

Aggregating Strategies for Long-term Forecasting

Alexander Korotin

A.KOROTIN@SKOLTECH.RU

*Skolkovo Institute of Science and Technology,
Nobel street, 3, Moscow, Moskovskaya oblast', Russia*

Vladimir V'yugin*

VYUGIN@IITP.RU

*Institute for Information Transmission Problems,
Bolshoy Karetny per. 19, build.1, Moscow, Russia*

Evgeny Burnaev†

E.BURNAEV@SKOLTECH.RU

*Skolkovo Institute of Science and Technology,
Nobel street, 3, Moscow, Moskovskaya oblast', Russia*

Editors: Alex Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgeni Smirnov and Ralf Peeters

Abstract

The article is devoted to investigating an application of aggregating algorithms to the problem of the long-term forecasting. We examine the classic aggregating algorithms based on the exponential reweighing. For the general Vovk's aggregating algorithm we provide its probabilistic interpretation and its generalization for the long-term forecasting. For the special basic case of Vovk's algorithm we provide two its modifications for the long-term forecasting. The first one is theoretically close to an optimal algorithm and is based on replication of independent copies. It provides the time-independent regret bound with respect to the best expert in the pool. The second one is not optimal but is more practical (explicitly models dependencies in observations) and has $O(\sqrt{T})$ regret bound, where T is the length of the game.

Keywords: aggregating algorithm, long-term forecasting, prediction with experts' advice, delayed feedback.

1. Introduction

We consider the online game of prediction with experts' advice. A master (aggregating) algorithm at every step $t = 1, \dots, T$ of the game has to combine aggregated prediction from predictions of a finite pool of N experts (see e.g. Littlestone and Warmuth (1994), Freund and Schapire (1997), Vovk (1990), Vovk (1998), Cesa-Bianchi and Lugosi (2006), Adamskiy et al. (2016) among others). We investigate the adversarial case, that is, no assumptions are made about the nature of the data (stochastic, deterministic, etc.).

In the classical online scenario, all predictions at step t are made for the next step $t + 1$. The true outcome is revealed immediately at the beginning of the next step of the game and the algorithm suffers loss using a loss function.

* Vladimir V'yugin was supported by the Russian Foundation for Basic Research grants 16-01-00576 and 16-29-09-649 ofi.m.

† Evgeny Burnaev was supported by the Ministry of Education and Science of Russian Federation, grant No. 14.606.21.0004, grant code: RFMEFI60617X0004.

In contrast to the classical scenario, we consider the long-term forecasting. At each step t of the game, the predictions are made for some pre-determined point $t + D$ ahead (where $D \geq 1$ is some fixed known horizon), and the true outcome is revealed only at step $t + D$.

The performance of the aggregating algorithm is measured by the regret over the entire game. The regret R_T is the difference between the cumulative loss of the online aggregating algorithm and the loss of some given comparator. A typical comparator is the best fixed expert in the pool or the best fixed convex linear combination of experts. The goal of an aggregating algorithm is to minimize the regret, that is, $R_T \rightarrow \min$.

It turns out that there exists a wide range of aggregating algorithms for the classic scenario ($D = 1$). The majority of them are based on the exponential reweighing methods (see [Littlestone and Warmuth \(1994\)](#), [Freund and Schapire \(1997\)](#), [Vovk \(1990\)](#), [Vovk \(1998\)](#), [Cesa-Bianchi and Lugosi \(2006\)](#), [Adamskiy et al. \(2016\)](#), etc.). At the same time, several algorithms come from the general theory of online convex optimization by [Hazan \(2016\)](#). Such algorithms are based on online gradient descent methods.

There is no right answer to the question which category of algorithms is better in practice. Algorithms from both groups have good theoretical performance. Regret (with respect to some given comparator) is usually bounded by a sublinear function of T :

1. $R_T \leq O(T)$ for Fixed Share with constant share by [Herbster and Warmuth \(1998\)](#);
2. $R_T \leq O(\sqrt{T})$ for Regularized Follow The Leader according to [Hazan \(2016\)](#);
3. $R_T \leq O(\ln T)$ for Fixed Share with decreasing share by [Adamskiy et al. \(2016\)](#);
4. $R_T \leq O(1)$ in aggregating algorithm by [Vovk \(1998\)](#);

and so on. In fact, the applicability of every particular algorithm and the regret bound depends on the properties of the loss function (convexity, Lipschitz, exponential concavity, mixability, etc.).

When it comes to long-term forecasting, many of these algorithms do not have theoretical guarantees of performance or even do not have a version for the long-term forecasting. Long-term forecasting implies delayed feedback. Thus, the problem of modifying the algorithms for the long-term forecasting can be partially solved by the general results of the theory of forecasting with the delayed feedback.

The main idea in the field of the forecasting with the delayed feedback belongs to [Weinberger and Ordentlich \(2002\)](#). They studied the simple case of binary sequences prediction under fixed known delay feedback D . According to their results, an optimal¹ predictor $p_D^*(x_{t+D}|x_t, \dots, x_1)$ for the delay D can be obtained from an optimal predictor $p_1^*(x_{t+1}|x_t, \dots, x_1)$ for the delay 1. The method implies running D independent copies of predictor p_1^* on D disjoint time grids $GR_d = \{t \mid t \equiv d \pmod{D}\}$ for $1 \leq d \leq D$. Thus,

$$p_D^*(x_{t+D}|x_t, \dots, x_1) = p_1^*(x_{t+D}|x_t, x_{t-D}, x_{t-2D}, \dots)$$

for all t . We illustrate the optimal partition of the timeline in Figure 1.

It turns out, their result also works for the general problem of forecasting under the fixed known delay feedback, in particular, for the prediction with expert advice (we prove

1. An optimal predictor is any predictor with the regret which is less or equal to the minimax regret. There may exist more than one optimal predictor.

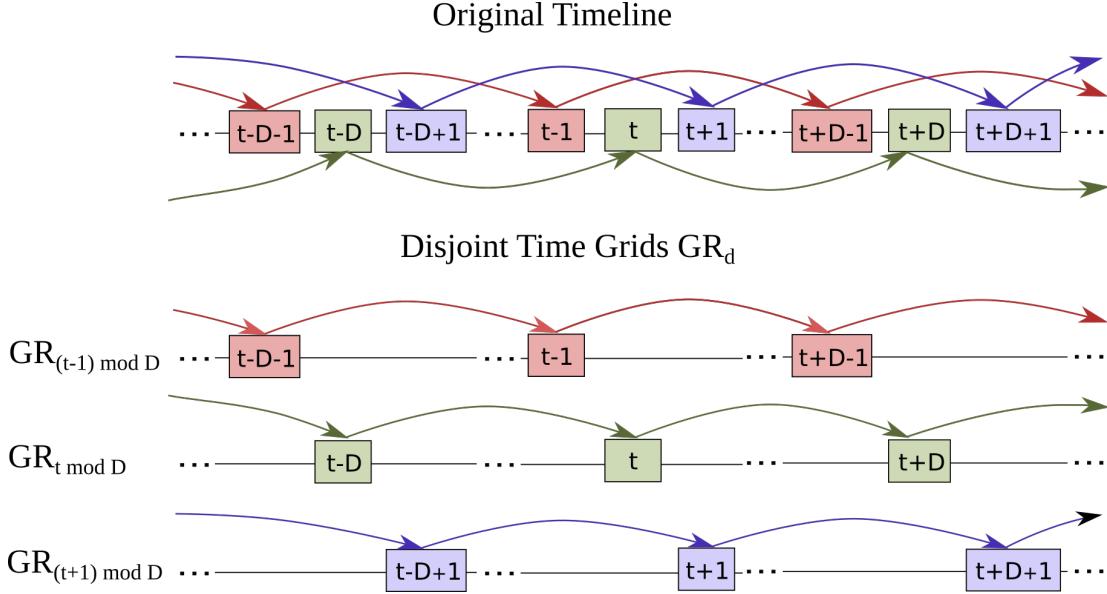


Figure 1: The optimal approach to the problem of forecasting with the fixed known delay D . The timeline is partitioned into D disjoint grids GR_d . Games on different grids are considered separately. Each game has fixed known delay 1 (not D).

this in Appendix A). Thus, it is easy to apply any 1-step-ahead forecasting aggregating algorithm to the problem of long-term forecasting by running its D independent copies on D disjoint grids GR_d (for $d = 1, \dots, D$). We call algorithms obtained by this method as *replicated algorithms*.

