Influence of hyperparameters on online aggregation with countable experts

S. M. Kunin-Bogoiavlenskii, A. V. Zukhba
kunin-bogoiavlenskii.sm@phystech.edu; a__l@mail.ru
Moscow Insitute of Physics and Technology

Aggregating forecasts from multiple experts is a valuable method to improve prediction accuracy. This paper examines the influence of hyperparameters on the accuracy of the aggregation algorithm for a countable number of experts. We implement a time series generator with specified properties and an aggregating forecasting model. We conduct a series of experiments with various hyperparameters of the algorithm (including initialization weights, train window). The experiments confirm that these hyperparameters have a significant influence on the algorithm's performance.

Keywords: *online learning; aggregating algorithm; prediction with experts' advice; Fixed Share, Mixing Past Posteriors (MPP)*

1 Introduction

This work is inspired by the algorithm, developed in the article [4], considering online game of prediction with experts' advice. The data is presented as a time series, consisting of outcome pairs — «signal» and «response». In contrast to the classical statistical theory of sequential prediction, we make no assumptions about the nature of the data (which could be deterministic, stochastic, etc.). We use machine learning methods to build forecasters within a game-theoretic approach. The online learning master model considers a series of reference forecasters, referred to as experts, to build its opinion by combining their predictions.

The general prediction algorithm with expert advice follows this structure: Learning progresses in trials at discrete points in time. During each step , expert models, based on past observational data subsamples, provide their predictions. The master model then makes a decision using the chosen aggregating algorithm. At the end of the trial, the generator presents the true outcome, and both the master and expert models are scored using a loss function. The difference between the master's cumulative losses and the expert's cumulative losses is defined as regret. The traditional goal of the aggregating algorithm is to keep it as small as possible.

We use special assumptions about the data generation structure when building forecasting strategies. It is assumed that there are multiple generators, whose structure is unknown to the predictors. The time series is obtained by merging segments, each produced by one of the generators. These segments are called areas of stationarity, and can be studied using machine learning methods. Each corresponding local predictive model will be constructed based on data from the area of stationarity and can be then successfully applied in other areas of stationarity generated by the same generator.

In this formulation of the forecasting problem, the series of prediction steps is divided into segments that frame arbitrary sequences of expert strategies. The sequence of segments and its associated sequence of experts is called a partition. The modified goal of the aggregating algorithm is to perform well relative to the best partition. Accordingly, the new concept of algorithm regret is the difference between the algorithm's losses and the cumulative losses of the sequence of experts. This change allows for a more accurate modeling of real-life conditions, where the nature of responses may change over time and different experts may predict with varying degrees of success depending on the current trend.

The corresponding algorithm is called Fixed Share [5]. A further proposed generalization of it is the Mixing Past Posteriors (MPP) method [6]. The cumulative losses of the aggregating algorithm are related to convex combinations of expert losses. The concept of regret also changes. Now the algorithm's cumulative losses are compared to the cumulative losses of convex combinations of expert strategies.

A characteristic feature of the problem considered in [4] is the absence of a predefined set of competing expert strategies, as was the case in the works cited above. Instead, new expert strategies are constructed at each step of the online learning process. The master must aggregate the forecasts of all expert strategies constructed up to that time in real-time at each step. Algorithm GMPP, proposed in [4], is the foundation of our experiments.

2 Problem statement

Let $x_1, x_2, \dots$ be the sequence of signals belonging to the sample space $\mathcal{X}$. The masters goal is to predict the sequence of the corresponding responces $y_1, y_2, \dots$ belonging to the outcome space $\mathcal{Y}$. We assume that the master knows $a, b \in \mathbb{R}$ — the bounds of the responces sequence, s.t. $\forall i \quad a \leqslant y_i \leqslant b$.

