CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Bournemouth University Research Online

Aggregation Algorithm vs. Average For Time Series Prediction

Waqas Jamil[1], Yuri Kaliniskan[2], and Hamid Bouchachia[3]

jamylwaqas@gmail.com
[1,2]Department of Computer Science,
Royal Holloway, University of London. TW20 0EX, Egham, UK
Yuri.Kalnishkan@rhul.ac.uk
[1,3]Department of Computing & Informatics, Machine Intelligence Group,
Bournemouth University, BH12 5BB,Poole, UK
abouchachia@bournemouth.ac.uk

Abstract. Learning with expert advice as a scheme of on-line learning has been very successfully applied to various learning problems due to its strong theoretical basis. In this paper, for the purpose of times series prediction, we investigate the application of Aggregation Algorithm, which a generalisation of the famous weighted majority algorithm. The results of the experiments done, show that the Aggregation Algorithm performs very well in comparison to average.

Keywords: Aggregation Algorithm; time-series; auto-regressive-moving-average; auto-regressive-integrated-moving-average; Fourier transform; on-line learning.

1 Introduction

A time series is a set of repeated observations of the same variable, such as stock return or GNP. A time series consists of cyclic(for example daily fluctuations), seasonal(variation in data due to calender related effect) or irregular effects(any movement other than seasonal or cyclic). Machine Learning methods are now been used to analyse large data [Ahmed et al., 2010]. Traditionally time series auto-regressive moving average (ARMA) and auto-regressive-integrated moving average(ARIMA) models were designed to work in batch mode, rather than the on-line mode, however work on on-line ARMA models [Anava et al., 2013] and ARIMA models [Liu et al., 2016] has been done.

There exist two classes of modelling techniques for time series: statistical learning and competitive on-line learning [Anava et al., 2013], the former assumes that the observations are drawn from some unknown distribution. Representative techniques of this class include the well-known autoregressive moving average (ARMA) and its alike seen as a standard time-series modelling techniques. The motivation behind developing competative on-line learning algorithm is that the statistical(ARMA) models have strong distributional assumptions, due to which they have asymptotic guarantees [Kuznetsov and Mohri, 2016].

On-line learning has received a great attention from the machine learning community. Its origin goes back to the late 1980's and early 1990's with the advent of the paradigm of prediction with expert advice. The early work appeared in a number of seminal papers by [Haussler et al., 1994]. On-line learning consists of learning a sequentially presented set of training data upon arrival, without re-examining data that has been processed so far. In general on-line learning is practical for applications where the data set is large and cannot be processed at once due to memory constraints. Practically an on-line learner receives a new data instance, along with current hypothesis, checks if the data instance is covered by the current hypothesis and updates the hypothesis accordingly. The protocol of on-line learning can be summarized as follows: the learner receives an observation; the learner makes a decision; the learner receives the ground truth; learner incurs the loss and updates its hypothesis. The learning process is based on the minimisation of the loss (regret) which corresponds to the discrepancy between the loss and the loss of the best expert in hindsight.

Neural network is another approach used in time-series, in particular financial and economic time-series. Neural networks are universal function approximators that can map any non-linear function [White, 1989], which makes them extremely attractive when dealing with non-linearity, as linearity in ARMA imposes limits on their flexibility. A hybrid of ARIMA and neural network models has also been used [Díaz-Robles et al., 2008] in the past. The basic model is; the target variables are composed of a linear and non-linear component; it estimates linear portion using ARIMA; error term consists of non-linear relationship with previous errors, for which the neural networks are used.

Prediction models have parameters and we are faced with the problem of selecting the best set of parameters. If we have little information on the predictive behaviour of parameters, one may want to keep all models and predict using the average.

Methods of competitive prediction provide a better alternative to averaging.

The approach of this paper is drawn from the area of competitive on-line prediction, where the goal is merging predictions of experts. In this paper we merge predictions of time series models with respect to the square loss. [DeSantis et al., 1988] for the first time presented the Bayesian mixing scheme using log-loss, later [Littlestone and Warmuth, 1989] presented what was known as weighted majority algorithm and Vovk generalised them which resulted in Aggregation Algorithm(AA) [Vovk, 1992] and [Vovk, 1990]. AA has been proven in [Vovk, 1995] to be optimal in some cases.