Nevertheless, one may say that such a theoretically optimal approach is practically far from optimal because it uses only $\frac{1}{D}$ of observed data at every step of the game. Moreover, separate learning processes on grids GR_d (for $d = 1, \dots, D$) do not even interact.

Gradient-descent-based aggregating algorithms have several non-replicated adaptations for the long-term forecasting. The most obvious adaptation is the delayed gradient descent by Quanrud and Khashabi (2015). Also, the problem of prediction with experts' advice can be considered as a special case of online convex optimization with memory by Anava et al. (2015). Both approaches provide $R_T \leq O(\sqrt{T})$ classical regret bound.² Thus, the practical and theoretical problem of modifying the gradient descent based aggregating algorithms for long-term forecasting can be considered as solved.

In this work, we investigate the problem of modifying aggregating algorithms based on exponential reweighing for the long-term forecasting. We consider the general aggregating algorithm by Vovk (1999) for the 1-step-ahead forecasting and provide its reasonable non-replicated generalization for the D -th-step-ahead forecasting. These algorithms are denoted by G_1 and G_D respectively. We obtain a general expression for the regret bound of G_D .

2. We do not include the value of the forecasting horizon D in regret bounds because in this article we are interested only in the regret asymptotic behavior w.r.t. T . In all algorithms that we discuss the asymptotic behavior w.r.t. D is sublinear or linear.

As an important special case, we consider the classical exponentially reweighing algorithm V_1 by [Vovk \(1998\)](#), that is a case of G_1 , designed to compete with the best expert in the pool. The algorithm V_1 can be considered close to an optimal one because it provides constant T -independent regret bound. We provide its replicated modification V_D for the long-term forecasting. We also propose a non-replicated modification V_D^{FC} for the long-term forecasting of V_1 (motivated by reasonable practical approach).³ Our main result here is that the regret bound for V_D^{FC} is $O(\sqrt{T})$.

All the algorithms that we develop and investigate require the loss function to be exponentially concave. This is a common assumption (see e.g. [Kivinen and Warmuth \(1999\)](#)) for the algorithms based on the exponential reweighing.⁴

The main contributions of this article are the following:

1. Developing the general non-replicated exponentially reweighing aggregating algorithm G_D for the problem of long-term prediction with experts' advice and estimating its regret.
2. Developing the non-replicated adaptation V_D^{FC} of the powerful aggregating algorithm V_1 by [Vovk \(1998\)](#). The obtained algorithm has $O(\sqrt{T})$ regret bound with respect to the best expert in the pool.

In our previous work (see [Korotin et al. \(2017\)](#)) we also studied the application of algorithm V_1 to the long-term forecasting. We applied the method of Mixing Past Posteriors by [Bousquet and Warmuth \(2003\)](#) to connect the independent learning processes on separate grids GR_d (for $d = 1, \dots, D$). We obtained the algorithm V_D^{GC} that *partially connects* the learning processes on these grids.⁵

In contrast to our previous work, in this article we consider the general probabilistic framework for the long-term forecasting (algorithms G_1 and G_D). We obtain the algorithm V_D^{FC} that *fully connects* the learning processes on different grids, see details in Subsection 4.3.

The article is structured as follows.

In Section 2 we set up the problem of long-term prediction with experts' advice and state the protocol of the online game.

In Section 3 we discuss the aggregating algorithms for the 1-step-ahead forecasting. In Subsection 3.1 we describe the general model G_1 by [Vovk \(1999\)](#) and consider its special case V_1 in Subsection 3.2.

In Section 4 we discuss aggregating algorithms for the D -th-step-ahead forecasting: we develop general model G_D in Subsection 4.1. Then we discuss its two special cases: algorithm V_D in Subsection 4.2 that is a replicated version of V_1 and our non-replicated version V_D^{FC} in Subsection 4.3. We prove the $O(\sqrt{T})$ regret bound for V_D^{FC} .

In Appendix A we generalize the result by [Weinberger and Ordentlich \(2002\)](#) to the case of long-term prediction with experts' advice and prove that the approach with replicating 1-step-ahead predictors for D -th-step-ahead forecasting is optimal.

3. FC — full connection.

4. Usually, even more general assumption is used that the loss function is mixable.

5. GC — grid connection.

2. Preliminaries

We use **bold** font to denote vectors (e.g. $\mathbf{w} \in \mathbb{R}^M$ for some integer M). In most cases, superscript refers to index/coordinate of an element in the vector (e.g. $(w^1, \dots, w^N) = \mathbf{w}$). Subscript is always used to indicate time (e.g. $l_t, R_T, \omega_\tau, w_t^n$, etc.).

For any integer M we denote the probability simplex of dimension M by

$$\Delta_M = \{\mathbf{p} \text{ such that } (\mathbf{p} \in \mathbb{R}_+^M) \wedge (\|\mathbf{p}\|_1 = 1) \wedge (\mathbf{p} > 0)\}.$$

We use the notation $\mathbf{e} \in \mathbb{R}_+^M$ to denote the unit vector $(1, 1, \dots, 1)$. The dimension M of the vector is always clear from the context. Note that $\frac{\mathbf{e}}{M} \in \Delta_M$.

The words **prediction** and **forecasting** are absolute synonyms in this paper.

2.1. A Game of Long-Term Forecasting with Experts' Advice

We consider the online game of D -th-step-ahead forecasting of time series $\omega_t \in \Omega$ by aggregating a finite pool of N forecasting experts. We use $\mathcal{N} = \{1, \dots, N\}$ to denote the pool and $n \in \mathcal{N}$ as an index of an expert.

At each integer step $t = 1, 2, \dots, T - D$ experts $n \in \mathcal{N}$ present their forecasts $\xi_{t+D}^n \in \Xi$ of time series $\{\omega_\tau\}_{\tau=1}^T$ for the time moment $t+D$. The master (aggregating) algorithm combines these forecasts into a single (aggregated) forecast $\gamma_{t+D} \in \Gamma \subset \Xi$ for the time moment $t+D$.

After the corresponding outcome ω_{t+D} is revealed (on the step $t+D$ of the game), both experts and algorithm suffer losses using a loss function $\lambda : \Omega \times \Xi \rightarrow \mathbb{R}_+$. We denote the loss of expert $n \in \mathcal{N}$ on step $t+D$ by $l_{t+D}^n = \lambda(\omega_{t+D}, \xi_{t+D}^n)$ and the loss of the aggregating algorithm by $h_{t+D} = \lambda(\omega_{t+D}, \gamma_{t+D})$. We fix the protocol of the game below.

Protocol (D -th-step-ahead forecasting with Experts' advice)

Get the experts $n \in \mathcal{N}$ predictions $\xi_t^n \in \Xi$ for steps $t = 1, \dots, D$.

Compute the aggregated predictions $\gamma_t \in \Gamma$ for steps $t = 1, \dots, D$.

FOR $t = 1, \dots, T$

1. Observe the true outcome $\omega_t \in \Omega$.

2. Suffer losses from past predictions

(a) Compute the losses $l_t^n = \lambda(\omega_t, \xi_t^n)$ for all $n \in \mathcal{N}$ of the experts' forecasts ξ_t^n made at the step $t - D$.

(b) Compute the loss $h_t = \lambda(\omega_t, \gamma_t)$ of the aggregating algorithm's forecast γ_t made at the step $t - D$.

3. Make the forecast for the next step (if $t \leq T - 1$)

(a) Get the experts $n \in \mathcal{N}$ predictions $\xi_{t+D}^n \in \Xi$ for the step $t + D$.

(b) Compute the aggregated prediction $\gamma_{t+D} \in \Gamma$ of the algorithm.

ENDFOR

We assume that the forecasts ξ_t^n of all experts $n \in \mathcal{N}$ for first D time moments $t = 1, \dots, D$ are given before the game.

The variables $L_T^n = \sum_{t=1}^T l_t^n$ (for all $n \in \mathcal{N}$) and $H_T = \sum_{t=1}^T h_t$ correspond to the cumulative losses of expert n and the aggregating algorithm over the entire game respectively. We also denote the vector of experts' forecasts for the step $t+D$ by $\xi_{t+D} = (\xi_{t+D}^1, \dots, \xi_{t+D}^N)$.

