{k-1} < n\right\}.$$

The following definition links between a structure of missing observations and a binary vector b. This, in turn, will be used to link between a vector $w \in \mathbb{R}^{2^d-1}$ and a structure of missing observations.

Definition 1. Let $b = (b_1,\ldots,b_d) \in \{0,1\}^d$ and set $m = \max\{k \mid b_k = 1\}$. We say that b is a *semi-valid path* with respect to $X_{(t-d:t-1)}$ (and denote $b \overset{sv}{\sim} X_{(t-d:t-1)}$), if $b_m = \mathbf{1}\{X_{t-m}\} = 1$ and $b_i \geq \mathbf{1}\{X_{t-i}\}$ for $i < m$. If in addition $(b_1,\ldots,b_m)$ does not contain p successive zeros we say that b is a *valid path* with respect to $X_{(t-d:t-1)}$ (and denote $b \overset{v}{\sim} X_{(t-d:t-1)}$).

Basically, each structure of missing data corresponds to many valid paths; each of these corresponds to coefficient of a revealed signal in Equation (2). To see that, note that

$$\tilde{X}_t^{\text{REC}}(\alpha) = \sum_{k=1}^{d} p_k(\alpha | X_{(t-d:t-1)}) X_{t-k} \mathbf{1}\{X_{t-k}\},$$

where $p_k(\alpha | X_{(t-d:t-1)})$ is a polynomial in α that is determined by the structure of missing data in $X_{(t-d:t-1)}$.

Definition 2. For a given vector $X_{(t-d:t-1)}$, we define the function $\Phi(X_{(t-d:t-1)}) \in \mathbb{R}^{2^d-1}$ as follows:

$$\left[\Phi(X_{(t-d:t-1)})\right]_n = \begin{cases} X_{t-m(n)} & \text{if } b(n) \overset{sv}{\sim} X_{(t-d:t-1)}, \\ 0 & \text{otherwise.} \end{cases}$$

Here, $\left[\Phi(X_{(t-d:t-1)})\right]_n$ denotes the n-th coordinate of the vector $\Phi(X_{(t-d:t-1)})$. Similarly, we define an auxiliary function Φ^p for valid paths:

$$\left[\Phi^p(X_{(t-d:t-1)})\right]_n = \begin{cases} X_{t-m(n)} & \text{if } b(n) \overset{v}{\sim} X_{(t-d:t-1)}, \\ 0 & \text{otherwise.} \end{cases}$$

From now on, we use the notation $\tilde{X}_t(w)$ for predictions of the form $w^\top \Phi(X_{(t-d:t-1)})$, and $\tilde{X}_t^p(w)$ for predictions of the form $w^\top \Phi^p(X_{(t-d:t-1)})$.

Algorithm 1 LAZY OGD (on ℓ_2-ball with radius D)

Input: learning rate η_t.
Set $a_1 = 0$.
for $t = 1$ to T **do**
  Play a_t and incur loss $f_t(a_t)$
  Set $a_{t+1} = -\frac{\eta_t \sum_{i=1}^{t} \nabla f_i(a_i)}{\max\{1, \frac{\eta_t}{D} \| \sum_{i=1}^{t} \nabla f_i(a_i) \|\}}$
end for

3.2. Algorithm and Analysis

We take a step back to present an online algorithm: LAZY OGD (Algorithm 1), which is aimed at minimizing regret in the general online learning framework. LAZY OGD is a special instance of the FTRL algorithm when $\mathcal{K}$ is an ℓ_2-ball with radius D. Let $\{f_t\}_{t=1}^T$ be convex loss functions for which $\max_{a,t}\{\|\nabla f_t(a)\|\} \leq G$ and denote $a^* = \arg\min_{\|a\| \leq D} \sum_{t=1}^T f_t(a)$. Then, the regret of LAZY OGD is bounded as follows:

$$\begin{aligned}\mathcal{R}_T^{\text{Lazy}} &= \sum_{t=1}^T f_t(a_t) - \min_{\|a\| \leq D} \sum_{t=1}^T f_t(a) \\ &\leq \frac{\|a^*\|^2}{\eta_T} + \sum_{t=1}^T \eta_t \|\nabla f_t(a_t)\|^2 \\ &\leq 3D \sqrt{\sum_{t=1}^T \|\nabla f_t(a_t)\|^2} \leq 3GD\sqrt{T}.\end{aligned}$$

A complete analysis can be found in (Hazan, 2011; Shalev-Shwartz, 2012).

