# Model Selection in Online Learning for Times Series Forecasting

Waqas Jamil and Abdelhamid Bouchachia

Bournemouth University,
Machine Intelligence Group.
{wjamil,abouchachia}@bournemouth.ac.uk

**Abstract.** This paper discusses the problem of selecting model parameters in time series forecasting using aggregation. It proposes a new algorithm that relies on the paradigm of prediction with expert advice, where online and offline autoregressive models are regarded as experts. The desired goal of the proposed aggregation-based algorithm is to perform not worse than the best expert in the hindsight. The theoretical analysis shows that the algorithm has a guarantee that holds for any data sequence. Moreover, the empirical evaluation shows that the algorithm outperforms other popular model selection criteria such as Akaike and Bayesian information criteria on cyclic behaving time series.

**Keywords:** Model selection; Online learning; Aggregation Algorithm; Time series.

## 1 Introduction

Model selection is about choosing a model from a set of fitted models that performs better on a given data. In statistics Akaike information criterion (AIC) [1] and Schwarz criterion [17] are popular model selection techniques. These criteria are not competing rules since they are useful for different scenarios. For instance, AIC achieves asymptotic efficiency [18], while BIC originates from Bayesian hypothesis testing of the regular exponential family using an asymptotic approximation to identify the correct model when the sample size increases [14]. These criteria are developed by considering batch learning, and sometimes they achieve a slower rate of convergence as shown in [5] where the presented algorithm has a similar flavour to the fixed-share algorithm [7].

To fit a model one needs to input the coefficients and the parameters. In batch learning this can be done by for example using cross validation. In contrast, in online learning due to the sequential arrival of the data methods like cross validation can't be used. Over the past, numerous attempts have been made to address the problem of online model selection of time series. For instance, Noshad et al. [12] proposed a dynamic algorithm that operates sequentially to select a suitable model. In [10] an adaptive algorithm is used for automatically selecting the best model but is not applicable generally. It only allows to reduce data communication in wireless sensor networks. The approach discussed by Prado and Lopes [13] addresses the issue of parameter selection in the state space representation of time series. Sato [16] uses variational Bayes to provide complete online model selection mechanism with a guarantee, by averaging over an ensemble of

models. Our work differs in the sense that guarantee is held for sure, not on average or high probability as discussed in the past.

In this study, we investigate model selection in the context of online time series forecasting. There have been some recent but very scarce attempts to address the problem of time series prediction using online learning. For instance, Anava et al. [2] proposed an online version of Auto-Regressive-Moving-Average (ARMA), while Liu et al. [11] investigated online Auto-Regressive-Integrated-Moving-Average (ARIMA). The fundamental idea behind the online version of ARMA and ARIMA models is that ARMA is a subset of AR. The online ARIMA model is an extension of the online ARMA model. These papers present a reasonable approach to handle the uni-variate time series prediction problem.

This paper presents an approach that uses aggregation, similarly to the approach presented by Romanenko [15], but focuses on mixing online ARMA models leading to a novel utterly online framework. The proposed approach does not require the use of any information criterion and has the guarantee of not being too far from the best model. Furthermore, it has the possibility of beating the best model for time series prediction. The bounds of the competitive online algorithms, such as the strong Aggregation Algorithm (AA) by Vovk [21], are guaranteed to hold. That is, the error bounds of competitive online statistics algorithms do not contain only with high probability or on average as one can encounter with many forecast combination algorithms, but such bounds do hold with certainty.

In our work we include a formal proof of the bound of the ARMA-OGD along with derivations of AA's substitution functions associated with square loss prediction games. The novelty of our work lies in the modification of AA and ARMA Gradient Descent (ARMA-OGD) algorithms. More precisely:

1. We plug ARMA-OGD into AA to perform online model selection.
2. We explicitly show that for our suggested approach, the following type of bound holds regardless of the data generating mechanism:

$$L_T \leq L_T^* + \frac{1}{\eta} \log n$$

where $L_T$ is the cumulative loss incurred by of the learning algorithm up until time $T$, $L_T^*$ is the cumulative loss of the best learning strategy in the hindsight. The learning rate is denoted by $\eta$ and $n$ is a finite integer denoting the number of experts.

Following is the organisation. In next two sections we provide context to our work by briefly discussing the essential features of Aggregation Algorithm (AA) and ARMA-OGD that we later are used to combine the two algorithms. In section 3 we provide an explicit algorithm that combines AA and ARMA-OGD by modifying them. Section 4 illustrates the guarantee of AA+ARMA-OGD on two real world datasets. Section 5 concludes our work.