[Box et al., 2015] were the ones who probably launched auto-regressive integrated moving-average, the Box-Jenkins methodology eloberated by [Hibon and Makridakis, 1997]. The idea was extended to state-space representation, by [Durbin, 2004]. The book by [Hyndman and Athanasopoulos, 2014] captures broad spectrum of work on time series.

2 Background

We develop an underlying system of obtaining predictions with the objective of comparing simple average against the AA.

The system we provide is a hybrid system i.e. the prediction comes from usual statistical learning approach which we briefly outline in this section along with the on-line protocol, then we use a competitive on-line learning algorithm on those predictions, which are outlined in the next section.

We highlight the usefulness of AA in time-series in a similar fashion as done in [Romanenko, 2015], the novelty of this paper is in mixing of ARIMA's. We hope the idea is extended to the on-line time-series set-up, since on-line time-series also require parameters selection.

2.1 The underlying system

The observed time series is a realization of a *stochastic processes*. A stochastic process is any collection of random variables $X_t, t \in \mathbb{T}$ defined on a common probability space Ω, here t denotes the time. So $\mathbb{T}$ can be either discrete or continuous set of series. In our case we only deal with discrete-time series [Berchtold, 1995].

If a random variable X is indexed to time, we denote time by t, the observations $\{X_t, t \in \mathbb{T}\}$, where $\mathbb{T}$ is a time indexed set, for example it may be a set of integers. The stochastic process is described by a probability distribution for $\{X_t\}$, where often elements lack independence. The distribution is usually characterized using the moments.

The objective of time-series models is to make predictions, so sometimes it is also referred as predictive inference. Time-series methods make prediction based on historical pattern of the data, measurements are taken at successive periods, such as over day, month, year etc. At the very heart of time series analysis lies ARMA models, mathematically represented as:

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{i=1}^{q} \theta_i W_{t-i} + W_t \tag{1}$$

We can divide ARMA models into two parts, auto-regressive(AR) part and the moving-average(MA) part. The challenge is often to determine the correct order p and q for AR and MA part respectively. Generally the notation used is ARMA(p,q)[1]. So AR(p) can be represented by the following equation:

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + W_t \tag{2}$$

[1] The equations used for time-series ARMA,ARIMA,AR, and MA are adopted from [Liu et al., 2016]

Similarly for MA(q) the following equation:

$$X_t = \sum_{i=1}^{q} \theta_i W_{t-i} + W_t \tag{3}$$

where X_t is stationary, $\phi \in \mathbb{R}^p$, $\theta \in \mathbb{R}^q$ parameters, and W_t is a Gaussian white noise series with mean 0 and variance σ^2. The AR models is very similar to the multiple linear regression models, except that the X_t is regressed on the past values of X_t [Liu et al., 2016], whereas MA has been used for data smoothing, without explicitly using the term *moving average*, they were described as *instantaneous averages* by [Yule, 1909]. Exponential moving average were formally applied in [Haurlan, 1968] to track stock prices. Exponential smoothing developed by [Brown, 2004] and [Holt, 2004] is a techniques regularly used now a days for data smoothing.

In practical situation often a time-series is not a realisation of a stationary process [Liu et al., 2016], so ARIMA(p, d, q) are used:

$$\nabla^d X_t = \sum_{i=1}^{p} \phi_i \nabla^d X_{t-i} + \sum_{i=1}^{q} \theta_i W_{t-i} + W_t \tag{4}$$

where $\nabla X_t = X_t - X_{t-1}$, $\phi \in \mathbb{R}^p$, $\theta \in \mathbb{R}^q$, and $W_t \backsim N(0, \sigma^2)$. It is worth noting that ARMA(p, q) is a special case of ARIMA($p, 0, q$).