Our algorithm (Algorithm 2) is an adaptation of LAZY OGD to $\tilde{\mathbb{B}}_d$, but notice that in its current form it is inefficient (since the dimension of $\tilde{\mathbb{B}}_d$ is exponential in d). This form is easier to analyze and is thus stated here; an efficient version of Algorithm 2 is presented in Section 3.3. In the sequel, we denote $D = \max_{w \in \tilde{\mathbb{B}}_d}\{\|w\|\}$ and $G = \max_{w,t}\{\|\nabla \ell_t(X_t, \tilde{X}_t(w))\|\}$. From the definition of $\tilde{\mathbb{B}}_d$ it follows directly that $D = 2^{d/2}$. The value of G depends on the loss functions considered; for instance, if we consider the squared loss then $G = 2^{d/2}$.

The following is our main theorem:

Theorem 3.1. *Algorithm 2 generates an online sequence $\{w_t\}_{t=1}^T$ for which it holds that:*

$$\begin{aligned}&\mathcal{R}_T = \sum_{t=1}^T \ell_t(X_t, \tilde{X}_t(w^t))\mathbf{1}\{X_t\} \\ &- \min_{\alpha \in \mathcal{K}} \sum_{t=1}^T \ell_t(X_t, \tilde{X}_t^{REC}(\alpha))\mathbf{1}\{X_t\} \leq 3GD\sqrt{\sum_{t=1}^T \mathbf{1}\{X_t\}},\end{aligned}$$

if we choose $\eta_t = \frac{D}{G\sqrt{\sum_{\tau=1}^t \mathbf{1}\{X_\tau\}}}$.

Algorithm 2

Input: learning rate η_t.
Set $w_1 = 0$.
for $t = 1$ to T **do**
  Play w^t and incur $f_t(w^t) = \ell_t(X_t, \tilde{X}_t(w^t))\mathbf{1}\{X_t\}$
  Set $w^{t+1} = -\frac{\eta_t \sum_{\tau=1}^t \nabla f_\tau(w^\tau)\mathbf{1}\{X_\tau\}}{\max\{1, \eta_t \| \sum_{\tau=1}^t \nabla f_\tau(w^\tau)\| \cdot 2^{-d/2}\}}$
end for

Proof. Algorithm 2 is simply LAZY OGD on the decision set $\tilde{\mathbb{B}}_d$. Thus, by applying it we get that

$$\begin{aligned}&\sum_{t=1}^T \ell_t\big(X_t, \tilde{X}_t(w^t)\big)\mathbf{1}\{X_t\} \\ &- \min_{w \in \tilde{\mathbb{B}}_d} \sum_{t=1}^T \ell_t\big(X_t, \tilde{X}_t(w)\big)\mathbf{1}\{X_t\} \leq 3GD\sqrt{\sum_{t=1}^T \mathbf{1}\{X_t\}}.\end{aligned} \tag{3}$$

Now, we can write

$$\begin{aligned}&\min_{w \in \tilde{\mathbb{B}}_d} \sum_{t=1}^T \ell_t\big(X_t, \tilde{X}_t(w)\big)\mathbf{1}\{X_t\} \\ &\overset{(a)}{\leq} \min_{w \in \tilde{\mathbb{B}}_d} \sum_{t=1}^T \ell_t\big(X_t, \tilde{X}_t^p(w)\big)\mathbf{1}\{X_t\} \\ &\overset{(b)}{\leq} \min_{\|\alpha\|_\infty \leq 1} \sum_{t=1}^T \ell_t\big(X_t, \tilde{X}_t^{\text{REC}}(\alpha)\big)\mathbf{1}\{X_t\},\end{aligned} \tag{4}$$

where $\tilde{X}_t^p(w)$ is of the form $w^\top \Phi^p(X_{(t-d:t-1)})$.