## 2 Background

### 2.1 Aggregating Algorithm

Let $\Gamma$ be the prediction space and $\Omega = [Y_1, Y_2]$ be the outcome space, such that $Y_2 > Y_1$, the $n$ number of experts $\theta_k$ for $k = 1, 2, ..., n < \infty$, makes predictions $\gamma_t^{\theta_k} \in \Gamma$ on each trial $t \in \mathbb{Z}$; the learner makes a prediction by aggregating experts predictions; nature chooses an outcome; each expert $Loss_{expert} = \sum_{t=1}^{T} \lambda(\gamma_t^{\theta_k}, \omega_t)$ and learner loss $Loss_{learner} = \sum_{t=1}^{T} \lambda(\gamma_t, \omega_t)$ is calculated using square loss. It is not assumed that there is a model generating the outcomes and the nature is considered as an oblivious adversary. The initialisation of the experts weights is done uniformly (each expert is assigned the same weight initially). AA [21, 22] works under the protocol of Prediction With Experts Advice (PWEA), which is as follows:

**Protocol 1** Prediction With Expert Advise

1: **for** $t = 1, 2, ...$ **do**
2: Experts $\theta_k \in \Theta$ predicts $\gamma_t^{\theta_k} \in \Gamma,\ k = 1, 2, ..., n$
3: Learner output $\gamma_t \in \Gamma$
4: Nature output $\omega_t \in \Omega$
5: Learner suffers loss $\lambda(\gamma_t, \omega_t)$
6: Experts $\theta \in \Theta$ suffers loss $\lambda(\gamma_t^{\theta_k}, \omega_t)$
7: **end for**

AA generalises the weighted majority algorithm providing an exponentially weighted average that has bounds in the case of mixable game. For $\eta > 0$, a loss function is called $\eta-$mixable if there exists a substitution function (more on it later) for it such that [19, 21]:

$$\lambda(\omega, \gamma) \leq g(\omega) = \log_\beta \int \beta^{(\omega - p)^2} P(dp) \tag{1}$$

where $\forall \gamma \in \mathbb{R}$, $\lambda(Y_1, \gamma) \leq g(Y_1)$ & $\lambda(Y_2, \gamma) \leq g(Y_2)$ such that $\beta = e^{-\eta}$ for $\eta > 0$, and $g$ represents the generalised prediction corresponding to the probability distribution $P \in \mathbb{R}$, such that $\omega, p \in [Y_1, Y_2]$ for $Y_2 > Y_1$.

The loss of AA cannot be much larger than that of the best expert, for a mixable finite experts game by equally initialising the weights of the experts.

$$Loss(AA) \leq Loss_{best}(\theta) + \frac{\log n}{\eta} \tag{2}$$

where $\theta \in \Theta$, $\eta$ is the learning rate, and $n$ is the number of experts. This bound (eq. 2) is shown in [20] to be optimal i.e. it cannot be improved by any other prediction algorithm. It is for this reason, that we have chosen AA in this paper to apply for time series prediction.

AA takes two parameters, the learning rate $\eta > 0$ and a prior probability which indicates the initial weights of the experts. At every step $t$, we update the weights. So intuitively, if an expert makes a mistake, we would reduce its weight. AA uses a *substitution function* which maps the generalised prediction $g(\omega)$ into $\Gamma$,

**Algorithm 1** Aggregation Algorithm

1: Initialise weights $w_0^\theta, \theta = 1, 2, ...n$
2: **for** $t = 1, 2, ..$ **do**
3: Notice experts prediction $\gamma_t^\theta$
4: Normalise experts weight $w_t^\theta = \frac{w_{t-1}^\theta}{\sum_{i=1}^N w_{t-1}^i}$
5: Use substitution function to obtain $\gamma_t$
6: Notice actual outcomes $\omega_t$
7: Update the experts weights $w_t^\theta = w_t^\theta e^{-\eta\lambda(\gamma_t^\theta, \omega_t)}$
8: **end for**

Next we explain line 5 of Algorithm 1 by focusing on work done in [21]. Consider $\Omega = \{-1, 1\}$ and $\gamma \in [-1, 1]$. AA's prediction without using the substitution function is $(g(-1), g(1))$ (a point on a plane), which does not lie on the losses curve $((-1-\gamma)^2, (1-\gamma)^2)$. AA's prediction $(g(-1), g(1))$ is transformed to the point $(e^{-\eta(-1-\gamma)^2}, e^{-\eta(1-\gamma)^2})$ by the use of substitution function and the set of permitted predictions becomes $(e^{-\eta(-1-\gamma)^2}, e^{-\eta(1-\gamma)^2})$ (for more details see [19]). To find the learning rate $\eta$, for which the curve $(e^{-\eta(-1-\gamma)^2}, e^{-\eta(1-\gamma)^2})$ is convex is equivalent to the problem of finding the values of second derivative for which $(e^{-\eta(-1-\gamma)^2}, e^{-\eta(1-\gamma)^2})$ is less or equal to 0 for all values of $\gamma \in [-1, 1]$. Therefore:

$$(u, v) = (e^{-\eta(-1-\gamma)^2}, e^{-\eta(1-\gamma)^2})$$

$$\frac{\partial u}{\partial \gamma} = -2\eta(1+\gamma)e^{-\eta(1+\gamma)^2}$$

$$\frac{\partial v}{\partial \gamma} = 2\eta(1-\gamma)e^{-\eta(1-\gamma)^2}$$

**Lemma 1.** *The restricted square loss game is $\eta$-mixable if and only if $\eta \leq \frac{1}{2}$.*

*Proof.* Applying the chain rule we obtain:

$$\frac{\partial v}{\partial u} = \frac{2\eta(1-\gamma)e^{-\eta(1-\gamma)^2}}{-2\eta(1+\gamma)e^{-\eta(1+\gamma)^2}} = -\frac{1-\gamma}{1+\gamma}e^{4\eta\gamma} = \frac{(\gamma-1)e^{4\eta\gamma}}{\gamma+1}$$

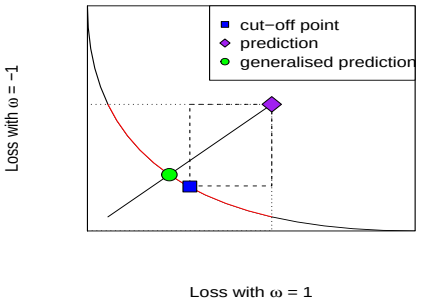


**Fig. 1.** $((-1-\gamma)^2, (1-\gamma)^2)$ curve where $\gamma \in [-1, 1]$.

The inf-sup of the ratio $\frac{g(\omega)}{\lambda(\omega,\gamma)}$ is obtained where the line $((0,0), (g(1), g(-1))$ intersects with the losses curve. From Fig 1 the intersection between losses (the line intersecting the red curve) and the between the (red) curve is at $((\gamma-1)^2, (\gamma+1)^2)$ (cut-of point in green see Fig 1), thus the inf-sup of the ratio is $\frac{(\gamma-1)^2}{(\gamma+1)^2} = \frac{g(1)}{g(-1)}$, which is non-linear (difficult to use in practice), so instead we use a different point (prediction point in blue, see Fig 1) on the curve (prediction and generalised prediction are mapped by a square in Fig 1).

To achieve the minimum or inflection point, the second derivative must be negative or null, we proceed as follows:

$$\frac{\partial^2 v}{\partial u^2} = \frac{\partial v}{\partial u \partial d\gamma}\frac{\partial \gamma}{\partial u} = \frac{e^{4\eta\gamma}}{(1+\gamma)^2}(4\eta(1-\gamma)(1+\gamma)-2)2\eta(1+\gamma)e^{-\eta(1+\gamma)^2}$$

$$= \frac{e^{4\eta\gamma}}{(1+\gamma)^2}\left(\frac{4\eta(1-\gamma^2)-2}{2\eta(1+\gamma)e^{-\eta(1+\gamma)^2}}\right) \tag{3}$$

The term in equation (3) will be negative or zero if and only if: $4\eta(1-\gamma^2)-2 \leq 0$ which implies $\eta \leq 0.5$, since $\gamma^2 \in [0,1]$. □

**Proposition 1.** *For a game of square loss with $\Omega = \{-1, 1\}$, then $\gamma = \frac{g(-1)-g(1)}{4}$ is a substitution function.*

*Proof.* The curve $((\gamma-1)^2, (\gamma+1)^2)$ for $\gamma \in [-1,1]$ contains all possible values of $\gamma$. The point $(g(1), g(-1))$ represents generalised prediction. The substitution function maps generalised prediction to actual predictions, thus $(\gamma+1)^2 - g(-1) = (\gamma-1)^2 - g(1)$. By doing simple algebraic manipulation, we get: $\gamma = \frac{g(-1)-g(1)}{4}$ □

**Lemma 2.** *The square loss game $\Omega = [-Y, Y]$ where $Y \in \mathbb{R}$ is $\eta-$mixable if and only if $\eta \leq \frac{1}{2Y^2}$.*

*Proof.* We find the values of $\eta$ for which the game is mixable by exponentiation of the generalised prediction. We have $e^{(-Y-\gamma)^2}$ and $e^{(Y-\gamma)^2}$ where $\gamma \in [-Y, Y]$. If we instead use $e^{\frac{\hat{\eta}}{Y^2}(-Y-\gamma)^2}$ and $e^{\frac{\hat{\eta}}{Y^2}(Y-\gamma)^2}$ our game becomes restricted square loss game for which as seen in Lemma 1 the game is mixable if and only if, $\eta \leq 0.5$. By writing $\eta = \frac{\hat{\eta}}{Y^2}$, we have $\eta Y^2 \leq 0.5$ which implies that $\eta \leq \frac{1}{2Y^2}$. □