Our main experiments uses data from Meteorology, so it is reasonable to assume that there is some sort of cyclic behaviour or periodicity in it, which is also justified in [Jones and Brelsford, 1967]. In order to tackle periodicity we make use of the Fourier series. A Fourier series is a specific type of infinite mathematical series involving trigonometric functions, the series were introduced by [baron Fourier, 1831]. Fourier series are used in applied mathematics, and especially in physics and electronics, to express periodic functions such as those that comprise communications signals in waveform, however Fourier series truly began with the profound work of Fourier on heat conduction at the beginning of the 19th century [Walker, 1988], Fourier proposed that initial temperatures could be represented as a series of sin functions. The discrete-time Fourier transform is a periodic function, often defined in terms of a Fourier series. [Strang, 1994] states "*The discrete fourier transform is the most important discrete transform, used to perform Fourier analysis in many practical applications*". The discrete Fourier transform takes a time-based pattern, measures every possible cycle, and returns the overall amplitude, offset, and rotation speed for every cycle that was found, [Press, 2007] provides a comprehensive guide on its application.The data is cyclic or periodic, thus we prefer using Fourier series [Bracewell, 1965] approach with ARIMA, instead of SARIMA models [Hu et al., 2007], and our model becomes:

$$Y_t = c + \sum_{k=1}^{K} \left[\alpha_k \sin\left(\frac{2\pi k t}{m}\right) + \beta_k \cos\left(\frac{2\pi k t}{m}\right) \right] \tag{5}$$

Where X_t is the ARIMA or ARMA model [2]. The value of k can be chosen to adjust the data [3], and c is some constant.

2.2 On-line protocol

We now define the notion of game-theoretic frame work, or to be more specific we define [Vovk and Zhdanov, 2009] general prediction game. Let Γ be the prediction space and Ω be the outcome space, on each trial $t \in \mathbb{Z}$. Also we assume the number of experts $n < \infty$, then each expert makes a prediction $\gamma_t \in \Gamma$; a learner observes $\langle \gamma_1^t ... \gamma_n^t \rangle$; learner makes a γ^t; nature chooses an outcome $\omega^t \in \Omega$; finally each expert and learner loss is calculated using an appropriate loss function.

The following notation is used

- outcomes are denoted as ω_1, ω_2, they happen sequentially from the outcome space $\Omega = [A, B]$, where A and B are the minimum and maximum(of a particular data for example).
- predictions at time t is represented as γ_t, and they belong to prediction space $\Gamma = \{A, B\}$.
- $\theta_1, ..., \theta_N$ experts, there are *finite* number of experts and the experts space is denoted by Θ.
- we denote learner by L and its loss at time T as $Loss_T(L) = \sum_{t=1}^{T} \lambda(\gamma_t, \omega_t)$, where γ_t and ω_t and the expert loss is $Loss_T(\theta)$.

Following are the set of assumptions or rather a scenario under we work:

- We do not assume that there is a model generating outcomes
- Outcomes may be *adversarial* meaning, for example output of 1 every-time we predict zero
- We have access to the experts predictions and expert prediction is incorporated or consulted before a prediction is given
- The objective to be as close as possible to the best expert
- Experts maybe *adversarial* too

[2] An ARIMA is just the differencing of ARMA model, if ARMA models are not stationary, but are difference stationary then we call them ARIMA, where I is the number of difference we took to make it stationary [Ghosh, 1976], but `forcast` package by Rob J Hyndman provides more flexibility.

[3] We used (5) in our experiment from the implementation in the package `forecast` by Rob J Hyndman, for details [Hyndman and Athanasopoulos, 2014]

The protocol for on-line prediction described in introduction and here can be formalised as follows:[4]

Protocol Prediction With Expert Advise

for $t = 1, 2, ...$ **do**
  $\theta \in \Theta$ predicts $\gamma_t^\theta \in \Gamma$
  Learner output $\gamma_t \in \Gamma$
  System output $\omega_t \in \Omega$
  $\theta \in \Theta$ suffers losses $\lambda(\gamma_t^\theta, \omega_t)$
  Learner loss $\lambda(\gamma_t, \omega_t)$
end for

There are different loss functions that can be used, but for the purposes of our experiment we use square-loss, the main reasons are as follows:

- If the outcomes and predictions are bounded, square loss is bounded;
- bounded square loss is mixable [Vovk, 2001].

3 Applying AA on Time Series

The weighted majority algorithm is very restrictive, so we go on a better algorithm, known as the *Aggregating Algorithm*(AA). The AA takes two parameters, a prior probability say W_0 and the learning rate $\eta > 0$. W_0 is related to the weights and the 0 in the subscript makes it the initial weights. Before we describe the AA, we explain the mechanics of the algorithm i.e. obtaining the generalised prediction, later that can be easily used for the purposes of the actual prediction. At every step t experts weight is updated, so intuitively, if an expert makes a mistake we would reduce its weight, mathematically we represent this by using our defined notation:

$$g_t(\omega) = \log_\beta \int_\Theta \beta^{\lambda(\gamma_t, \omega^\theta)} W_{t-1}^* \mathrm{d}\theta \tag{6}$$

where $\theta \in \Theta$ and $W^* = \frac{W_{t-1}(d\theta)}{W_{t-1}(\Theta)}$ in simple words W^* represent the normalised weights. For mathematical details and optimality of AA see [Vovk, 2001].