The two key inequalities above are explained as follows. To prove (a), note that from the construction of Φ and Φ^p it follows that for any t, $\Phi^p(X_{(t-d:t-1)})$ can be written as

$$\big[\Phi^p(X_{(t-d:t-1)})\big]_n = \begin{cases} \big[\Phi(X_{(t-d:t-1)})\big]_n & \text{if } n \in \mathcal{N}, \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathcal{N}$ is the set of all numbers $n \in \big[2^d - 1\big]$ for which there exists $X \in \{\mathbb{R} \cup \{*\}\}^d$ such that $b(n) \overset{v}{\sim} X$. In particular, if we denote

$$w^* = \arg\min_{w \in \tilde{\mathbb{B}}_d} \sum_{t=1}^T \ell_t\big(X_t, \tilde{X}_t^p(w)\big)\mathbf{1}\{X_t\}$$

then w.l.o.g. we can assume that $w_n^* = 0$ for all $n \notin \mathcal{N}$. Now, note that for w^* it holds that $\tilde{X}_t^p(w^*) = \tilde{X}_t(w^*)$ for all t, which implies that

$$\sum_{t=1}^T \ell_t\big(X_t, \tilde{X}_t(w^*)\big)\mathbf{1}\{X_t\} = \sum_{t=1}^T \ell_t\big(X_t, \tilde{X}_t^p(w^*)\big)\mathbf{1}\{X_t\}.$$

Finally, since $\min_{w \in \mathbb{B}_d} \sum_{t=1}^T \ell_t\big(X_t, \tilde{X}_t(w)\big)\mathbf{1}\{X_t\} \leq \sum_{t=1}^T \ell_t\big(X_t, \tilde{X}_t(w^*)\big)\mathbf{1}\{X_t\}$, the claim holds.

Now, to prove (b), let us first denote

$$\alpha^* = \arg\min_{\|\alpha\|_\infty \le 1} \sum_{t=1}^{T} \ell_t\big(X_t, \tilde{X}_t(\alpha)\big)\mathbf{1}\{X_t\}.$$

Next, for a given vector $b \in \{0,1\}^d$ we define $\mathcal{I}(b) \in [d]^d$ as follows:

$$[\mathcal{I}(b)]_i = \begin{cases} i - \max\{j | j < i \text{ and } b_j = 1\} & \text{if } b_i = 1, \\ 0 & \text{otherwise,} \end{cases}$$

where $\max\{j : j < i \text{ and } b_j = 1\} = 0$ if no such j exists. In words, $\mathcal{I}(b)$ is a vector holding the distance between ones in b to the nearest one to their left[1].

Now, consider the following construction of w:

$$w_n = \begin{cases} \prod_{i=1}^{d} \alpha^*_{[\mathcal{I}(b(n))]_i} & \text{if } b(n) \overset{v}{\sim} X_{(t-d:t-1)} \\ 0 & \text{otherwise} \end{cases}$$

