Proceedings of Machine Learning Research 128:1–18, 2020 Conformal and Probabilistic Prediction and Applications

Mixing Past Predictions

Alexander Korotin A.KOROTIN@SKOLTECH.RU
Skolkovo Institute of Science and Technology, Moscow, Russia

Vladimir V'yugin VYUGIN@IITP.RU
Institute for Information Transmission Problems, Moscow, Russia
Skolkovo Institute of Science and Technology, Moscow, Russia

Evgeny Burnaev E.BURNAEV@SKOLTECH.RU
Skolkovo Institute of Science and Technology, Moscow, Russia

Editor: Alexander Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgueni Smirnov and Giovanni Cherubin

Abstract

In the framework of the theory of prediction with expert advice, we present an algorithm for online aggregation of the functional predictions. The approach implies that at each time step some algorithm issues a forecast in the form of a function and then the master algorithm combines these current and past functional forecasts into one aggregated functional forecast. We apply the proposed algorithm for the problem of long-term predictions of time series. By combining the past and current long-term functional forecasts, we obtain a smoothing mechanism that protects our algorithm from temporary changes in the trend of time series, noise and outliers. To evaluate the performance of presented aggregating algorithm as a long-term forecaster we use a new "integral" loss function and the delayed feedback approach. We apply this algorithm for the regression problems, we present some method for smoothing regression forecasts.

Keywords: Functional forecasts, Integral loss function, Mixable loss function, Aggregating algorithm, Long-term online forecasting, Prediction with expert advice, Online smoothing regression

1. Introduction

In this paper, we propose a method for online aggregation of a dynamically growing set of forecasting models. Let some algorithm periodically generate forecasting models, which are later compared with observations obtained online. At any time point, the performance of any such model is measured by cumulative loss suffered by that time. At any time moment, we build an aggregating model that has the best performance compared to any of such models (up to some regret).

We apply the method for the long-term forecasting of time series. The problem of long-term forecasting of time series is of high practical importance. Many classical (ARMA, ARIMA Box et al. 2015) and recent (e.g. Facebook Prophet[1]) time series forecasting approaches produce a model that is capable of predicting arbitrarily many time steps ahead.

1. https://github.com/facebook/prophet

More precisely, we consider any long-term forecast as a function that assigns to each future time moment the corresponding value of the time series.

The task of the learning algorithm is to combine in online regime all available predictions into one aggregated functional long-term prediction. We solve this problem in the framework of the theory of prediction with expert advice, we present an algorithm for online aggregation of the functional predictions. The approach implies that at each moment of time, we obtain a set of predictions in the form of functions defined on the same domain, and then the master algorithm combines these current and past forecasts into one aggregated functional forecast. To evaluate the performance of the presented aggregating algorithm as a long-term forecaster, we use a new "integral" loss function, which measures the discrepancy between functional forecasts.

We apply the proposed algorithm to the problem of long-term predictions of time series. The advantage of this approach is that when building the final forecast at each time step t for any interval ahead, one may use forecasts made earlier at the steps $t' < t$. Forecasts of each step $t' < t$ are made using less of the observed data. Nevertheless, they can be more robust to noise, outliers and novelty of the time interval $[t' + 1, t]$. Thus, the usage of such outdated forecasts may prove useful, especially if time series is stationary.

The first problem we state in the paper is the effective usage of the outdated forecasts. Formally, the learner is given a basic forecasting algorithm. This algorithm at every step t produces potentially infinite forecast for the steps $t + 1, t + 2, \ldots$ ahead in the form of a function from all time moments ahead. The goal of the learner at each time step t is to combine the current forecast and the forecasts made earlier into one aggregated long-term forecast for the time moments $t + 1, t + 2, \ldots$ ahead. We develop an algorithm which to efficiently combines these forecasts.

It is worth noting that an other important problem of time series prediction is that some forecasting models use a limited memory. This means than only a fixed number of previous observations are considered to fit the model and build the next forecast. For example, regression methods that use a rolling window have this restriction. The usage of limited history saves computational resources and may catch changing dependencies in data, but it also can decrease the accuracy of the model. In order to partially compensate this disadvantage, we use the approach to combine past and current predictions made by the forecasting method.

In the rest part of out paper, we consider the online supervised learning scenario. The data is represented by pairs (x, y) of predictor-response variables. Instead of point or interval predictions, the experts and the learner present predictions in the form of functions $f(\mathbf{x})$ from signals $\mathbf{x}$. For example, in case of linear regression, $f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x})$, where $\mathbf{w} \in \mathcal{R}^n$ is a weight vector and $\mathbf{x} \in \mathcal{R}^n$ is an argument. Signals $\mathbf{x} = \mathbf{x}_t$ and forecasts $\mathbf{w}_t$ appear gradually over time t and allow to calculate forecasts as the values $f_t(\mathbf{x}) = (\mathbf{w}_t \cdot \mathbf{x})$ of these functions, where $\mathbf{x} \in \mathcal{R}$ is arbitrary. We combine these regression functions using the prediction with expert advice methods.