Notice in (6) we use ω not ω_t, this is because we are at the moment making prediction. What AA does is uses a *substitution function*[5] which maps the generalised prediction into Γ.

In certain situations it may not be possible to use a substitution function that perfectly maps to Γ, we call such situations *non-mixable* scenario, however

[4] The notation are adopted from [Vovk, 2001] and [Vovk and Zhdanov, 2009]

[5] The substitution function used in this experiment was $\frac{B+A}{2} + \frac{g(B)-g(A)}{2(B-A)}$, by considering square loss with $\Omega = \Gamma = [A, B]$ and $\eta \leq \frac{2}{(B-A)^2}$, by using [Haussler et al., 1998] the restriction $[-1, 1]$ can be removed and then we solve the system of equations.

in the case of square loss it is possible to find a substitution function that maps to Γ. Mathematically we say that the loss function λ is η-mixable if there is a super-prediction $\Sigma = \{(x,y)|$ there is $\gamma \in \Gamma : x \geq \lambda(\gamma,-1), y \geq \lambda(\gamma,1)\}$ meaning $C(\eta) = 1$, and we can always solve the $\lambda(\gamma,-1) \leq g_t(-1)$ and $\lambda(\gamma,-1) \leq g_t(1)$. The loss of AA can not be much larger than the best expert, for a mixable finite experts game, and uniformly initialising the prior weights of the experts.

$$Loss(AA) = Loss_{best}(\theta) + \frac{\log N}{\eta} \tag{7}$$

where $\theta \in \Theta$, η is the learning rate, and N is the number of experts. This bound (7) is shown [Vovk, 1995] to be optimal in a very strong sense i.e. it can't be improved by any other prediction algorithm.

We now formally present the pseudo-code, where parameters are $\eta > 0$ and initial distribution is of $q_1, q_2, .., q_n$. This algorithm is also for the cases of non-mixable game.

Algorithm Aggregating Algorithm

initialise weights $w_0^n = q_n, n = 1, 2, ...N$
for $t = 1, 2, ..$ **do**
notice experts prediction γ_t^n, $n = 1, 2, ..., N$
weights normalisation $p_{t-1}^n = \frac{w_{t-1}^n}{\sum_{i=1}^N w_{t-1}^i}$
solve the system $(\omega \in \Omega)\lambda(\gamma,\omega) \leq -\frac{C(\eta)}{\eta} \log \sum_{n=1}^N p_{t-1}^n e^{-\eta\lambda(\gamma_t^n,\omega)}$ w.r.t γ and output a solution γ_t
notice ω_t
experts weights update $w_t^n = w_{t-1}^n e^{-\eta\lambda(\gamma_t^n,\omega_t)}, n = 1, 2, ..., N$
end for

4 Empirical evaluation

Our experiments uses [Dai, 2012a] and [Dai, 2012b] data, we address them as maximum temperature and minimum temperature data respectively, they have 3650 days of maximum and minimum temperatures in Degrees Celsius respectively, from year 1981 to 1990. The data can be downloaded and visualised from the links provided in maximum temperature and minimum temperature data . The reason behind using this data is its periodicity (the time series model used in our experiments don't update the coefficients or the value of k or the coefficients of the ARIMA models) and its size. We performed experiments by fitting 18 combinations of ARIMA models (experts for AA) with parameters $p = 0, 1, 2$, $d = 0, 1$, $q = 0, 1, 2$ see [Pankratz, 2012], and their respective coefficients on the first 365 days, then these models were used to obtain one-step ahead forecast for each model. For the Fourier part $k = 57$ for minimum temperature data and

$k = 95$ for maximum temperature, see (6). The value of k was chosen by doing cross-validation on the first 365 days.

To be more explicit, first 365 days were used to select the coefficients for the 18 models (experts for AA) and the value of k, then at each step these models give one step ahead prediction (experts predictions for AA) based on the previous observations.