In the general protocol sets Ξ and $\Gamma \subset \Xi$ may not be equal. For example, the problem of combining N soft classifiers into a hard one has $\Xi = [0, 1]$ and $\Gamma = \{0, 1\} \subsetneq \Xi$. In this article we assume that the sets of possible experts' and algorithm's forecasts are equal, that is $\Xi = \Gamma$. Moreover, we assume that $\Xi = \Gamma$ is a convex set. We will not use the notation Ξ anymore.

The performance of the algorithm is measured by the (cumulative) regret. The cumulative regret is the difference between the cumulative loss of the aggregating algorithm and the cumulative loss of some given comparator. A typical approach is to compete with the best expert in the pool. The cumulative regret with respect to the best expert is

$$R_T = H_T - \min_{n \in \mathcal{N}} L_T^n. \quad (1)$$

The goal of the aggregating algorithm is to minimize the regret, that is, $R_T \rightarrow \min$. In order to theoretically guarantee algorithm's performance, some upper bound is usually proved for the cumulative regret $R_T \leq f(T)$.

In the base setting (1), sub-linear upper bound $f(T)$ for the regret leads to asymptotic performance of the algorithm equal to the performance of the best expert. More precisely, we have $\lim_{T \rightarrow \infty} \frac{R_T}{T} = 0$.

2.2. Exponentially concave loss functions

We investigate learning with exponentially concave loss functions. Loss function

$$\lambda : \Omega \times \Gamma \rightarrow \mathbb{R}_+$$

is called η -exponentially concave (for some $\eta > 0$) if for all $\omega \in \Omega$ and all probability distributions $\boldsymbol{\pi}$ on set Γ the following holds true:

$$e^{-\eta \lambda(\omega, \gamma_{\boldsymbol{\pi}})} \geq \int_{\gamma \in \Gamma} e^{-\eta \lambda(\omega, \gamma)} \boldsymbol{\pi}(d\gamma), \quad (2)$$

where

$$\gamma_{\boldsymbol{\pi}} = \int_{\gamma \in \Gamma} \gamma \boldsymbol{\pi}(d\gamma) = \mathbb{E}_{\boldsymbol{\pi}} \gamma. \quad (3)$$

In (2) variable $\gamma_{\boldsymbol{\pi}}$ is called aggregated prediction. Since Γ is convex, we have $\gamma_{\boldsymbol{\pi}} \in \Gamma$.

If a loss function is η -exponentially concave, it is also η' -exponentially concave for all $\eta' \in (0, \eta]$. This fact immediately follows from the general properties of exponentially concave functions. For more details see the book by Hazan (2016).

Note that the basic square loss and the log loss functions are both exponentially concave. This fact is proved by Kivinen and Warmuth (1999).

3. Aggregating Algorithm for 1-Step-Ahead Forecasting

In this section we discuss basic aggregating algorithms for 1-step-ahead forecasting based on exponential reweighing. Our framework is built on the general aggregating algorithm G_1 by [Vovk \(1999\)](#), we discuss it in Subsection 3.1. The simplest and earliest version V_1 by [Vovk \(1998\)](#) of this algorithm is discussed in Subsection 3.2.

3.1. General Model

We investigate the adversarial case, that is, no assumptions (functional, stochastic, etc) are made about the nature of data and experts. However, it turns out that in this case it is convenient to develop algorithms using some probabilistic interpretations.

Recall that $\xi_t = (\xi_t^1, \dots, \xi_t^N)$ is a vector of experts' predictions of ω_t . Loss function $\lambda : \Omega \times \Gamma \rightarrow \mathbb{R}_+$ is η -exponentially concave for some $\eta > 0$.

We assume that data is generated using some probabilistic model with hidden states. The model is shown in Figure 2.

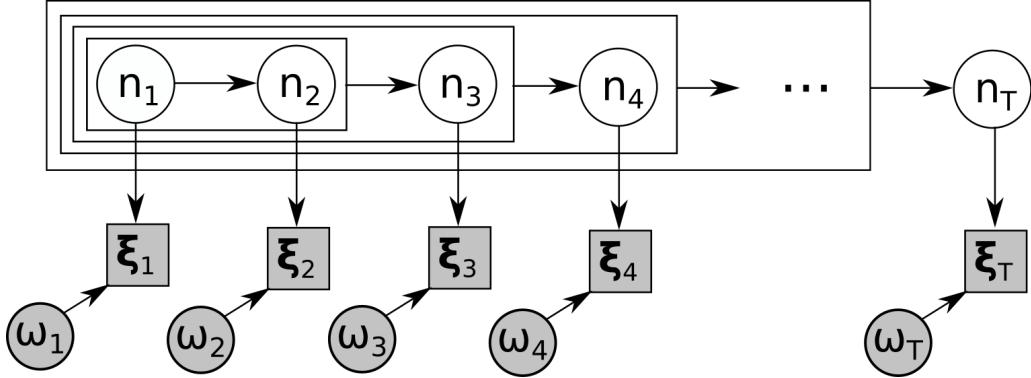


Figure 2: Probabilistic model of data generation process.

We suppose that there is some hidden sequence of experts $n_t \in \mathcal{N}$ (for $t = 1, 2, \dots, T$) that generates the experts predictions ξ_t . Particular, hidden expert n_t at step t is called active expert. The conditional probability to observe the vector ξ_t of experts' predictions at step t is

$$p(\xi_t | n_t) = \frac{e^{-\eta\lambda(\omega_t, \xi_t^{n_t})}}{Z_t},$$

where $Z_t = \int_{\xi \in \Gamma} e^{-\eta\lambda(\omega_t, \xi)} d\xi$ is the normalizing constant.⁶ We denote $\Xi_t = (\xi_1, \dots, \xi_t)$ and $\Omega_t = (\omega_1, \dots, \omega_t)$ for all $t = 1, \dots, T$.

For the first active expert n_1 some known prior distribution is given $p(n_1) = p_0(n_1)$. The sequence (n_1, \dots, n_T) of active experts is generated step-by-step. For $t \in \{1, \dots, T-1\}$

6. Constant Z_t is n_t -independent. In the article we do not need to compute the exact value of the normalizing constant Z_t .

each n_{t+1} is sampled from some known distribution $p(n_{t+1}|\mathcal{N}_t)$, where $\mathcal{N}_t = (n_1, \dots, n_t)$.⁷ Thus, active expert n_{t+1} depends on the previous experts \mathcal{N}_t .

The considered probabilistic model is.⁸

$$p(\mathcal{N}_T, \boldsymbol{\Xi}_T) = p(\mathcal{N}_T) \cdot p(\boldsymbol{\Xi}_T|\mathcal{N}_T) = \left[p_0(n_1) \prod_{t=2}^T p(n_t|\mathcal{N}_{t-1}) \right] \cdot \left[\prod_{t=1}^T p(\boldsymbol{\xi}_t|n_t) \right]. \quad (4)$$

A similar equation holds true for $t \leq T$:

$$p(\mathcal{N}_t, \boldsymbol{\Xi}_t) = p(\mathcal{N}_t) \cdot p(\boldsymbol{\Xi}_t|\mathcal{N}_t) = \left[p_0(n_1) \prod_{\tau=2}^t p(n_\tau|\mathcal{N}_{\tau-1}) \right] \cdot \left[\prod_{\tau=1}^t p(\boldsymbol{\xi}_\tau|n_\tau) \right].$$

The probability $p(\mathcal{N}_T)$ is that of hidden states (active experts).⁹

Suppose that the current time moment is t . We observe the experts' predictions $\boldsymbol{\Xi}_t$ made earlier, time series Ω_t and predictions $\boldsymbol{\xi}_{t+1}$ for the step $t+1$. Since we observe Ω_t and $\boldsymbol{\Xi}_t$, we are able to estimate the conditional distribution $p(\mathcal{N}_t|\boldsymbol{\Xi}_t)$ of hidden variables \mathcal{N}_t . This estimate allows us to compute the conditional distribution on the active expert n_{t+1} at the moment $t+1$. We denote for all $n_{t+1} \in \mathcal{N}$

$$w_{t+1}^{n_{t+1}} = p(n_{t+1}|\boldsymbol{\Xi}_t) = \sum_{\mathcal{N}_t \in \mathcal{N}^t} p(n_{t+1}|\mathcal{N}_t) p(\mathcal{N}_t|\boldsymbol{\Xi}_t). \quad (5)$$

We use the weight vector $\mathbf{w}_{t+1} = (w_{t+1}^1, \dots, w_{t+1}^N)$ to combine the aggregated prediction for the step $t+1$:

$$\gamma_{t+1} = \sum_{n_{t+1} \in \mathcal{N}} \xi_{t+1}^{n_{t+1}} p(n_{t+1}|\boldsymbol{\Xi}_t) = \langle \mathbf{w}_{t+1}, \boldsymbol{\xi}_{t+1} \rangle = \sum_{n=1}^N w_{t+1}^n \xi_{t+1}^n.$$

The aggregating algorithm is shown below. We denote it by $G_1 = G_1(p)$, where p indicates the probability distribution $p(\mathcal{N}_T)$ of active experts to which the algorithm is applied.