Our claim is that for the construction above, it holds that $\tilde{X}^p_t(w) = \tilde{X}_t(\alpha^*)$. Let us look first at $\tilde{X}_t(\alpha^*)$. By definition, it can be recursed until it "encounters" p successive revealed observations. Thus, it has the following structure:

$$\tilde{X}_t(\alpha^*) = \sum_{k=1}^{d} p_k(\alpha^* | X_{(t-d:t-1)}) X_{t-k}\mathbf{1}\{X_t\},$$

where $p_k(\alpha^*|X_{(t-d:t-1)})$ is a polynomial consisting of sums of products of $\alpha^*_1, \ldots, \alpha^*_p$. In fact, the polynomial $p_k(\alpha^*|X_{(t-d:t-1)})$ is a sum of elements, each of them is of the form $\prod_{i=1}^{d} \alpha^*_{[\mathcal{I}(b)]_i}$, where b is a valid path w.r.t. $X_{(t-d:t-1)}$ and in addition $k = \max\{i : b_i = 1\}$. This property follows directly from Definition 1 in Section 3.1. Thus, we can write

$$\begin{aligned}
\tilde{X}_t(\alpha^*) &= \sum_{k=1}^{d} p_k(\alpha^*|X_{(t-d:t-1)}) X_{t-k}\mathbf{1}\{X_{t-k}\} \\
&= \sum_{k=1}^{d} \sum_{b \in \mathcal{X}^v_t} \prod_{i=1}^{d} \alpha^*_{[\mathcal{I}(b)]_i} X_{t-k}\mathbf{1}\{k = \max\{i : b_i = 1\}\} \\
&= \sum_{n=1}^{2^d-1} \prod_{i=1}^{d} \alpha^*_{[\mathcal{I}(b(n))]_i} X_{t-m(n)}\mathbf{1}\{b(n) \overset{v}{\sim} X_{(t-d:t-1)}\} \\
&= \sum_{n=1}^{2^d-1} w_n \left[\Phi^p(X_{(t-d:t-1)})\right]_n \\
&= w^\top \Phi^p(X_{(t-d:t-1)}) = \tilde{X}^p_t(w),
\end{aligned}$$

where $\mathcal{X}^v_t$ is the set of all valid paths w.r.t. $X_{(t-d:t-1)}$. This establishes the claim in (b). Now, plugging (4) into (3) gives the stated result. □

Algorithm 3 Efficient Implementation of Algorithm 2

Input: learning rate η.
Set $w_1 = 0$.
for $t = 1$ to T **do**
  Predict $\tilde{X}_t$ and incur $\ell_t(X_t, \tilde{X}_t)\mathbf{1}\{X_t\}$
  Set prediction:
  $$\tilde{X}_{t+1} = \frac{-\eta \sum_{\tau=1}^{t} \mathrm{Err}_\tau K(t+1,\tau)}{\max\left\{1, \eta\sqrt{2^{-d}\sum_{s=1}^{t}\sum_{\tau=1}^{t} \mathrm{Err}_s \mathrm{Err}_\tau K(s,\tau)}\right\}}$$
end for

3.3. Efficient Implementation of Algorithm 2

Theorem 3.1 ignores computational considerations. Next, We show that $\tilde{X}_t(w^t)$ can in fact be generated efficiently. For $t = 1$ we have that $w_1 = 0$ and thus $\tilde{X}_1(w^1) = 0$. Next, assume that $\{\tilde{X}_\tau(w^\tau)\}_{\tau=1}^t$ are efficiently generated and prove for $\tilde{X}_{t+1}(w^{t+1})$. Now, notice that if we denote $\nabla \ell_\tau(w^\tau)\mathbf{1}\{X_t\} = \mathrm{Err}_\tau \Phi(X_{(\tau-d:\tau-1)})$, the above implies that Err_τ is known for all $\tau \le t$. Thus,

$$\begin{aligned}
\tilde{X}_{t+1}(w^{t+1}) &= \Phi(X_{(t-d+1:t)})^\top w^{t+1} \\
&= \frac{-\eta \sum_{\tau=1}^{t} \Phi(X_{(t-d+1:t)})^\top \nabla \ell_\tau(w^\tau)}{\max\{1, \eta\|\sum_{\tau=1}^{t} \nabla \ell_\tau(w^\tau)\| 2^{-d/2}\}} \\
&= \frac{-\eta \sum_{\tau=1}^{t} \mathrm{Err}_\tau K(t+1,\tau)}{\max\left\{1, \eta\sqrt{2^{-d}\sum_{s=1}^{t}\sum_{\tau=1}^{t} \mathrm{Err}_s \mathrm{Err}_\tau K(s,\tau)}\right\}},
\end{aligned}$$