Table 1. The table shows the overall cumulative losses of each expert(model)

ARIMA(p, d, q)	Daily min temp Melbourne, Aus	Daily max temp Melbourne, Aus
$(0, 0, 0)$	4.221216×10^4	1.157103×10^5
$(1, 0, 0)$	5.574656×10^4	2.343769×10^5
$(2, 0, 0)$	9.666095×10^4	5.388992×10^5
$(0, 1, 0)$	2.809737×10^4	1.056317×10^5
$(1, 1, 0)$	3.158952×10^4	1.566821×10^5
$(2, 1, 0)$	3.687223×10^4	2.471392×10^5
$(0, 0, 1)$	9.097908×10^4	1.8425422451×10^9
$(1, 0, 1)$	8.841317×10^4	4.3844789747×10^9
$(2, 0, 1)$	2.225837×10^9	2.0250958758×10^9
$(0, 1, 1)$	4.179097×10^4	1.147874×10^5
$(1, 1, 1)$	5.497162×10^4	2.318993×10^5
$(2, 1, 1)$	9.500385×10^4	4.907601×10^5
$(0, 0, 2)$	6.314768×10^8	8.264529109×10^8
$(1, 0, 2)$	7.016491×10^8	5.034708878×10^8
$(2, 0, 2)$	9.846070×10^8	3.27074535×10^7
$(0, 1, 2)$	1.474706×10^6	1.9450601×10^6
$(1, 1, 2)$	2.267545×10^5	1.54908386×10^7
$(2, 1, 2)$	1.322986×10^6	1.7343336×10^6

Table 2. The table shows the overall cumulative losses of each average and the AA.

Square Loss	Daily min temp Melbourne, Aus	Daily max temp Melbourne, Aus
Average	82052051	77401075
AA	28048.76	102187.7

Fig 1 demonstrates that AA follows the best expert, which in both (A and C for minimum temperature data) experiments is ARIMA$(0, 1, 0)$ with respective values of k. In our experiments overall loss of AA is even less than the best expert. This is because the best expert gives the lowest overall loss, however it does not give lower loss from the start, there are other experts who are able to compete with this expert and some outperform the best expert, all of this AA captures, and is able to outperform the best expert.

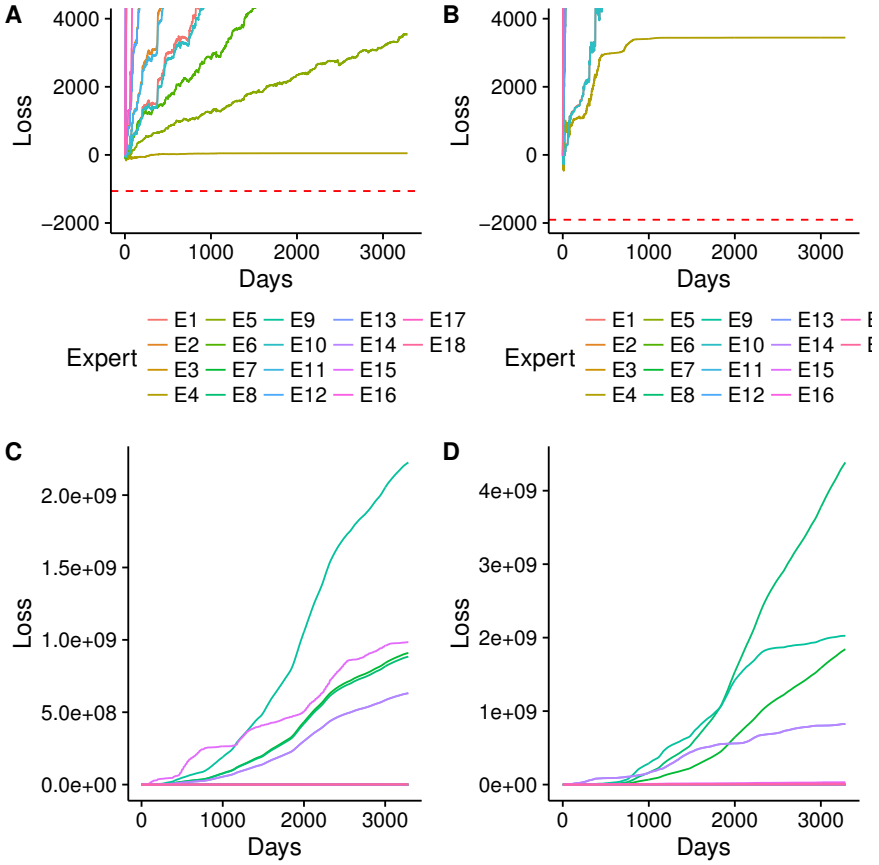

Fig. 1. The plot A and B shows that the lower bound $\left(\text{red dashed line} = -\frac{\log(18)}{\eta}\right)$, -1061.35 and -1904.307 respectively has not been violated by AA, whereas plot C and D show an overall behaviour.