Algorithm $G_1(p)$ (Aggregating algorithm for distribution p of active experts)

Set initial prediction weights $\mathbf{w}_1^{n_1} = p_0(n_1)$.

Get the experts $n \in \mathcal{N}$ predictions $\xi_1^n \in \Gamma$ for the step $t = 1$.

Compute the aggregated prediction $\gamma_1 = \langle \mathbf{w}_1, \boldsymbol{\xi}_1 \rangle$ for the step $t = 1$.

FOR $t = 1, \dots, T$

1. Observe the true outcome $\omega_t \in \Omega$.

7. In case $p(n_{t+1}|\mathcal{N}_t) = p(n_{t+1}|n_t)$, we obtain traditional Hidden Markov Process. The hidden state at step $t+1$ depends only on the previous hidden state at step t .

8. The correct way is to include the time series values ω_t as the conditional parameter in the model probabilistic distribution, that is, $p(\mathcal{N}_T, \boldsymbol{\Xi}_T|\Omega_T)$. We omit the values ω_t in probabilities $p(\cdot)$ in order not to overburden the notation.

9. The form $p(\mathcal{N}_T) = p_0(n_1) \prod_{t=2}^T p(n_t|\mathcal{N}_{t-1})$ is used only for convenience and association with online scenario. It does not impose any restrictions on the type of probability distribution. In fact, $p(\mathcal{N}_T)$ may be any distribution on \mathcal{N}^T .

2. Update the weights

(a) Calculate the prediction weights $\mathbf{w}_{t+1} = (w_t^1, \dots, w_t^N)$, where

$$w_t^{n_{t+1}} = p(n_{t+1} | \Xi_t)$$

for all $n_{t+1} \in \mathcal{N}$.

3. Make forecast for the next step (if $t \leq T-1$)

(a) Get the experts $n \in \mathcal{N}$ predictions $\xi_{t+1}^n \in \Gamma$ for the step $t+1$.

(b) Combine the aggregated prediction $\gamma_{t+1} = \langle \mathbf{w}_{t+1}, \xi_{t+1} \rangle \in \Gamma$ of the algorithm.

ENDFOR

To estimate the performance of the obtained algorithm, we prove Theorem 1. Recall that H_T is the cumulative loss of the algorithm over the entire game.

Theorem 1 *For the algorithm G_1 applied to the model (4) the following upper bound for the cumulative loss over the entire game holds true:*

$$H_T \leq -\frac{1}{\eta} \ln \left[\mathbb{E}_{p(\mathcal{N}_T)} [e^{-\eta L_T^{\mathcal{N}_T}}] \right]. \quad (6)$$

Similar results were obtained by [Vovk \(1999\)](#). In this article we reformulate these results in terms of our interpretable probabilistic framework. This is required for the completeness of the exposition and theoretical analysis of algorithm G_D (see Subsection 4.1).

Proof Define the mixloss at the step t :

$$m_t = -\frac{1}{\eta} \ln \left[\sum_{n_t \in \mathcal{N}} e^{-\eta \lambda(\omega_t, \xi_t^{n_t})} \cdot w_t^{n_t} \right]. \quad (7)$$

Since λ is η -exponentially concave function, for the aggregated prediction $\gamma_t = \langle \mathbf{w}_t, \xi_t \rangle$ and probability distribution \mathbf{w}_t we have

$$e^{-\eta \lambda(\omega_t, \gamma_t)} \geq \sum_{n_t \in \mathcal{N}} e^{-\eta \lambda(\omega_t, \xi_t^{n_t})} w_t^{n_t},$$

which is equal to $e^{-\eta h_t} \geq e^{-\eta m_t}$. We conclude that $h_t \leq m_t$ for all $t \in \{1, \dots, T\}$. Thus, the similar inequality is true for the cumulative loss of the algorithm and the cumulative mixloss:

$$H_T = \sum_{t=1}^T h_t \leq \sum_{t=1}^T m_t = M_T.$$

Now lets compute M_T .

For all t

$$\begin{aligned}
 m_t &= -\frac{1}{\eta} \ln \left[\sum_{n_t \in \mathcal{N}} e^{-\eta \lambda(\omega_t, \xi_t^{n_t})} \cdot p(n_t | \Xi_{t-1}) \right] = \\
 &= -\frac{1}{\eta} \ln \left[\sum_{n_t \in \mathcal{N}} Z_t \cdot p(\xi_t | n_t) \cdot p(n_t | \Xi_{t-1}) \right] = \\
 &= -\frac{1}{\eta} \ln Z_t - \frac{1}{\eta} \ln \left[\sum_{n_t \in \mathcal{N}} p(\xi_t | n_t) \cdot p(n_t | \Xi_{t-1}) \right] = \\
 &= -\frac{1}{\eta} \ln Z_t - \frac{1}{\eta} \ln p(\xi_t | \Xi_{t-1}).
 \end{aligned}$$

We compute

$$\begin{aligned}
 H_T \leq M_T &= \sum_{t=1}^T m_t = \frac{1}{\eta} \ln \prod_{t=1}^T Z_t - \frac{1}{\eta} \ln \prod_{t=1}^T p(\xi_t | \Xi_{t-1}) = \\
 &= \frac{1}{\eta} \ln \prod_{t=1}^T Z_t - \frac{1}{\eta} \ln p(\Xi_T) = -\frac{1}{\eta} \ln \left[\mathbb{E}_{p(\mathcal{N}_T)} [e^{-\eta L_T^{\mathcal{N}_T}}] \right]
 \end{aligned} \tag{8}$$

and finish the proof. ■

In the current form it is difficult to understand the meaning of the theorem. However, the main idea is partially shown in the following corollary.

Corollary 2 *The regret of the algorithm $G_1(p)$ with respect to the sequence $\mathcal{N}_T^* = \{n_1^*, \dots, n_T^*\}$ of experts has the following upper bound:*

$$R_T(\mathcal{N}_T^*) = H_T - \sum_{t=1}^T l_t^{n_t^*} \leq -\frac{1}{\eta} \ln p(\mathcal{N}_T^*). \tag{9}$$

Proof We simply bound (6):

$$H_T \leq -\frac{1}{\eta} \ln \left[\mathbb{E}_{p(\mathcal{N}_T)} [e^{-\eta L_T^{\mathcal{N}_T}}] \right] \leq -\frac{1}{\eta} \ln \left[p(\mathcal{N}_T^*) \cdot e^{-\eta L_T^{\mathcal{N}_T^*}} \right] = \sum_{t=1}^T l_t^{n_t^*} - \frac{1}{\eta} \ln p(\mathcal{N}_T^*). \quad \blacksquare$$

Applying algorithm G_1 to different probability models p makes it possible to obtain the upper regret bound with respect to particular sequences \mathcal{N}_T .¹⁰ Choosing different p makes it possible to obtain adaptive algorithms (e.g. Fixed Share by [Herbster and Warmuth](#)

10. In this article we do not raise the question of computational efficiency of algorithm $G_1(p)$. In fact, computational time and required memory can be pretty high for complicated distributions p , even $O(N^T)$. Nevertheless, all the special algorithms that we consider (V_1, V_D, V_D^{FC}) are computationally efficient. They require $\leq O(NT)$ computational time and $\leq O(ND)$ memory.

(1998), Adamskiy et al. (2016), Vovk (1999)). Such algorithms provide low regret not only with respect to the best constant sequence of experts, but also with some more complicated sequences.

For example, suppose we are to obtain minimal possible regret bound with respect to the subset of sequences $(\mathcal{N}^T)^* \subset \mathcal{N}^T$. In this case it is reasonable to set $p(\mathcal{N}_T) = \frac{1}{|(\mathcal{N}^T)^*|}$ for all $\mathcal{N}_T \in (\mathcal{N}^T)^*$ and $p(\mathcal{N}_T) = 0$ for all other \mathcal{N}_T .

Nevertheless, the most popular approach is to compete with the best (fixed) expert. This approach is simple and, at the same time, serves as the basis for our research.