We assume that there is an countable number of experts $i \in \mathcal{N}$, where $\mathcal{N}$ is the natural numbers set. The predictions of the master and experts belong to the decision space $\mathcal{D}$. Let $\lambda : \mathcal{D} \times \mathcal{Y} \to \mathbb{R}_+$ be the nonnegative loss function.

At each step t every expert $i \in \mathcal{N}$ provides his prediction $f_t^i = f_t^i(x_t) \in \mathcal{D}$. After obtaining them the master gives his prediction $p_t = p_t(x_t) \in \mathcal{D}$. Next, the generator reveals the true outcome y_t, and the losses are computed. Let $l_t^i = \lambda(f_t^i, y_t)$ be the loss of expert i, and $h_t = \lambda(p_t, y_t)$ be the master's loss

With designations $L_T^i = \sum_{t=1}^T l_t^i$, representing the cumulative loss of expert i during the first T steps, and $H_T = \sum_{t=1}^T h_t$, representing the master's cumulative loss during the first T steps, we can define the master's regret relative to the expert i as $R_t^i = H_T - L_T^i$.

Algorithm GMPP

Initialize the weights w_1^i so that $\sum\limits_{i \in \mathcal{N}} w_1^i = 1$

for $t = 1, 2, \dots$ **do**

- Expert f^t initialization
- Signal x_t is received
- Experts' predictions $f_t^i = f_t^i(x_t), 1 \leqslant i \leqslant t$
- Computation of normalized weights of experts $1 \leqslant i \leqslant t$:

$$\widehat{w}_t^i = \frac{w_t^i}{\sum_{j=1}^t w_t^j}$$

- Master's prediction evaluation $\gamma_t = \boldsymbol{Subst}(\mathbf{f_t}, \widehat{\mathbf{w}}_\mathbf{t}) = \dfrac{a+b}{2} + \dfrac{1}{2\eta(b-a)} \cdot \ln \dfrac{\sum\limits_{i \in \mathcal{N}} p_i e^{-\eta(b-f_i)^2}}{\sum\limits_{i \in \mathcal{N}} p_i e^{-\eta(a-f_i)^2}}$,

where $\widehat{\mathbf{w}}_\mathbf{t} = (\widehat{w}_t^1, \widehat{w}_t^2, \dots \widehat{w}_t^t)$, $\mathbf{f_t} = (f_t^1, f_t^2, \dots f_t^t)$

- True outcome y_t is revealed
- Computation of master's loss $h_t = \lambda(p_t, y_t)$ and experts' losses: $l_t^i = \begin{cases} \lambda(f_t^i, y_t), & \text{if } i \leqslant t \\ h_t, & \text{if } i > t \end{cases}$

· **Loss Update** weights modification

$$\widetilde{w}_t^i = \frac{w_t^i e^{\eta l_t^i}}{\sum\limits_{j=1}^{t} w_t^j e^{-\eta l_t^i} + e^{-\eta h_t}(1 - \sum\limits_{j=1}^{t} w_t^j)}$$

· **Mixing Update** weights modification

$$\widetilde{w}_{t+1}^i = \alpha_t \widetilde{w}_1^i + (1 - \alpha_t \widetilde{w}_t^i)$$

end for

3 Experiment

We implement the synthetic time series generator and the GMPP algorithm itself. We then conduct experiments to analyze the effects of varying hyperparameters.

3.1 Data generation

We define generators $G_1, G_2, \ldots, G_k$ by their weight vectors $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_k \in \mathbb{R}^d$. Here, d represents the dimensionality of the signals.

We generate signals as a series of vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \sim \mathcal{N}(0, I_d)$, indicating they are drawn from a zero-mean Gaussian distribution with an identity covariance matrix .

This series of vectors is then randomly divided into a sequence of segments. We denote these segments as $S_1, S_2, \ldots, S_m$. Each segment S_i is a sequence of vectors itself, containing elements from $\boldsymbol{x}_1$ to $\boldsymbol{x}_{s_i}$ (where s_i is the index of the last element in segment i). Importantly, each segment S_i is assigned to a specific generator $G_{g(i)}$ according to a random function g. This function determines which generator is responsible for creating responses for that segment.