where $K(s,\tau) = \Phi(X_{(s-d:s-1)})^\top \Phi(X_{(\tau-d:\tau-1)})$. The above is efficient to generate if and only if $K(s,\tau)$ is efficient to compute for all s and τ. This computation can be done in $O(d)$ computations despite the fact that $\Phi(X_{(s-d:s-1)}) \in \mathbb{R}^{2^d-1}$.

To see that, we first need to define an auxiliary function c as follows: $\left[c(s,\tau)\right]_k = \left[c(X_{(s-d:s-1)}, X_{(\tau-d:\tau-1)})\right]_k = \sum_{i=1}^{k-1}(1 - \mathbf{1}\{X_{s-i}\})(1 - \mathbf{1}\{X_{\tau-i}\})$. Essentially, $c(s,\tau)$ counts the number of relatively common missing observations in $X_{(s-d:s-1)}$ and $X_{(\tau-d:\tau-1)}$. Now, notice that

$$\begin{aligned}
K(s,\tau) &= \Phi(X_{(s-d:s-1)})^\top \Phi(X_{(\tau-d:\tau-1)}) \\
&= \sum_{n=1}^{2^d-1} \left[\Phi(X_{(s-d:s-1)})\right]_n \left[\Phi(X_{(\tau-d:\tau-1)})\right]_n \\
&= \sum_{n=1}^{2^d-1} X_{s-m(n)} X_{\tau-m(n)} \mathbf{1}\{b(n) \in \mathcal{X}^{sv}_s\}\mathbf{1}\{b(n) \in \mathcal{X} sv_\tau\} \\
&= \sum_{k=1}^{d} 2^{[c(s,\tau)]_k} X_{s-k} X_{\tau-k} \mathbf{1}\{X_{s-k}\}\mathbf{1}\{X_{\tau-k}\},
\end{aligned}$$

where $\mathcal{X}^{sv}_t$ is the set of all valid paths w.r.t. $X_{(t-d:t-1)}$, and the last equality follows from Definition 1.

[1] For instance, it holds that $\mathcal{I}((0,1,0,1)) = (0,2,0,2)$ and $\mathcal{I}((0,0,1,1)) = (0,0,3,1)$.

3.4. Some Extensions

We briefly discuss two issues: removal of assumption (2) and replacement of $\tilde{\mathbb{B}}_d$ with a smaller ball.

As mentioned before, assumption (2) makes sure that each prediction $\tilde{X}_t^{\text{REC}}(\alpha)$ considers at most d past observations. However, our algorithm is still applicable if this assumption does not hold. The theoretic guarantee then gets a different interpretation: we are almost as good as the best recursive $\text{AR}(p)$ predictor, but now the recursion is limited to at most d past observations. Essentially, this is equivalent to defining a family of recursive AR predictors with bounded memory, and compete against predictors in this family.

As for the second issue, recall that the dimension of the decision set $\tilde{\mathbb{B}}_d$ is exponential in d. This affects us only in the regret bound, as we proved earlier that computations can be done efficiently in our setting. To mitigate the regret bound effect, we define $\hat{\mathbb{B}}_d = \left\{ w \in \mathbb{R}^{2^d - 1} \ : \ \|w\|_2^2 \leq d \right\}$ and state the following corollary:

Corollary 3.2. *Algorithm 2 generates an online sequence $\{w_t\}_{t=1}^T$ for which it holds that:*

$$\mathcal{R}_T = \sum_{t=1}^{T} \ell_t(X_t, \tilde{X}_t(w^t))\mathbf{1}\{X_t\}$$

$$- \min_{\alpha \in \mathcal{K}} \sum_{t=1}^{T} \ell_t(X_t, \tilde{X}_t^{REC}(\alpha))\mathbf{1}\{X_t\} \leq 3GD\sqrt{\sum_{t=1}^{T} \mathbf{1}\{X_t\}},$$

where $\mathcal{K} = \left\{ \alpha \in \mathbb{R}^p \ : \ \alpha_i \leq \left(\frac{1}{\sqrt{2}}\right)^i \right\}$.