3.2. Case of Hidden Markov Process: Classical Vovk's Algorithm

Consider the simplified dependence of active experts shown on Figure 3. For all t expert n_{t+1} depends only on the previous expert n_t , that is, $p(n_{t+1}|\mathcal{N}_T) = p(n_{t+1}|n_t)$.

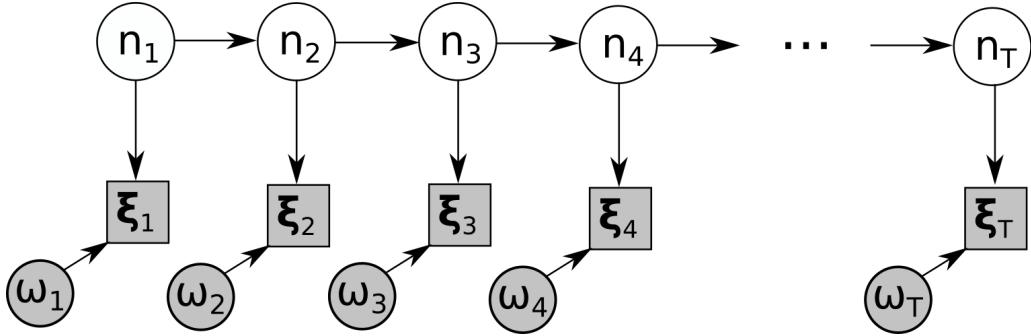


Figure 3: Hidden Markov Model model for data generation process.

The classic Vovk's aggregating algorithm V_1 is obtained from G_1 by applying it to the simple distribution p . We put $p(n_1) = \frac{1}{N}$ (for all $n_1 \in \mathcal{N}$) and $p(n_t|n_{t-1}) = [n_t = n_{t-1}]$ (for all $t > 1$ and $n_t, n_{t-1} \in \mathcal{N}$). We denote the algorithm G_1 for the described p by V_1 .

According to Corollary 2, algorithm V_1 has the following regret bound with respect to the best constant expert $\mathcal{N}_T^* = (n^*, \dots, n^*)$:

$$R_T(\mathcal{N}_T^*) = H_T - \sum_{t=1}^T l_t^{n^*} \leq -\frac{1}{\eta} \ln p(\mathcal{N}_T^*) = \frac{\ln N}{\eta}. \quad (10)$$

At the same time, it is easy to recurrently compute the weights \mathbf{w}_t step by step, that is, $\mathbf{w}_1 \rightarrow \mathbf{w}_2 \rightarrow \dots$. We get

$$w_{t+1}^{n_{t+1}} = p(n_t|\Xi_t) = \frac{e^{-\eta L_t^{n_t}}}{\sum_{n=1}^N e^{-\eta L_t^n}} = \frac{w_t^{n_t} e^{-\eta l_t^{n_t}}}{\sum_{n=1}^N w_t^n e^{-\eta l_t^n}}.$$

Thus, algorithm V_1 is a powerful and computationally efficient tool to aggregate experts in the game of the 1-step-ahead forecasting.

4. Aggregating Algorithm for Long-Term Forecasting

The Section is devoted to aggregating algorithms for the long-term forecasting. In Subsection 4.1 we provide a natural long-term forecasting generalization G_D of algorithm G_1 by [Vovk \(1999\)](#). We provide the general regret bound and discuss the difficulties that prevent us from obtaining general bound in a simple form. In Subsection 4.2 we show how the replicated version V_D for the long-term forecasting of V_1 fits into the general model, and prove its regret bound. In Subsection 4.3 we describe non-replicated version V_D^{FC} for the long-term forecasting of V_1 and prove its $O(\sqrt{T})$ regret bound.

4.1. General Model

We describe the natural algorithm obtained by enhancing G_1 for the problem of the D -th-step-ahead forecasting below and denote it by G_D . Note that weights \mathbf{w}_t in G_D differ for different D (for the same probability model p).

Algorithm $G_D(p)$ (Aggregating algorithm for distribution p of active experts)

Set initial prediction weights $w_t^{n_t} = p(n_t)$ for all $t = 1, \dots, D$ and $n_t \in \mathcal{N}$.

Get the predictions $\xi_t^n \in \Gamma$ of experts $n \in \mathcal{N}$ for steps $t = 1, \dots, D$.

Compute the aggregated predictions $\gamma_t = \langle \mathbf{w}_t, \xi_t \rangle$ for steps $t = 1, \dots, D$.

FOR $t = 1, \dots, T$

1. Observe the true outcome $\omega_t \in \Omega$.

2. Update the weights

(a) Calculate the prediction weights $\mathbf{w}_{t+D} = (w_t^1, \dots, w_t^N)$, where

$$w_t^{n_{t+D}} = p(n_{t+D} | \Xi_t)$$

for all $n_{t+D} \in \mathcal{N}$.

3. Make forecast for the next step (if $t \leq T-1$)

(a) Get the predictions $\xi_{t+D}^n \in \Gamma$ of experts $n \in \mathcal{N}$ for the step $t+D$.

(b) Combine the aggregated prediction $\gamma_{t+D} = \langle \mathbf{w}_{t+D}, \xi_{t+D} \rangle \in \Gamma$ of the algorithm.

ENDFOR

Despite the fact that algorithm G_D (for $D > 1$) is the direct modification of G_1 , it seems hard to obtain the adequate general bound of the loss of the form (6). Indeed, let us try to apply the ideas of Theorem 1 proof to algorithm G_D .

Denote by m_t the mixloss from (7). Recall that the weights $w_t^{n_t}$ here are equal to the probabilities $p(n_t | \Xi_{t-D})$ but not $p(n_t | \Xi_{t-1})$. Again, for η -exponentially concave function we have $h_t \leq m_t$. Similar to the proof of Theorem 1, we compute for all t

$$m_t = -\frac{1}{\eta} \ln Z_t - \frac{1}{\eta} \ln p(\xi_t | \Xi_{t-D}),$$

where we assume $\Xi_{t-D} = \emptyset$ for $t \leq D$. The cumulative mixloss is equal to

$$H_t \leq M_t = \sum_{t=1}^T m_t = \frac{1}{\eta} \ln \prod_{t=1}^T Z_t - \frac{1}{\eta} \ln \prod_{t=1}^T p(\xi_t | \Xi_{t-D}). \quad (11)$$

Unfortunately, for $D > 1$ in the general case this expression can not be simplified in the same way as in Theorem 1 for G_1 .

However, there exist some simple theoretical cases when this bound can be simplified. For a fixed D , the obvious one is when

$$p(\mathcal{N}_T) = \prod_{t=T-D+1}^T p(\widehat{\mathcal{N}}_t), \quad (12)$$

where we use $\widehat{\mathcal{N}}_t = (\dots, n_{t-2D}, n_{t-D}, n_t)$ for all $t = 1, \dots, T$. Note that (12) means that probability distributions on separate grids GR_d (for $d = 1, \dots, D$) are independent. In this case, the learning process separates into D disjoint one-step ahead forecasting games on grids GR_d . We have $p(\xi_t | \Xi_{t-D}) = p(\xi_t | \widehat{\Xi}_{t-D})$ and (11) is simplified to

$$\begin{aligned} H_t \leq M_t &= \frac{1}{\eta} \ln \prod_{t=1}^T Z_t - \frac{1}{\eta} \ln \prod_{t=T-D+1}^T p(\widehat{\Xi}_t) = \\ &= \frac{1}{\eta} \ln \prod_{t=1}^T Z_t - \frac{1}{\eta} \ln \prod_{t=T-D+1}^T p(\widehat{\Xi}_t) = -\frac{1}{\eta} \ln \left[\mathbb{E}_{p(\mathcal{N}_T)} [e^{-\eta L_T^{\mathcal{N}_T}}] \right], \end{aligned}$$

where $\widehat{\Xi}_t = \{\xi_t, \xi_{t-D}, \xi_{t-2D}, \dots\}$ for all t .

In Appendix A we prove that the approach (12) may be considered as optimal when the goal is to compete with the best expert. Nevertheless, there are no guarantees that this approach is optimal in the general case.