**Proposition 2.** *For a square loss game with $\Omega = [-Y, Y]$, where $Y \in \mathbb{R}$ then:*

$$\gamma = \frac{g(-Y) - g(Y)}{4Y}$$

*is a substitution function.*

*Proof.* By solving $(\gamma+Y)^2 - g(-Y) = (\gamma-Y)^2 - g(Y)$, we obtain our desired result. □

**Lemma 3.** *The square loss game $\Omega = [Y_1, Y_2]$ where $Y_1, Y_2 \in \mathbb{R}$ and $Y_1 < Y_2$ is $\eta-$mixable if and only if $\eta \leq \frac{2}{(Y_2-Y_1)^2}$.*

*Proof.* We need to prove for the curve $(u, v) = (e^{-\eta(\gamma-Y_1)^2}, e^{-\eta(\gamma-Y_2)^2})$ that:

$$\frac{\partial^2 v}{\partial u^2} = \frac{\frac{\partial^2 v}{\partial \gamma \partial u}}{\frac{\partial u}{\partial \gamma}} \leq 0$$

By performing above differentiation for the curve, we get $\frac{1}{Y_1-\gamma} + 2\eta(Y_2-\gamma) \leq 0 \Rightarrow \eta \leq \frac{1}{2(Y_2-\gamma)(\gamma-Y_1)}$. We notice that $\max_{\gamma\in[Y_1,Y_2]}(Y_2-\gamma)(\gamma-Y_1) = \frac{1}{4}(Y_2-Y_1)^2$ and the curve is concave $\forall\gamma$ provided that $\eta \leq \frac{2}{(Y_2-Y_1)^2}$. □

**Proposition 3.** *For a square loss game with* $\Omega = [Y_1, Y_2]$*, where* $Y_1, Y_2 \in \mathbb{R}$ *and* $Y_1 < Y_2$ *then:*

$$\gamma = \frac{Y_2 + Y_1}{2} - \frac{g(Y_2) - g(Y_1)}{2(Y_2 - Y_1)}$$

*is a substitution function.*

*Proof.* To find $\gamma$, consider $(Y_1 - \gamma)^2 + g(Y_1) = (Y_2 - \gamma)^2 + g(Y_2)$. By using the fact $(Y_1^2 - Y_2^2) = (Y_1 + Y_2)(Y_1 - Y_2)$ and re-arranging, $2\gamma(Y_2 - Y_1) = g(Y_2) - g(Y_1) - (Y_1 + Y_2)(Y_1 - Y_2)$, we get the substitution function. □

### 2.2 ARMA-OGD

**Algorithm 2** ARMA-OGD$(p, q)$

1: ARMA order $p, q$, Learning rate $\eta$, $m = q.\log_{1-\beta}\left((TLM_{max})^{-1}\right)$
2: **for** $t = 1, 2, ...(T-1)$ **do**
3: Predict $\hat{X}_t(\gamma_t) = \sum_{i=1}^{m+k} \gamma_t^i X_{t-i}$
4: Observe $X_t$ and suffer loss $\lambda^m(\gamma_t, \omega_t)$
5: Let $\nabla_t = \nabla\lambda^m(\gamma_t, \omega_t)$
6: Set $\gamma_{t+1} \leftarrow \prod_{\mathcal{K}}\left(\gamma_t - \frac{1}{\eta}\nabla_t\right)$
7: **end for**

ARMA-OGD$(p, q)$ was introduced by Anava et al. [2]. The psudo-code of the algorithm is presented in Algorithm 2. We proceed by defining some notation. The prediction set $\mathcal{K}$ contains $m + p-$dimensional coefficient vectors and is defined as $\mathcal{K} = \{\gamma \in \mathbb{R}^{m+p}, |\gamma_j| \leq c, j = 1, ..., m\}$. We denote the diameter of $\mathcal{K}$ by $D$ and bound $D = 2c\sqrt{(m+p)}$. The upper bound of convex loss $|| \nabla \lambda(\gamma, \omega)||$ for all $t$ $\gamma \in \mathcal{K}$ on sequence $|X_t| \leq X_{max}$, is denoted by $G = D(X_{max})^2$. We say $M_{max}$ is the upper bound on $|W_t|$ for all $t = 1, 2, ..., T$ if we assume that noise is adversarial and when noise are i.i.d then $\mathbb{E}(|\beta_t|) < M_{max} < \infty$ and $L$ denotes Lipshitz constant which is assumed to be greater than zero. The coefficients $|\alpha_i|$ are less then some constant $c \in \mathbb{R}$ and $\sum_{i=1}^{q} |\theta_i| < 1 - \beta$ where $\beta > 0$. We next present the proof of Theorem 5 mentioned in [2] but not shown due to the similarity to Theorem 1 of their paper.