The proof follows by a simple calculation, and is thus omitted here. Note that in the above $D = \sqrt{d}$ and G is again determined by the selection of the loss functions. This case captures natural scenarios, in which the effect of past observations decays as they are more distant.

4. Illustrative Examples

The following experiments demonstrate the effectiveness of the proposed algorithm under various synthetic settings. We start by presenting some of the state-of-the-art baselines for the problem at hand.

4.1. Baselines

Most work on time series with missing observations considers what we call the *offline setting*: given a time series that contains missing observations, compute the model parameters (in our case, the AR coefficients) and\or impute the missing data. Our *online setting* can be seen as a sequential offline setting, in which at time t we are given the time series values up to time $t-1$ and our task is to predict the signal at time t. In light of this, we adapt the offline baselines presented below to the online setting. We note that this adaptation does not weaken the offline baselines in any way, and we use it only for comparison purposes.

Algorithm 4 OGDIMPUTE

Input: learning rate η
Initialize $\alpha^1 = 0$, and set $X_t = 0$ for $t \leq 0$
for $t = 1$ to T **do**
  Predict $\tilde{X}_t^{\text{AR}}(\alpha^t) = \sum_{i=1}^{p} \alpha_i^t X_{t-i}$
  Observe loss $\ell_t\big(X_t, \tilde{X}_t^{\text{AR}}(\alpha^t)\big)\mathbf{1}\{X_t\}$
  If $\mathbf{1}\{X_t\} = 0$, then set $X_t = \tilde{X}_t^{\text{AR}}(\alpha^t)$
  Set $\alpha^{t+1} = \Pi_{\mathcal{K}}\big(\alpha^t - \eta \nabla \ell_t\big(X_t, \tilde{X}_t^{\text{AR}}(\alpha^t)\big)\mathbf{1}\{X_t\}\big)$
end for

Yule-Walker estimator. We use the well-known Yule-Walker estimator, and adapt it to our setting as follows: at first, we initialize some AR coefficients. At time t, at our disposal is the data seen up to time $t-1$, where missing observations are filled by corresponding past predictions of the algorithm. Then, the AR coefficients are computed by solving the Yule-Walker equations, and a prediction is made accordingly.

Expectation Maximization (EM). Our EM baseline is based on the algorithm originally proposed by (Shumway & Stoffer, 1982). Roughly speaking, the algorithm assumes an underlying state-apace model and employs the Kalman filter to estimate its parameters. The estimation is done by maximizing the log-likelihood using iterative EM steps, and the resulting Kalman smoothed estimator is used to complete the missing observations.

ARLSIMPUTE. This algorithm was proposed by (Choong et al., 2009) to cope originally with missing observations in DNA microarray data. The algorithm is based on the following iterative method: at the first iteration, initialize all the missing observations to 0. Then, in every iteration compute LS estimator for the AR coefficients and update the value of the missing observations by maximizing the log-likelihood.

OGDIMPUTE. We propose another algorithm for the problem at hand, denoted Algorithm 4. Basically, the algorithm applies the standard Online Gradient Descent algorithm to learn the AR coefficients, while filling missing observations with their past predictions. Whereas this algorithm is very fast and simple to implement, its downside is the lack of theoretic guarantee.