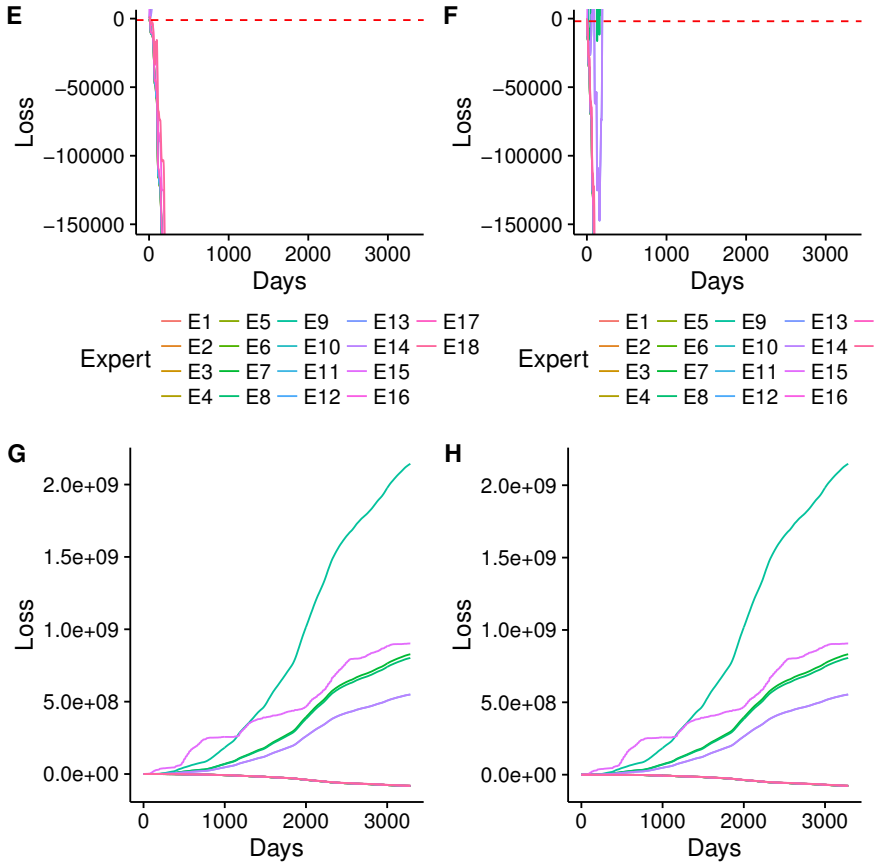

Fig. 2. The plot E and F shows that the lower bound $\left(\text{red dashed line} = -\frac{\log(18)}{\eta}\right)$, -1061.35 and -1904.307 respectively is violated by the average.