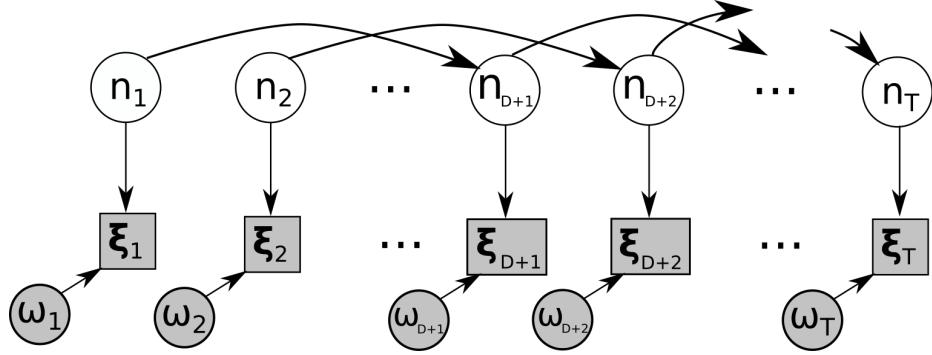
4.2. Optimal Approach: Replicated Vovk's Algorithm

Algorithm V_1 can be considered as close to optimal (for the 1-step-ahead forecasting and competing with the best expert in the pool) because of its low constant regret bound $R_T \leq \frac{\ln N}{\eta}$. One can obtain the aggregating algorithm V_D for the long-term forecasting whose regret is close to optimal when competing with the best constant active expert. The idea is to run D independent one-step-ahead forecasting algorithms V_1 on D disjoint subgrids GR_d (for $d \in \{1, \dots, D\}$). This idea is motivated by Theorem 4 from Appendix A.

To show how this method fits into the general model G_D , we define $p(\mathcal{N}_D) \equiv (\frac{1}{N})^D$ for all $\mathcal{N}_D \in \mathcal{N}^D$. Next, for all $t > D$ we set $p(n_{t+1} | \mathcal{N}_t) = p(n_{t+1} | n_{t+1-D}) = [n_{t+1} = n_{t+1-D}]$, that is, active expert n_{t+1} depends only on expert n_{t+1-D} that was active D steps ago. The described model is shown in Figure 4.

For simplicity, we assume that T is a multiple of D . Since V_D runs D independent copies of V_1 , it is easy to estimate its regret with respect to the best expert:

$$R_T(\mathcal{N}_T^*) = H_T - \sum_{t=1}^T l_t^{n^*} \leq -\frac{1}{\eta} \ln p(\mathcal{N}_T^*) = D \frac{\ln N}{\eta}.$$

Figure 4: The probabilistic model for algorithm V_D .

At the same time, the weights' updating process is simple (it can be separated into grids). We have

$$w_t^{n_t} = p(n_t | \Xi_{t-D}) = \frac{e^{-\eta \hat{L}_{t-D}^{n_t}}}{\sum_{n=1}^N e^{-\eta \hat{L}_{t-D}^n}} = \frac{w_{t-D}^{n_t} e^{-\eta l_{t-D}^{n_t}}}{\sum_{n=1}^N w_{t-D}^n e^{-\eta l_{t-D}^n}},$$

where $\hat{L}_t^{n_t} = \sum_{\tau=1}^t l_{\tau}^{n_t}$ for all t . This formula allows efficient recurrent computation

$$\dots \rightarrow \mathbf{w}_{t-D} \rightarrow \mathbf{w}_t \rightarrow \mathbf{w}_{t+D} \rightarrow \dots$$

4.3. Practical Approach: Non-Replicated Vovk's Algorithm

Despite the fact that algorithm V_D is theoretically close to optimal for competing with the best expert, it has several practically obvious disadvantages. With the increase of D the overall length of subgames decreases ($\sim \frac{T}{D}$) and subgrids become more infrequent. Moreover, to set the forecasting weight \mathbf{w}_{t+D} at step t we use only $\approx \frac{t}{D}$ previous observations and forecasts.

One may wonder why not use all the observed losses to set the weight \mathbf{w}_{t+D} . We apply the algorithm G_D to the probability distribution p from Subsection 3.2 which is a case of model from Figure 3. We denote the obtained algorithm by V_D^{FC} .

The weights are efficiently recomputed $\dots \rightarrow \mathbf{w}_{t-1} \rightarrow \mathbf{w}_t \rightarrow \mathbf{w}_{t+1} \rightarrow \dots$ according to the formula

$$w_t^{n_t} = p(n_t | \Xi_{t-D}) = \frac{e^{-\eta L_{t-D}^{n_t}}}{\sum_{n=1}^N e^{-\eta L_{t-D}^n}} = \frac{w_{t-1}^{n_t} e^{-\eta l_{t-D}^{n_t}}}{\sum_{n=1}^N w_{t-1}^n e^{-\eta l_{t-D}^n}}.$$

Theorem 3 *For the η^λ -exponentially concave (for some η^λ) and L -Lipshitz for all ω w.r.t. second argument $\gamma \in \Gamma$ and $\|\cdot\|_\Gamma$ loss function*

$$\lambda : \Omega \times \Gamma \rightarrow [0, H] \subset \mathbb{R}_+$$

with $\max_{\gamma \in \Gamma} \|\gamma\|_\Gamma \leq B$ there exists T_0 such that for all $T \geq T_0$ the following holds true: there exists $\eta^\lambda > \eta^ > 0$ such that algorithm $V_D^{FC} = G_D(p)$ with N experts and learning*

rate η^* has regret bound

$$R_T \leq O(\sqrt{N \ln N} \cdot \sqrt{T})$$

with respect to the best expert in the pool.

Proof We use the superscript $(\dots)^D$ to denote the variables obtained by algorithm $G_D(p)$ (for example, weights \mathbf{w}_t^D , predictions γ_t^D , etc.). Our main idea is to prove that the weights \mathbf{w}_t^D are approximately equal to the weights \mathbf{w}_t^1 obtained in the one-step-ahead forecasting game $G_1(p)$ with the same experts and same time series ω_t . Thus, the forecasts γ_t^D and γ_t^1 have approximately the same losses h_t^D and h_t^1 respectively.

We compare both algorithms with the same (yet unknown) learning rate η . Note that $\mathbf{w}_{t+1}^1 = \mathbf{w}_{t+D}^D$. For all t we have

$$(w_t^{n_t})^1 = (w_{t+D-1}^{n_t})^D \sim (w_t^{n_t})^D \cdot e^{-\eta L_{[t-D+1, t-1]}^{n_t}},$$

where $L_{[t-D+1, t-1]}^{n_t} = \sum_{\tau=t-D+1}^{t-1} l_{\tau}^{n_t}$. We estimate the difference between the cumulative losses H_T^1 and H_T^D of forecasts of aggregating algorithms $G_1(p) = V_1$ and $G_D(p) = V_D^{FC}$ for the given p from Subsection 3.2.

$$\begin{aligned} |H_T^1 - H_T^D| &= \left| \sum_{t=1}^T h_t^1 - \sum_{t=1}^T h_t^D \right| \leq \sum_{t=1}^T |h_t^1 - h_t^D| = \sum_{t=1}^T |\lambda(\omega_t, \gamma_t^1) - \lambda(\omega_t, \gamma_t^D)| \leq \\ &L \sum_{t=1}^T \|\gamma_t^1 - \gamma_t^D\|_{\Gamma} = L \sum_{t=1}^T \|\langle \mathbf{w}_t^1, \boldsymbol{\xi}_t \rangle - \langle \mathbf{w}_t^D, \boldsymbol{\xi}_t \rangle\|_{\Gamma} = L \sum_{t=1}^T \|\langle \mathbf{w}_t^1 - \mathbf{w}_t^D, \boldsymbol{\xi}_t \rangle\|_{\Gamma} \leq \\ &L \sum_{t=1}^T \sum_{n=1}^N |(w_t^n)^1 - (w_t^n)^D| \cdot \|\boldsymbol{\xi}_t^n\|_{\Gamma} \leq BL \sum_{t=1}^T \sum_{n=1}^N |(w_t^n)^1 - (w_t^n)^D| \leq \\ &BLTN \cdot \max_{t,n} |(w_t^n)^1 - (w_t^n)^D| \quad (13) \end{aligned}$$

Our goal is to estimate the maximum. In fact, we are to estimate the maximum possible single weight change over $D-1$ steps in algorithm $G_1(p)$ or $G_D(p)$.