4.2. Generating the Synthetic Data

To compare the performance of the proposed algorithms we design several different settings (presented below). In order to ensure the stability of the results we average them

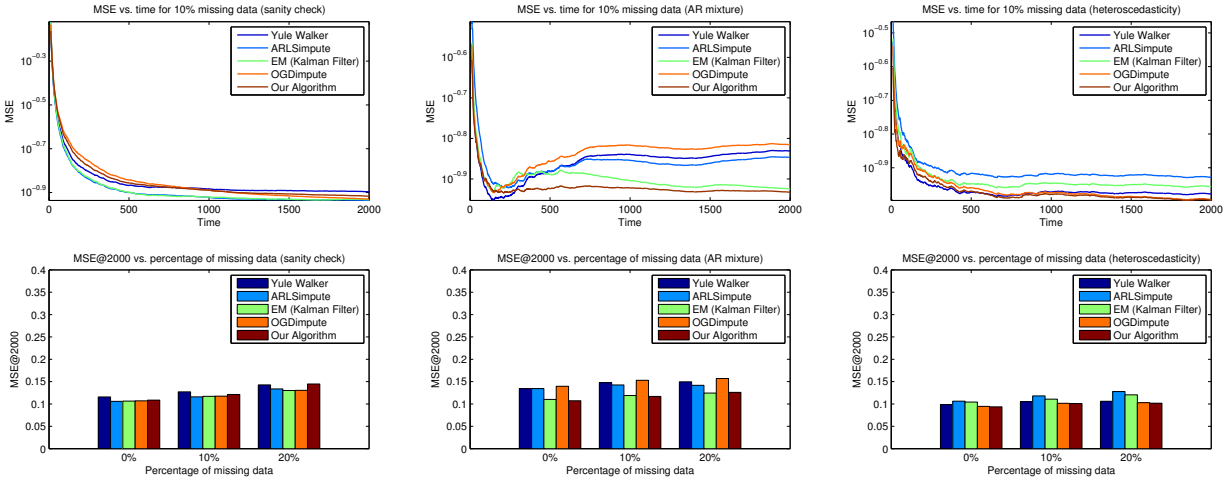

Figure 1. Experimental results for synthetic data.

	Setting 1. sanity check			Setting 2. AR mixture			Setting 3. heteroscedasticity		
	0%	**10%**	**20%**	**0%**	**10%**	**20%**	**0%**	**10%**	**20%**
Yule-Walker	0.1156	0.1270	0.1427	0.1343	0.1476	0.1496	0.0986	0.1023	0.1059
ARLSIMPUTE	**0.1058**	**0.1160**	0.1336	0.1344	0.1424	0.1417	0.1063	0.1150	0.1276
EM (Kalman filter)	0.1065	0.1170	**0.1301**	0.1101	0.1189	**0.1244**	0.1042	0.1078	0.1205
OGDIMPUTE	0.1071	0.1175	0.1305	0.1396	0.1530	0.1568	0.0945	0.0984	0.1028
Our algorithm	0.1085	0.1212	0.1447	**0.1072***	**0.1168***	0.1260	**0.0937**	**0.0979**	**0.1018**

Table 1. Experimental results for synthetic data.

over 50 runs. For our algorithm, we used $d = 3p$ in all considered settings. In our tables, we mark with bold font the best results, and add an asterisk to indicate significance level of 0.05.

Setting 1 (sanity check). We generate a time series using the coefficient vector $\alpha = [0.6, -0.5, 0.4, -0.4, 0.3]$ and i.i.d. noise terms that are distributed $\mathcal{N}(0, 0.3^2)$. We then omit some of the data points in a random manner (that is, each data point is omitted with a certain probability).

Setting 2 (AR mixture). Our motivation in this setting is to examine the functionality of the different algorithms when faced with changing environments. Thus, we consider a predefined set of AR coefficients, and generate time series by alternating between them in a random manner. We add an additive noise which is distributed $Uni[-0.5, 0.5]$. Here also, the missing data is omitted in a random manner.