**Theorem 1.** *For any data sequence* $\{X_t\}_{t=1}^T$ *such that* $p, q \geq 1$*, and set* $\eta = \frac{1}{X_{max}^2\sqrt{T}}$*, Algorithm 2 predicts using a convex loss function, with the following guarantee:*

$$\sum_{t=1}^{T} \lambda(\gamma_t, \omega_t) - min_{\alpha,\beta} \sum_{t=1}^{T} \mathbb{E}[f_t(\alpha, \beta)] = O(4c(m+p)X_{max}^2\sqrt{T})$$

*Proof.* Let $(\alpha^*, \beta^*) = \text{argmin}_{\alpha,\beta} \sum_{t=1}^{T} \mathbb{E}[f_t(\alpha, \beta)]$.We know for any convex loss function we have [23]:

$$\sum_{t=1}^{T} \lambda^m(\gamma_t, \omega_t) - \min \sum_{t=1}^{T} \lambda^m\left(X_t, \left(\sum_{i=1}^{m+k} \gamma_t^i X_{t-i}\right)\right) = O(4c(m+p)X_{max}^2\sqrt{T})$$

Now by using the fact that ARMA$(p, q)$ can be represented by AR$(\infty)$ [6], by using entire past, we can recursively write:

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

plugging in initial condition $X_t^\infty = X_1$, we get the loss suffered as:

$$f_t^\infty(\alpha, \beta) = \lambda(X_t, X_t^\infty(\alpha, \beta))$$

which is not convex. The loss function here considers entire data. We need to replace $f_t^\infty$ by $f_t$, which can be done by considering some weight $w_i(\alpha, \beta)$ function and write our loss function as follows:

$$f_t^\infty(\alpha, \beta) = \lambda(X_t, \sum_{i=1}^{t} w_i(\alpha, \beta) X_{t-i})$$

This allows the loss function to update prediction by using only the last outcome in contrast to using the entire history. By setting $m \in \mathbb{N}$, the prediction can be rewritten as:

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

Plugging in the initial condition $X_t^m(\alpha, \beta) = X_t$ for all $t, m \leq 0$, the loss suffered by the prediction at time $t$ becomes:

$$f_t^m(\alpha, \beta) = \lambda(X_t, X_t^m(\alpha, \beta))$$

By considering last $(m + k)$ observations and since $\min_\gamma \lambda^m(\gamma_t, \omega_t) \leq f_t^m(\alpha^* \beta^*)$ (Lemma 2 in [2]), we have:

$$\sum_{t=1}^{T} \lambda^m(\gamma_t, \omega_t) - \sum_{t=1}^{T} f_t^m(\alpha^* \beta^*) = O(4c^2(m+p) X_{max}^2 \sqrt{T})$$

From Lemma 3 in [2] we know that the following holds:

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

$$\sum_{t=1}^{T} \lambda^{q \log_{1-\epsilon}\left((TLM_{max})^{-1}\right)}(\gamma_t, \omega_t) - \sum_{t=1}^{T} f_t^{q \log_{1-\epsilon}\left((TLM_{max})^{-1}\right)}(\alpha^*, \beta^*)$$

$$= \sum_{t=1}^{T} \lambda(\gamma_t, \omega_t) - \min_{\alpha,\beta} \sum_{t=1}^{T} \mathbb{E}[f_t(\alpha, \beta)]$$

From Lemma 4 in [2] we know that the following holds:

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

$$\sum_{t=1}^{T} \lambda^{q \log_{1-\epsilon}\left((TLM_{max})^{-1}\right)}(\gamma_t, \omega_t) - \sum_{t=1}^{T} f_t(\alpha^*, \beta^*)$$

$$= O\left(4c \left(q \log_{1-\epsilon} (TLM_{max})^{-1} + p\right) X_{max}^2 \sqrt{T}\right)$$

which was to be proven. □

## 3 AA+ARMA-OGD

In this section, we provide an explicit algorithm for AA+ARMA-OGD$(p, q)$ (Algorithm 3). Each of our expert is an ARMA-OGD model with different values of parameters $p, q$. To obtain a competitive guarantee we combine the ARMA-OGD models using AA. Algorithms 2 uses Online Gradient Decent (OGD). The analysis done in [23] shows that the OGD attains the following regret when the learning rate is defined to be $\frac{1}{\sqrt{t}}$:

$$L_T - L_T^* = \mathcal{O}\left(\sqrt{T}\right) \tag{4}$$

where $L_T$ denotes the cumulative loss of the algorithm and $L_T^*$ denotes the cummulative loss of the best strategy in the hindsight. We now explain the details of Algorithm 2 projection step with the aid of Fig 2. In OGD we have some prediction $\gamma$ which is a point in a convex set. For a given convex loss function we move in the direction of the first derivative (gradient) of the loss incurred at time $\lambda(\gamma_t, \omega_t)$. By moving in the direction of the gradient we might go outside the convex set as there is no restriction that will stop us from going out the convex set (notice $\lambda(\gamma_t, \omega_t)$ is slightly outside the sphere in Fig 2). To keep the prediction inside the convex set, we do a projection by finding the closest point in the convex set to the point we chose i.e. we predict $\gamma_{t+1} = \prod_{\mathcal{K}}\left(\gamma_t - \frac{1}{\eta}\nabla\lambda(\gamma_t, \omega_t)\right)$, where $\nabla\lambda(\gamma_t, \omega_t)$ denotes the gradient of the current loss and $\prod_{\mathcal{K}}$ represents Euclidean projection onto set $\mathcal{K}$ i.e. $\prod_{\mathcal{K}}(\gamma) = \text{argmin}||\gamma - x||_2$.