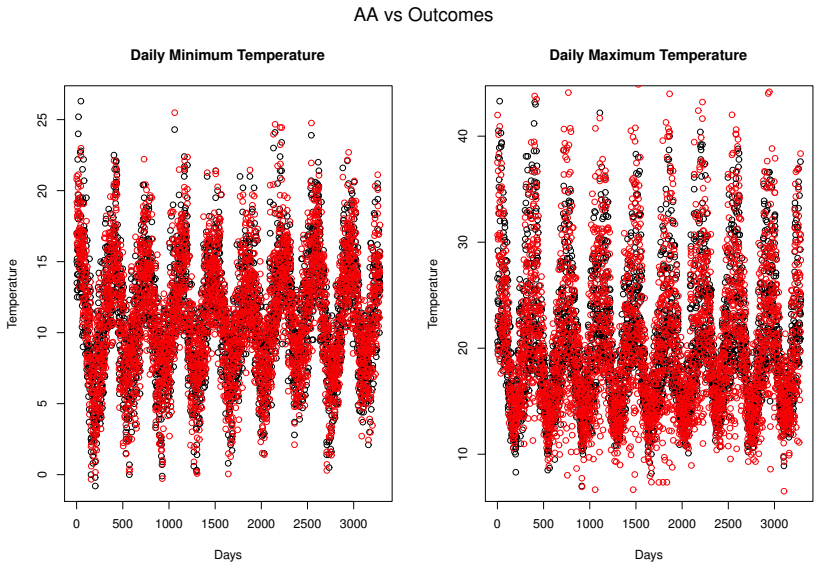

Fig. 3. The plot shows how well the AA(red) fits the actual outcomes(black). It is the extremes which AA is not predicting well, it fits well the points between the extremes.

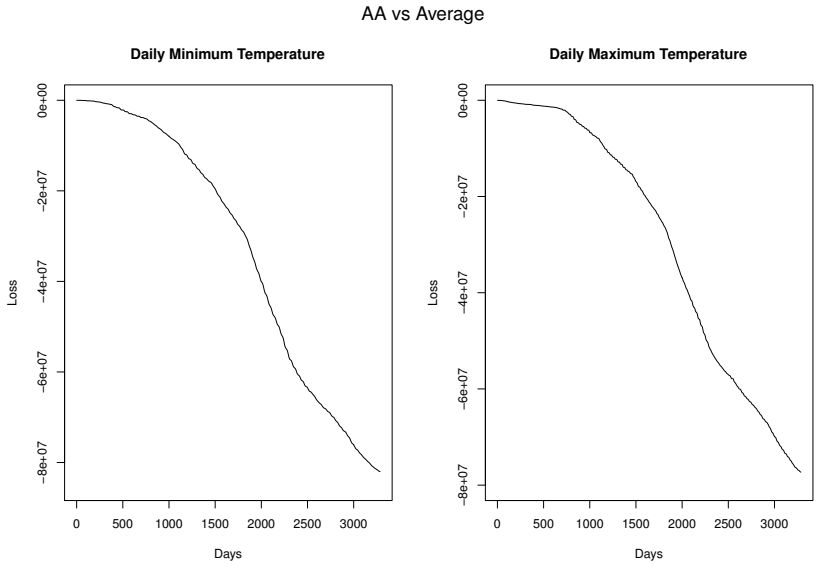

Fig. 4. The decreasing behaviour of the difference between square losses of AA and average suggest that AA outperforms the average.

One can compare Fig 2 (E and G for minimum temperature data) with Fig 1 and easily see averaging is not favourable, since the spread between the expert with maximum loss and the expert with minimum loss is colossal. Fig 3 shows that AA actually does very well in predicting the outcomes, only the extreme temperatures are missed mainly.

Fig 4 is the plot demonstrating the difference between the AA and the average (AA minus average) square loss, since average is greater than the AA, hence the decreasing behaviour in the curve.

5 Conclusion

We have studied the performance of AA, and average on minimum temperature and maximum temperature data . We conclude that, best expert ARIMA$(0, 1, 0)$ with $k = 95$ for maximum temperature data and $k = 54$ for minimum temperature data are outperformed by AA prediction, furthermore AA is significantly better than taking simple average as shown in Table 2 (the square loss of AA is substantially lower then the average) Fig 1 clearly show that the theoretical bound by AA has not been violated, due to which AA gives a better result than the simple average. Experts predictions are used by AA to produce a better overall result.

Acknowledgements

Waqas Jamil and Hamid Bouchachia has been supported by the European Commission under the Horizon 2020 Grant 687691 related to the project: *PROTEUS: Scalable Online Machine Learning for Predictive Analytics and Real-Time Interactive Visualization.*

Yuri Kalnishkan has been supported by the Leverhulme Trust through the grant RPG-2013-047 'Online self-tuning learning algorithms for handling historical information'.

References

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

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

Ahmed et al., 2010. Ahmed, N. K., Atiya, A. F., Gayar, N. E., and El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5-6):594–621.

Anava et al., 2013. Anava, O., Hazan, E., Mannor, S., and Shamir, O. (2013). Online learning for time series prediction. In *COLT*, pages 172–184.

baron Fourier, 1831. baron Fourier, J. B. J. (1831). *Analyse des équations déterminées*, volume 1. Firmin Didot.