Setting 3 (heteroscedasticity). Here we test the robustness of the different algorithms to unequally distributed noise terms. Thus, we generate a time series using the coefficient vector $\alpha = [0.11, -0.5]$, and noise terms that are distributed normally, with expectation that is the value of the previous noise term and variance 0.3^2. This implies that consecutive noise terms are positively correlated. The missing data points are chosen randomly here as well.

As evident in Figure 1 and Table 1, our online algorithm outperforms the other algorithms when the time series exhibits some complicated structure. In the case where the error terms are Gaussian and the time series complies with the AR model (sanity check), we can see that all algorithms perform roughly the same, as can be expected by the theoretical guarantees. We point out that whereas the offline algorithms (especially the EM based and ARLSIMPUTE) require rather large computational power, the two online algorithms are fast and quite simple to implement.

5. Discussion and Conclusion

In this work we studied the problem of time series prediction using the AR model in the presence of missing data. We considered a setting in which the signal, along with the missing data, are allowed to be arbitrary. We then defined the notion of learning in this setting with respect to the best (in hindsight) AR predictor, and showed that we can be almost as good as this predictor.

It remains for future work to study whether the dependence on the parameter d in the regret bound can be improved. It would also be interesting to study whether our approach could be extended to more complex time series models, such as ARCH, ARMA, and ARIMA.

Acknowledgments

The research leading to these results has received funding from the European Unions Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 336078 – ERC-SUBLRN. We would also like to acknowledge the work of (Hazan et al., 2015), which incited the non-proper learning technique in this work.

References

Anava, Oren, Hazan, Elad, Mannor, Shie, and Shamir, Ohad. Online learning for time series prediction. *arXiv preprint arXiv:1302.6927*, 2013.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, learning, and games*. Cambridge University Press, 2006.

Choong, Miew Keen, Charbit, Maurice, and Yan, Hong. Autoregressive-model-based missing value estimation for dna microarray time series data. *Information Technology in Biomedicine, IEEE Transactions on*, 13(1): 131–137, 2009.

Dempster, Arthur P, Laird, Nan M, and Rubin, Donald B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

Ding, Jie, Han, Lili, and Chen, Xiaoming. Time series ar modeling with missing observations based on the polynomial transformation. *Mathematical and Computer Modelling*, 51(5):527–536, 2010.

Dunsmuir, William and Robinson, PM. Estimation of time series models in the presence of missing data. *Journal of the American Statistical Association*, 76(375):560–568, 1981.

Durbin, James and Koopman, Siem Jan. *Time series analysis by state space methods*. Number 38. Oxford University Press, 2012.

Hazan, Elad. The convex optimization approach to regret minimization. *Optimization for machine learning*, pp. 287, 2011.

Hazan, Elad, Livni, Roi, and Mansour, Yishay. Classification with low rank and missing data. *CoRR*, abs/1501.03273, 2015.

Honaker, James and King, Gary. What to do about missing values in time-series cross-section data. *American Journal of Political Science*, 54(2):561–581, 2010.

Kalman, Rudolph Emil. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

Liu, Xiangheng and Goldsmith, Andrea. Kalman filtering with partial observation losses. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 4, pp. 4180–4186. IEEE, 2004.

Shalev-Shwartz, Shai. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

Shumway, Robert H and Stoffer, David S. An approach to time series smoothing and forecasting using the em algorithm. *Journal of time series analysis*, 3(4):253–264, 1982.

Sinopoli, Bruno, Schenato, Luca, Franceschetti, Massimo, Poolla, Kameshwar, Jordan, Michael I, and Sastry, Shankar S. Kalman filtering with intermittent observations. *Automatic Control, IEEE Transactions on*, 49(9): 1453–1464, 2004.

Wang, Zidong, Yang, Fuwen, Ho, Daniel WC, and Liu, Xiaohui. Robust finite-horizon filtering for stochastic systems with missing measurements. *Signal Processing Letters, IEEE*, 12(6):437–440, 2005.