W.l.o.g. we assume that the maximum is achieved at step t on the 1-st coordinate ($n=1$). We denote $\mathbf{x} = \mathbf{w}_t^D \in \Delta_N$ and $\mathbf{a} \in \Delta_N$, where $a_n \sim e^{-\eta L_{[t-D+1, t-1]}^n}$ (so that $(w_t^n)^1 \sim a_n x_n$). The latter equation imposes several restrictions on \mathbf{a} . In fact, for all $n, n' \in \mathcal{N}$ the following must be true: $\frac{a_n}{a_{n'}} \leq e^{-\eta(D-1)H}$, where H is the upper bound for the loss $\lambda(\omega, \gamma)$. We denote the subset of such vectors \mathbf{a} by $\Delta'_N \subsetneq \Delta_N$.

Note that $\mathbf{w}_t^1 \sim (x_1 \cdot a_1, x_2 \cdot a_2, \dots, x_N \cdot a_N)$. We are to bound the maximum

$$\max_{x \in \Delta_N} \left[\max_{a \in \Delta'_N} \left[\left| x_1 - \frac{x_1 a_1}{\sum_{n=1}^N x_n a_n} \right| \right] \right].$$

We consider the case $x_1 > \frac{x_1 a_1}{\sum_{n=1}^N x_n a_n}$ (the other case is similar). In this case

$$\max_{x \in \Delta_N} \left[\max_{a \in \Delta'_N} \left[x_1 - \frac{x_1 a_1}{\sum_{n=1}^N x_n a_n} \right] \right] = \max_{x \in \Delta_N} \left(x_1 \left[\max_{a \in \Delta'_N} \left[1 - \frac{a_1}{\sum_{n=1}^N x_n a_n} \right] \right] \right).$$

We examine the behavior of $[1 - \frac{a_1}{\sum_{n=1}^N x_n a_n}]$ under the fixed \mathbf{x} .

$$\left[1 - \frac{a_1}{\sum_{n=1}^N x_n a_n} \rightarrow \max_{\mathbf{a} \in \Delta'_N} \right] \iff \left[\frac{\sum_{n=2}^N x_n a_n}{a_1} = \sum_{n=2}^N \frac{a_n}{a_1} x_n \rightarrow \max_{\mathbf{a} \in \Delta'_N} \right] \quad (14)$$

Since $\frac{a_n}{a_1} \leq e^{-\eta(D-1)H}$, we have $\frac{\sum_{n=2}^N x_n a_n}{a_1} \leq (1 - x_1) e^{-\eta(D-1)H}$, and the argument \mathbf{a} that maximizes (14) does not depend on x_2, \dots, x_N . W.l.o.g. we can assume that $\mathbf{a} = (a, \frac{1-a}{N-1}, \frac{1-a}{N-1}, \dots, \frac{1-a}{N-1})$, where $a = \frac{e^{-\eta(D-1)H}}{(N-1) + e^{-\eta(D-1)H}} < \frac{1}{N}$. At the same time, we can assume that $\mathbf{x} = (x, \frac{1-x}{N-1}, \frac{1-x}{N-1}, \dots, \frac{1-x}{N-1})$. Thus,

$$\max_{x \in \Delta_N} \left[\max_{a \in \Delta'_N} \left[x_1 - \frac{x_1 a_1}{\sum_{n=1}^N x_n a_n} \right] \right] = \max_{x \in (0,1)} \left[x - \frac{x a}{x a + \frac{(1-x)(1-a)}{N-1}} \right].$$

The derivative is equal to zero at $x^* = \frac{1-a-\sqrt{a(1-a)(N-1)}}{1-aN}$. Substituting $x = x^*$ and $a = \frac{e^{-\eta(D-1)H}}{(N-1) + e^{-\eta(D-1)H}}$ we obtain the N -independent expression:

$$\frac{(1 - \sqrt{e^{-\eta(D-1)H}})^2}{1 - e^{-\eta(D-1)H}}. \quad (15)$$

We are interested in the behavior near $\eta = 0$. A closer look at the Maclaurin series of numerator and denominator give us the decomposition

$$\frac{0 + 0\eta + \eta^2 \frac{(D-1)^2 H^2}{4} + \dots}{0 + \eta(D-1)H + \dots}.$$

At $\eta \rightarrow 0$ the function is equal to 0. There exist some $\eta^0 > 0$, such that for all $0 \leq \eta < \eta_0$ expression (15) can be bounded by some η -linear function $U(D, H) \cdot \eta = [\frac{(D-1)H}{4} + \epsilon]\eta$. Thus, expression (13) is bounded by

$$TN \cdot \left[B \cdot L \cdot N \cdot U(D, H) \right] \cdot \eta = FNT\eta$$

We set $F = B \cdot L \cdot U(D, H)$. Next,

$$H_T^D \leq H_T^1 + |H_T^1 - H_T^D| \leq [L_T^* + \frac{\ln N}{\eta}] + FNT\eta,$$

where $L_T^* = \min_{n \in \mathcal{N}} L_T^n = \min_{n \in \mathcal{N}} [\sum_{t=1}^T l_t^n]$ is the loss of the best constant expert.

Choosing $\eta^* = \arg \min_{\eta > 0} [\frac{\ln N}{\eta} + FNT\eta] = \sqrt{\frac{\ln N}{FNT}}$, we obtain

$$R_T \leq 2\sqrt{FN \ln N} \cdot \sqrt{T} = O(\sqrt{N \ln N} \sqrt{T})$$

regret bound with respect to the best expert in the pool.

Note that in order to use linear approximation of maximum we need $\eta \leq \eta^0$. Moreover, to bound the loss H_T^1 we need η^* -exponentially concave function λ . This means that $\eta^* \leq \min\{\eta^0, \eta^\lambda\}$. Since $\eta^* = \sqrt{\frac{\ln N}{FNT}} \sim \frac{1}{\sqrt{T}}$, there exists some huge T_0 such that for $T \geq T_0$ the required conditions are met. \blacksquare

5. Conclusion

The problem of long-term forecasting is of high importance. In the article we developed general algorithm $G_D(p)$ for the D -th-step ahead forecasting with experts' advice (where p is the distribution over active experts). The algorithm uses the ideas of the general aggregating algorithm G_1 by Vovk (1999). We also provided the expression for the upper bound for the loss of algorithm $G_D(p)$ for any probability distribution p over active experts.

For its important special case V_D^{FC} we proved the $O(\sqrt{T})$ regret bound w.r.t. the best expert in the pool. Algorithm V_D^{FC} is a practical long-term forecasting modification of algorithm V_1 by Vovk (1998).

It seems possible to apply the approach from the proof of Theorem 3 in order to obtain simpler and more understandable loss bound for algorithm $G_D(p)$ for any probability distribution p . This statement serves as the challenge for our further research.

References

Dmitry Adamskiy, Wouter M. Koolen, Alexey Chernov, and Vladimir Vovk. A closer look at adaptive regret. *Journal of Machine Learning Research*, 17(23):1–21, 2016. URL <http://jmlr.org/papers/v17/13-533.html>.

Oren Anava, Elad Hazan, and Shie Mannor. Online learning for adversaries with memory: Price of past mistakes. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 784–792, Cambridge, MA, USA, 2015. MIT Press. URL <http://dl.acm.org/citation.cfm?id=2969239.2969327>.

Olivier Bousquet and Manfred K. Warmuth. Tracking a small set of experts by mixing past posteriors. *J. Mach. Learn. Res.*, 3:363–396, March 2003. ISSN 1532-4435. URL <http://dl.acm.org/citation.cfm?id=944919.944940>.

Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006. ISBN 0521841089.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139, 1997. ISSN 0022-0000. doi: <https://doi.org/10.1006/jcss.1997.1504>. URL <http://www.sciencedirect.com/science/article/pii/S002200009791504X>.

Elad Hazan. Introduction to online convex optimization. *Found. Trends Optim.*, 2(3–4):157–325, August 2016. ISSN 2167-3888. doi: [10.1561/2400000013](https://doi.org/10.1561/2400000013). URL <https://doi.org/10.1561/2400000013>.

Mark Herbster and Manfred K. Warmuth. Tracking the best expert. *Mach. Learn.*, 32(2): 151–178, August 1998. ISSN 0885-6125. doi: [10.1023/A:1007424614876](https://doi.org/10.1023/A:1007424614876). URL <https://doi.org/10.1023/A:1007424614876>.

Jyrki Kivinen and Manfred K. Warmuth. Averaging expert predictions. In Paul Fischer and Hans Ulrich Simon, editors, *Computational Learning Theory*, pages 153–167, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. ISBN 978-3-540-49097-5.