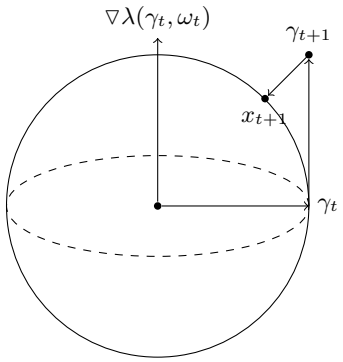


**Fig. 2.** Online Gradient Descent

Algorithm 3 has the following guarantee that holds for all $T$, regardless of the data generating mechanism:

$$Loss_{\text{Best ARMA-OGD}} - Loss_{\text{AA+ARMA-OGD}} \geq -\frac{\log n}{\eta} \tag{5}$$

where $n$ denotes the number of experts and for the details of the learning rate please see Lemma 3.

**Algorithm 3** AA+ARMA-OGD $(p, q)$

Input for each expert parameters $p^{\theta_{1,\dots,n}}$,$q^{\theta_{1,\dots,n}}$, $\eta > 0$. Initialise experts weight $w_0^{\theta_k} = \mathbf{1}$
**for** $k = 1, 2, ..., n$ **do**
  **for** $t = (max(p^{\theta_k}, q^{\theta_k}) + 1), ...$ **do**
    Read experts predictions $\gamma_t^{\theta_k} = \hat{X}_t^{\theta_k}(\hat{\gamma}_t^{\theta_k}) = \sum_{i=1}^{p^{\theta_k}} \alpha_t^{\theta_k} X_{t-i}^{\theta_k} + \sum_{i=1}^{q^{\theta_k}} \beta_t^{\theta_k} \epsilon_{t-i}^{\theta_k}$
    Normalise experts weights $w_t^{\theta} = \frac{w_{t-1}^{\theta}}{\sum_{i=1}^{N} w_{t-1}^{i}}$
    Predict $\gamma_t = \frac{Y_2+Y_1}{2} + \frac{g(Y_2)-g(Y_1)}{2(Y_2-Y_1)}$ # This is AA+ARMA-OGD prediction using proposition 3
    Notice actual outcomes $\omega_t \in \mathbb{R}$
    Calculate error $\epsilon_t^{\theta_k} = \gamma_t^{\theta_k} - \omega_t$ # notice $\omega_t$ is a value, so $\omega_t$ is subtracted from each experts $(k = 1, 2, ...n)$ prediction $\gamma_t^{\theta_k}$.
    Average $\epsilon_t^{\theta_k} = \frac{\sum_i^t \epsilon_i^{\theta_k}}{t - max(p^{\theta_k}, q^{\theta_k})}$
    Apply Gradient Decent on $\alpha^{\theta_k}$ and $\beta^{\theta_k}$

$$\alpha_{OGD}^{\theta_k} = -2\epsilon_t^{\theta_k} \sum_{i=1}^{p^{\theta_k}} X_{t-i}^{\theta_k} \ , \ \beta_{OGD}^{\theta_k} = -2\epsilon_t^{\theta_k} \sum_{i=1}^{q^{\theta_k}} \epsilon_{t-i}^{\theta_k}$$

    Calculate $\alpha^{\theta_k}$ and $\beta^{\theta_k}$:

$$\alpha_t^{\theta_k} = \alpha_{t-1}^{\theta_k} - \frac{\alpha^{\theta_k}}{\sqrt{t}} \alpha_{OGD}^{\theta_k} \ , \ \beta_t^{\theta_k} = \beta_{t-1}^{\theta_k} - \frac{\beta^{\theta_k}}{\sqrt{t}} \beta_{OGD}^{\theta_k}$$

    Project $\alpha^{\theta_k}$ and $\beta^{\theta_k}$ to simplex:

$$\alpha_t^{\theta_k} = \frac{\alpha_{t-1}^{\theta_k}}{max\left(1, \sum_{i=1}^{t} \sqrt{(\alpha_t^{\theta_k})^2}\right)} \ , \ \beta_t^{\theta_k} = \frac{\beta_{t-1}^{\theta_k}}{max\left(1, \sum_{i=1}^{t} \sqrt{(\beta_t^{\theta_k})^2}\right)}$$

    Update the experts weights $w_t^{\theta_k} = w_t^{\theta_k} e^{-\eta(\epsilon_t^{\theta_k})^2}$
  **end for**
**end for**

## 4 Empirical evaluation

Fig 3 shows the behaviour of the two-time series, [4] and [3]. The two time-series refers to 3650 days and exhibits cyclic (stationary) behaviour. Minimum temperature time series lies in the range $[-0.8, 26.3]$ and maximum temperature time series lies in the range $[7, 43.3]$. By using Lemma 3, we calculate $\eta \approx 0.0027$ and $\eta \approx 0.0015$.