Berchtold, 1995. Berchtold, A. (1995). Autoregressive modelling of markov chains. In *Statistical Modelling*, pages 19–26. Springer.

Box et al., 2015. Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Bracewell, 1965. Bracewell, R. (1965). The fourier transform and its applications. *New York*, 5.

Brown, 2004. Brown, R. G. (2004). *Smoothing, forecasting and prediction of discrete time series*. Courier Corporation.

DeSantis et al., 1988. DeSantis, A., Markowsky, G., and Wegman, M. N. (1988). Learning probabilistic prediction functions. In *Foundations of Computer Science, 1988., 29th Annual Symposium on*, pages 110–119. IEEE.

Díaz-Robles et al., 2008. Díaz-Robles, L. A., Ortega, J. C., Fu, J. S., Reed, G. D., Chow, J. C., Watson, J. G., and Moncada-Herrera, J. A. (2008). A hybrid arima and artificial neural networks model to forecast particulate matter in urban areas: The case of temuco, chile. *Atmospheric Environment*, 42(35):8331–8340.

Durbin, 2004. Durbin, J. (2004). Introduction to state space time series analysis. *State Space and unobserved component models: Theory and Applications*, pages 3–25.

Ghosh, 1976. Ghosh, N. (1976). Time series analysis and forecasting (the box-jenkins approach). *Journal of the Operational Research Society*, 27(3):644–644.

Haurlan, 1968. Haurlan, P. (1968). *Measuring Trend Values*. Trade Levels.

Haussler et al., 1998. Haussler, D., Kivinen, J., and Warmuth, M. K. (1998). Sequential prediction of individual sequences under general loss functions. *IEEE Transactions on Information Theory*, 44(5):1906–1925.

Haussler et al., 1994. Haussler, D., Littlestone, N., and Warmuth, M. K. (1994). Predicting {0, 1}-functions on randomly drawn points. *Information and Computation*, 115(2):248–292.

Hibon and Makridakis, 1997. Hibon, M. and Makridakis, S. (1997). Arma models and the box–jenkins methodology.

Holt, 2004. Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, 20(1):5–10.

Hu et al., 2007. Hu, W., Tong, S., Mengersen, K., and Connell, D. (2007). Weather variability and the incidence of cryptosporidiosis: comparison of time series poisson regression and sarima models. *Annals of epidemiology*, 17(9):679–688.

Hyndman and Athanasopoulos, 2014. Hyndman, R. J. and Athanasopoulos, G. (2014). *Forecasting: principles and practice*. OTexts.

Jones and Brelsford, 1967. Jones, R. H. and Brelsford, W. M. (1967). Time series with periodic structure. *Biometrika*, 54(3-4):403–408.

Kuznetsov and Mohri, 2016. Kuznetsov, V. and Mohri, M. (2016). Time series prediction and online learning. In *29th Annual Conference on Learning Theory*, pages 1190–1213.

Littlestone and Warmuth, 1989. Littlestone, N. and Warmuth, M. K. (1989). The weighted majority algorithm. In *Foundations of Computer Science, 1989., 30th Annual Symposium on*, pages 256–261. IEEE.

Liu et al., 2016. Liu, C., Hoi, S. C., Zhao, P., and Sun, J. (2016). Online arima algorithms for time series prediction. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Pankratz, 2012. Pankratz, A. (2012). *Forecasting with dynamic regression models*, volume 935. John Wiley & Sons.

Press, 2007. Press, W. H. (2007). *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press.

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

Strang, 1994. Strang, G. (1994). Wavelets. *American Scientist*, 82(3):250–255.

Vovk, 1992. Vovk, V. (1992). Universal forecasting algorithms. *Information and Computation*, 96(2):245–277.

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

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

Vovk, 1990. Vovk, V. G. (1990). Aggregating strategies. In *Proc. Third Workshop on Computational Learning Theory*, pages 371–383. Morgan Kaufmann.

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

Walker, 1988. Walker, J. S. (1988). *Fourier analysis*. Oxford University Press.

White, 1989. White, H. (1989). Learning in artificial neural networks: A statistical perspective. *Neural computation*, 1(4):425–464.

Yule, 1909. Yule, G. U. (1909). The applications of the method of correlation to social and economic statistics. *Journal of the Royal Statistical Society*, 72(4):721–730.