Alexander Korotin, Vladimir V'yugin, and Eugeny Burnaev. Long-term sequential prediction using expert advice. *CoRR*, abs/1711.03194, 2017. URL <http://arxiv.org/abs/1711.03194>.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, February 1994. ISSN 0890-5401. doi: 10.1006/inco.1994.1009. URL <http://dx.doi.org/10.1006/inco.1994.1009>.

Kent Quanrud and Daniel Khashabi. Online learning with adversarial delays. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1270–1278. Curran Associates, Inc., 2015. URL <http://papers.nips.cc/paper/5833-online-learning-with-adversarial-delays.pdf>.

V Vovk. A game of prediction with expert advice. *J. Comput. Syst. Sci.*, 56(2):153–173, April 1998. ISSN 0022-0000. doi: 10.1006/jcss.1997.1556. URL <http://dx.doi.org/10.1006/jcss.1997.1556>.

V. Vovk. Derandomizing stochastic prediction strategies. *Machine Learning*, 35(3):247–282, Jun 1999. ISSN 1573-0565. doi: 10.1023/A:1007595032382. URL <https://doi.org/10.1023/A:1007595032382>.

Volodimir G. Vovk. Aggregating strategies. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, COLT '90, pages 371–386, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc. ISBN 1-55860-146-5. URL <http://dl.acm.org/citation.cfm?id=92571.92672>.

M. J. Weinberger and E. Ordentlich. On delayed prediction of individual sequences. *IEEE Transactions on Information Theory*, 48(7):1959–1976, Jul 2002. ISSN 0018-9448. doi: 10.1109/TIT.2002.1013136.

Appendix A. Optimal Approach to Delayed Feedback Forecasting with Experts' Advice

The appendix is devoted to obtaining the minimax regret bound for the problem of the D -th-step-ahead forecasting as a function of the minimax bound for the 1-step-ahead forecasting. We consider the general protocol of the D -th-step-ahead forecasting game with experts' advice from Subsection 2.1. Within the framework of the task, we desire to compete with the best expert in the pool.

We use Ω_t to denote the sequence $(\omega_1, \dots, \omega_t)$ of time series values at first t steps. Pair $I_t = (\Omega_t, \Xi_{t+D})$ is the information that an online algorithm knows on the step t of the game. Let $S_{D,t}$ be the set of all possible online randomized prediction aggregation algorithms with the forecasting horizon D and game length t . Each online algorithm $s \in S_{D,t}$ for given time series Ω_t and experts answers Ξ_t provides a sequence of distributions $(\pi_1^s(\gamma), \dots, \pi_t^s(\gamma))$, where each distribution $\pi_\tau^s(\gamma)$ is a function of $I_{\tau-D}$ for $\tau = 1, \dots, t$. We write $\pi_\tau^s(\gamma) = \pi^s(\gamma | I_{\tau-D})$.

The expected cumulative loss of the algorithm $s \in S_{D,t}$ on a given I_t is

$$H_t^D(s, I_t) = \sum_{\tau=1}^t [\mathbb{E}_{\pi_\tau^s} \lambda(\omega_\tau, \gamma)]$$

and the cumulative loss of expert n :

$$L_t^n(I_t) = \sum_{\tau=1}^t \lambda(\omega_\tau, \xi_\tau^n)$$

The online performance of the algorithm $s \in S_{D,t}$ for the given I_t is measured by the expected cumulative regret over t rounds:

$$R_t^D(s, I_t) = H_t^D(s, I_t) - \min_{n \in \mathcal{N}} L_t^n(I_t)$$

with respect to the best expert. Here $\lambda(\omega, \gamma) : \Omega \times \Gamma \rightarrow \mathbb{R}_+$ is some loss function, not necessary convex, Lipschitz or exponentially concave. The performance of the strategy s is

$$R_t^D(s) = \max_{I_t} R_t^D(s, I_t),$$

that is the maximal expected regret over all possible games I_t . The strategy $s_{D,t}^*$ that achieves the minimal regret

$$s_{D,t}^* = \arg \min_{s_D \in S_{D,t}} R_t^D(s)$$

is called optimal (may not be unique).

Theorem 4 *For the given Ω_T, Ξ_T , forecasting horizon D and game length T (such that T is a multiple of D) we have*

$$R_T^D(s_{D,T}) \geq D \cdot R_{T/D}^1(s_{1,T/D}).$$

Proof Let $s = s_{D,T}^* \in S_{D,T}$ be the optimal strategy. We define new one-step ahead forecasting strategy $s'' \in S_{1,T}$ based on s . Let

$$\pi_t^{s''} = \frac{1}{D} \sum_{\tau=\lceil \frac{t}{D} \rceil D+1}^{\lceil \frac{t}{D} \rceil D+D} \pi_\tau^s = \frac{1}{D} \sum_{\tau=\lceil \frac{t}{D} \rceil D+1}^{\lceil \frac{t}{D} \rceil D+D} \pi^s(\gamma | I_{\tau-D}).$$

For every one-step ahead forecasting game $I'_{T/D} = (\Omega'_{T/D}, \Xi'_{T/D})$ we create a new D -step ahead forecasting game $I''_T = (\Omega''_T, \Xi''_T)$. We set $\omega_t'' = \omega'_{[t/D]+1}$ and $(\xi_t^n)'' = (\xi_{[t/D]+1}^n)'$ for all $t = 1, 2, \dots, T$ and $n \in \mathcal{N}$.

The last step is to define one-step ahead forecasting strategy s' for one-step ahead forecasting game $I'_{T/D}$. We set $\pi_t^{s'} = \pi_{(t-1)D+1}^{s''}$.

We compute the loss of the algorithm s on $I''_{T/D}$.

$$\begin{aligned}
 H_T^D(s, I''_t) &= \sum_{t=1}^T \mathbb{E}_{\pi_t^s} [\lambda(\omega_t'', \gamma)] = \\
 \sum_{t=1}^T \mathbb{E}_{\pi_t^s} [\lambda(\omega'_{[t/D]+1}, \gamma)] &= \sum_{t=1}^{T/D} \left[\sum_{\tau=1}^D \mathbb{E}_{\pi_{(t-1)D+\tau}^s} [\lambda(\omega_\tau'', \gamma)] \right] = \tag{16}
 \end{aligned}$$

$$\begin{aligned}
 \sum_{t=1}^{T/D} \left[D \cdot \mathbb{E}_{\pi_{(t-1)D+1}^{s''}} [\lambda(\omega_t'', \gamma)] \right] &= D \sum_{t=1}^{T/D} \left[\mathbb{E}_{\pi_{(t-1)D+1}^{s''}} [\lambda(\omega_t'', \gamma)] \right] = \tag{17} \\
 &= D \sum_{t=1}^{T/D} \left[\mathbb{E}_{\pi_t^{s''}} [\lambda(\omega_t', \gamma)] \right] = D \cdot H_{T/D}^1(s', I'_{T/D}). \tag{18}
 \end{aligned}$$

In line (16) we split the sum into T/D sequential groups of losses of length D . In transition from (16) to (17) we note that the strategy $\pi^{s''}$ provides the same loss within each group. In transition from (17) to (18) we use the definition of $\pi^{s''}$. Also we note that

$$L_T^n(I''_T) = D \cdot L_{T/D}^n(I'_{T/D}). \tag{19}$$

Thus, for every $I'_{T/D}$ we have

$$R_T^D(s, I''_t) = D \cdot R_{T/D}^1(s', I'_{T/D}).$$

According to the definition of the minimax regret, for the one-step forecasting game of length T/D there exists such $I'_{T/D}$ that

$$R_{T/D}^1(s', I'_{T/D}) \geq R_{T/D}^1(s_{1,T/D})$$

For $I'_{T/D}$ and the corresponding sequence we obtain

$$R_T^D(s) \geq R_T^D(s, I''_t) = D \cdot R_{T/D}^1(s', I'_{T/D}) \geq D \cdot R_{T/D}^1(s_{1,T/D})$$

This ends the proof. ■

In fact, from Theorem 4 we can conclude that an optimal aggregating algorithm A_D^* for the long-term forecasting with experts' advice and competing with the best expert can be obtained by a simple replicating technique from the optimal algorithm A_1^* for the 1-step ahead forecasting and competing with the best expert.

However, if the goal is not to compete with the best expert (but, for example, to compete with the best alternating sequence of experts with no more than K switches or some other limited sequence), this theorem may also work. All the computations still remain true in the general case, except for (19). This equality should be replaced (if possible) by the analogue.