We set five ARMA-OGD$(p, q)$, $p = 1, 2, 3, 4, 5$ and $q = 0$ as our experts. We call the ARMA-OGD with the least loss as Best Online ARMA-OGD (BOARMA-OGD). Notice in Fig 4 it is shown that the guarantee (5) given by Algorithm 3 holds. For minimum and maximum temperature time series the the right side of the inequality (5) is $-\frac{\log 5}{\eta} \approx -591$ and $-\frac{\log 5}{\eta} \approx -1060$ respectively.

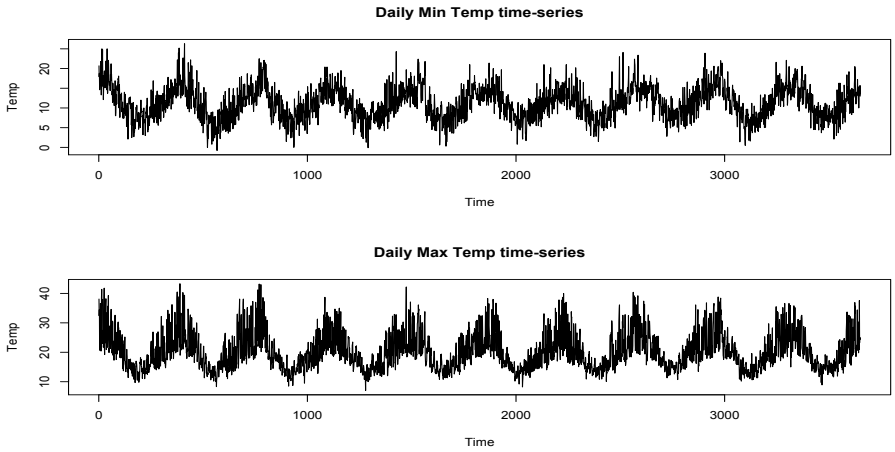


**Fig. 3.** Minimum and maximum temperature time-series

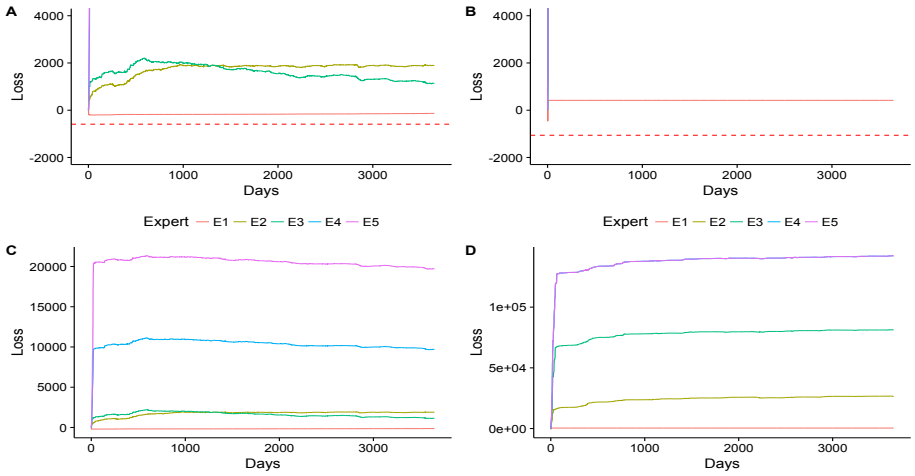


**Fig. 4.** Theoretical guarantee AA+ARMA-OGD. Plot A is zoomed plot C and both refer to the minimum temperature time series. Similarly, B is zoomed D and refers to the maximum temperature time series. The dotted red lines in plot A and B refer to AA+ARMA-OGD guarantee.

For the sake of comparison, we fit statistical ARMA model with fourier series [8]:

$$Y_t = \sum_{m=1}^{K} \left[ \alpha_m \sin\left(\frac{2\pi m t}{L}\right) + \beta_m \cos\left(\frac{2\pi m t}{L}\right) \right] + X_t \tag{6}$$

where $X_t$ is stationary ARMA/ARIMA$(p, q)$, $\alpha \in \mathbb{R}^p$, $\beta \in \mathbb{R}^q$, and $Y_t$ is periodic on interval $[-L, L]$. We choose parameters of (6) using AIC and call it the Best Batch

ARMA (BBARMA) model. We then fit a set of ARMA models and perform aggregation using AA (AA+BARMA) for the details on the set of batch ARMA models used please see [9]. Table 1 reports the cumulative losses of all the fitted models. Our suggested Algorithm AA+ARMA-OGD is the best performing model on both time series, but this is not what the model guarantees. The guarantee is that in the worst case the model will be close to BOARMA-OGD. We may say usually AA+ARMA-OGD will outperform the best performing model when there are several models performing close to each other. The prediction quality of AA+ARMA-OGD depends on the quality of the underlying experts.