Ultimately, we obtain the series of responses $y_1, y_2, \ldots, y_n$ by taking the dot product of each signal vector with the weight vector of the corresponding generator and adding the noise: $y_i = \langle \boldsymbol{w}_{g(i)}, \boldsymbol{x}_i \rangle + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}(0, 1)$

3.2 Experts

Each expert f_t is initialized as a linear regression model. This model is trained on a fixed window of l past observations. The training process yields a weight vector, which is calculated as $\theta_t = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y}$, where $\boldsymbol{X}^T = (\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \ldots \boldsymbol{x}_{t-l}),\ \boldsymbol{y} = (y_{t-1}, y_{t-2}, \ldots y_{t-l})$.

Given a new signal vector $\boldsymbol{x}$, the expert's prediction is obtained by calculating the dot product between the weight vector and the signal vector: $f_t(\boldsymbol{x}) = \langle \boldsymbol{\theta}_t, \boldsymbol{x} \rangle$

The «ideal expert» is formed based on a hindsight analysis of the experiment results, assuming we know the segment boundaries. For each segment, the ideal expert's predictions are set equal to the predictions of the best local expert (the one that has the lowest cumulative sum on that segment).

3.3 Fixed parameters

Signals dimention $d = 10$
Time series length $T = 2000$.
Responses borders $(a, b) = (-10, 10)$
Number of experts $w = 3$

3.4 Hyperparameters

We run algorithm with various hyperparameters. As a metric of quality we use master's regret relative to the «ideal expert».

Initialization weights

Default weights: $w_1^t = \frac{1}{(t+1)\ln^2(t+1)}$
Experimental: $\frac{1}{t^\alpha}$, $\frac{1}{(t+1)\ln(t+1)\ln^2\ln(t+c)}$, $\frac{1}{e^t}$, etc.

Mixing update coefficients

Default coefficient: $\alpha_t = \frac{1}{t+1}$
Experimental: $\frac{1}{(t+1)^\beta}$, $\frac{1}{t+c}$, etc.

Window size

We vary window size to check algorithm performance in situations when experts lack information or when they often train on pieces made by different generators

4 Results

Weight function	Regret
$1/((t+1)\ln^2(t+1))$	136360.61
$1/t^{1.01}$	94758.81
$1/t^{1.1}$	98359.26
$1/t^2$	113298.39
$1/((x+4)*\ln(x+4)*\ln^2\ln(x+4))$	127634.11

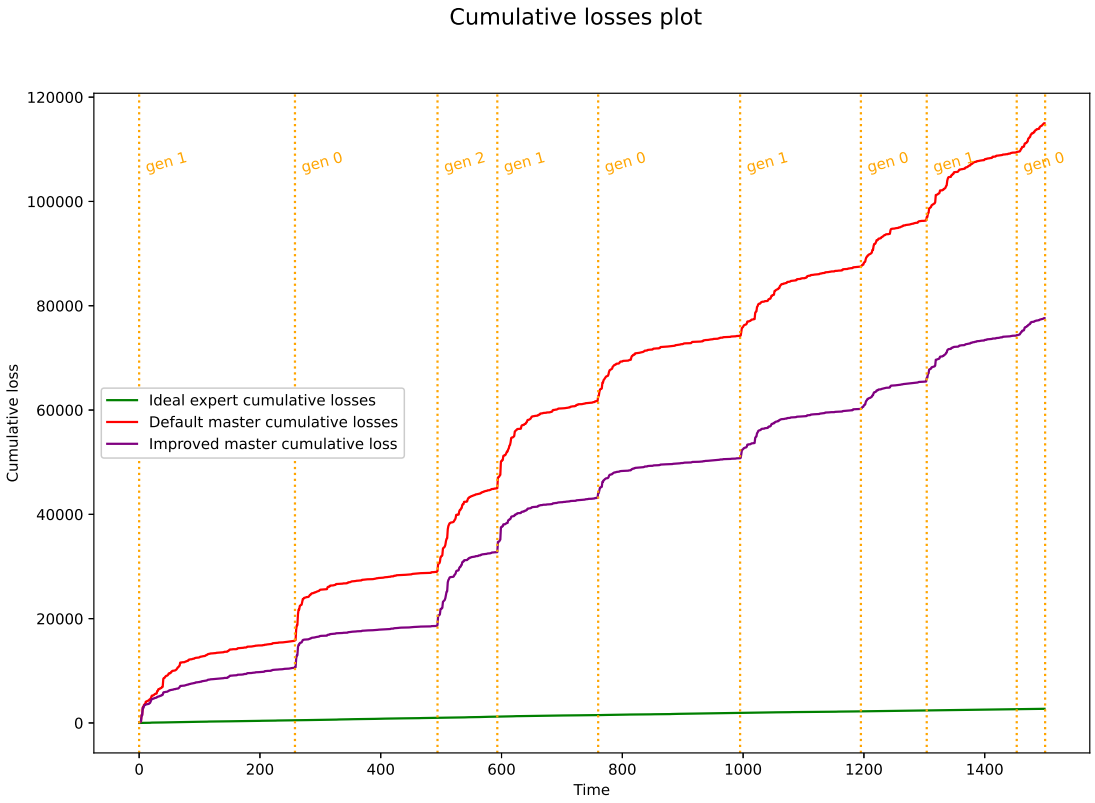
Table 1 Dinamics of regret value over different sizes of train window

	Weight function	
Window size	$\frac{1}{(t+1)\ln^2(t+1)}$	$\frac{1}{t^{1.01}}$
5	188586.09	141788.84
10	136360.61	94758.81
20	129543.68	93456.18
50	143610.19	97600.05
100	151434.18	101551.31
200	164746.91	113835.86

Table 2 Regret value for different weight functions

Alpha function	regret
$1/(x+1)$	94758.81
$1/(x+1)^{0.5}$	149472.52
$1/(x+1)^{1.5}$	92351.58
$1/(x+1)^2)$	99120.36
$1/e^{x/3}$	99477.13
$1/(x+10)$	94729.22
$1/(x+100)$	94441.30
$1/(x+1000)$	92194.36

Table 3 Regret value for different alpha functions



5 Conclusion

References

[1] N. Cesa-Bianchi, G. Lugosi. 2006. *Prediction, Learning, and Games*. Available at: `https://ii.uni.wroc.pl/~lukstafi/pmwiki/uploads/AGT/Prediction_Learning_and_Games.pdf`

[2] Hyndman, R. J. & Athanasopoulos, G., 2nd edition. 2018. *Forecasting: Principles and Practice. OTexts: Melbourne, Australia* . Available at: `https://otexts.com/fpp2/`

[3] V. V'yugin. 2022. Matematicheskie osnovy mashinnogo obucheniya i prognozirovaniya [Mathematical Foundations of Machine Learning and Forecasting]. Available at: `http://iitp.ru/upload/publications/6256/vyugin1.pdf`

[4] V. V'yugin, V. Trunov. 2023. Prognozirovanie lokal'no statsionarnykh dannykh s ispol'zovaniem predskazanii ekspertnykh strategiy [Prediction of Locally Stationary Data Using Prediction with Expert Advice]. Available at: http://www.jip.ru/2023/470-487-2023.pdf

[5] M. Herbster, M. Warmuth. 1998. *Tracking the best expert*. Available at: https://link.springer.com/content/pdf/10.1023/A:1007424614876.pdf

[6] O. Bousquet, M. Warmuth. 2002. *Tracking a small set of experts by mixing past posteriors*. Available at: https://www.jmlr.org/papers/volume3/bousquet02b/bousquet02b.pdf