**Table 1.** Cumulative losses.

| Model | Min temp | Max temp |
|---|---|---|
| BBARMA | 28097 | 105632 |
| AA+BARMA | 28049 | 102188 |
| BOARMA-OGD | 27768 | 86550 |
| AA+ARMA-OGD | 27634 | 86131 |

## 5 Conclusion

In this paper, we introduced a way to tackle the problem of model selection in online learning for time series forecasting. Unlike statistical ARMA models our algorithm AA+ARMA-OGD is not restricted to the stationary time-series.

It has a guarantee for experts and their aggregation − experimental evaluation show how this guarantee holds.

In the future, we will investigate the spectral analysis of ARMA-OGD.

## Acknowledgement

The European Commission has supported Waqas Jamil and Abdelhamid Bouchachia under the Horizon 2020 Grant 687691 related to the project: *PROTEUS: Scalable Online Machine Learning for Predictive Analytic and Real-Time Interactive Visualisation.*

## Bibliography

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*, pages 199–213. Springer, 1998.

[2] O. Anava, E. Hazan, S. Mannor, and O. Shamir. Online learning for time series prediction. In *COLT*, pages 172–184, 2013.

[3] *Daily maximum temperatures in Melbourne, Australia*. Australian Bureau of Meteorology, 2012. https://datamarket.com/data/set/2323/daily-maximum-temperatures-in-melbourne-australia-1981-1990#!ds=2323&display=line.

[4] *Daily minimum temperatures in Melbourne, Australia*. Australian Bureau of Meteorology, 2012. https://datamarket.com/data/set/2324/daily-minimum-temperatures-in-melbourne-australia-1981-1990#!ds=2324&display=line.

[5] T. V. Erven, S. D. Rooij, and P. Grünwald. Catching up faster in bayesian model selection and model averaging. In *Advances in Neural Information Processing Systems*, pages 417–424, 2007.

[6] J. D. Hamilton. *Time series analysis*, volume 2. Princeton university press Princeton, 1994.

[7] M. Herbster and M. K. Warmuth. Tracking the best expert. *Machine Learning*, 32 (2):151–178, 1998.

[8] R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2014.

[9] W. Jamil, Y. Kalnishkan, and A. Bouchachia. Aggregation algorithm vs. average for time series prediction. In *In Proceedings of the ECMLPKDD 2016 Workshop on Large-scale Learning from Data Streams in Evolving Environments STREAMEVOLV-2016, 9, 2016.*, 2016.

[10] Y.-A. Le Borgne, S. Santini, and G. Bontempi. Adaptive model selection for time series prediction in wireless sensor networks. *Signal Processing*, 87(12):3010–3020, 2007.

[11] C. Liu, S. C. Hoi, P. Zhao, and J. Sun. Online arima algorithms for time series prediction. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[12] M. Noshad, J. Ding, and V. Tarokh. Sequential learning of multi-state autoregressive time series. In *Proceedings of the 2015 Conference on research in adaptive and convergent systems*, pages 44–51. ACM, 2015.

[13] R. Prado and H. F. Lopes. Sequential parameter learning and filtering in structured autoregressive state-space models. *Statistics and Computing*, 23(1):43–57, 2013.

[14] C. Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.

[15] A. Romanenko. Aggregation of adaptive forecasting algorithms under asymmetric loss function. In *International Symposium on Statistical Learning and Data Sciences*, pages 137–146. Springer, 2015.

[16] M.-A. Sato. Online model selection based on the variational bayes. *Neural Computation*, 13(7):1649–1681, 2001.

[17] G. Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6 (2):461–464, 1978.

[18] R. Shibata. Selection of the order of an autoregressive model by akaike's information criterion. *Biometrika*, 63(1):117–126, 1976.

[19] V. Vovk. Aggregating strategies. In *Proc. Third Workshop on Computational Learning Theory*, pages 371–383. Morgan Kaufmann, 1990.

[20] V. Vovk. A game of prediction with expert advice. In *Proceedings of the eighth annual conference on Computational learning theory*, pages 51–60. ACM, 1995.

[21] V. Vovk. Competitive on-line statistics. *International Statistical Review/Revue Internationale de Statistique*, pages 213–248, 2001.

[22] V. Vovk and F. Zhdanov. Prediction with expert advice for the brier game. *Journal of Machine Learning Research*, 10(Nov):2445–2471, 2009.

[23] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. Technical Report CMU-CS-03-110, School of Computer Science, Carnegie Mellon University, 2003.