In general, we consider the game-theoretic on-line learning model in which a master (aggregating) algorithm has to combine predictions from a set of experts. The problem setting we investigated can be considered as the part of or Prediction with Expert Advice (PEA) framework (see e.g. Littlestone and Warmuth 1994; Freund and Schapire 1997; Vovk 1990, 1998; Cesa-Bianchi and Lugosi 2006 among others). In this framework the learner is usually

called the aggregating algorithm. The aggregating algorithm combines the predictions from a set of experts in the online mode during time steps $t = 1, 2, \ldots, T$.

Adamskiy et al. (2019) showed how Aggregating Algorithm could be applied to predict a vector of outcomes under the loss equal to the sum of coordinate losses. We generalize this result to the prediction of a function using the loss equal to the integral w.r.t. the function's argument (see also Korotin et al. 2019).

In this paper, we use the general notion of integral mixability (and exp-concavity). We consider the online scenario to predict the function: at each step t the aggregating algorithm receive (past) forecasts – the functions $f_i(x)$ issued at steps $i \leq t$, and has to combine these functions into a single forecast – a function $\gamma_t(x)$. The true output is a function $y_t(x)$. For the function-valued forecasting it is reasonable to measure loss via integral loss functions which naturally arise from loss functions used for comparing one-dimensional outcomes, i.e. $\lambda(\gamma_t(\cdot), y(\cdot)) = \int \lambda(\gamma_t(x), y(x))u(x)dx$, where $u(x)$ is a density function. The integral function is mixable if the basic loss function is mixable. The definition of integral mixability is a generalization of the notion of vector mixability which was introduced by Adamskiy et al. (2019).

A natural example of mixable integral loss function is Continuous Ranked Probability Score (CRPS)

$$\mathrm{CRPS}(F, y) = \int (F(x) - H(x - y))^2 dx,$$

where $F(x)$ is a cumulative probability distribution function and $H(x)$ is the Heaviside function: $H(x) = 0$ for $x < 0$ and $H(x) = 1$ for $x \geq 0$ (Epstein 1969, Matheson and Winkler 1976, Bröcker 2012, etc). The integral mixability property of CRPS was proved by V'yugin and Trunov (2019) and further studied by Dzhamtyrova and Kalnishkan (2019).

In practice for time series prediction, the square loss function is widely used. The square loss function is mixable (see Vovk 1998). Thus, for our goal the Vovk's aggregating algorithm (AA) is the most appropriate, since it has theoretically best performance among all known algorithms for mixable losses. We use the aggregating algorithm as the base and modify it for the ahead long-term interval forecasting with the usage of additional outdated forecasts.

The long-term forecasting considered in this paper is a case of the forecasting with a delayed feedback. As far as we know, the problem of the delayed feedback forecasting was first considered by Weinberger and Ordentlich (2002).

The article is structured as follows. In Section 2, we give some preliminary notions. In Section 3, we introduce the notion of integral loss function and study the property of its mixability. In Section 4, we present the algorithm for combining long-term past forecasts. Theorem 3 presents a performance bound for regret of this algorithm for adversarial case. In Section 5, we apply PEA approach for a case of the online supervised learning and the algorithm from Section 4 for online smoothing regression. Also, we provide experiments conducted on synthetic data and show the effectiveness of the proposed method.

2. Preliminaries

In this section, we present the main notions which will be used in further analysis.

2.1. Prediction with expert advice

In this section, we recall the main ideas of prediction with expert advice theory. Let Ω be a set of outcomes and Γ be a set of all forecasts, $\lambda(\gamma, y)$ be a loss function, $\gamma \in \Gamma$, $y \in \Omega$. Let also, a pool $\mathcal{N}$ of the experts be given. We assume that $\mathcal{N} = \{1, 2, \dots\}$ is a countable set.

Suppose that outcomes $y_1, y_2, \dots \in \Omega$ are revealed online – step by step. Learning proceeds in steps $t = 1, 2, \dots,$. At each time moment t experts $i \in \mathcal{N}$ present their predictions $f_{i,t} \in \Gamma$ and the aggregating algorithm presents its own forecast $\gamma_t \in \Gamma$. When the corresponding outcome y_t is revealed, the experts suffer their losses $l_{i,t} = \lambda(f_{i,t}, y_t)$, $i \in \mathcal{N}$, and the aggregating algorithm suffers a $h_t = \lambda(\gamma_t, y_t)$.

Protocol 1

FOR $t = 1, \dots, T$

1. Receive the forecasts $f_{i,t}$ of the experts $i \in \mathcal{N}$.
2. Present the forecast γ_t of the learner.
3. Observe the true outcome y_t and compute the losses $l_{i,t} = \lambda(f_{i,t}, y_t)$ of the experts and the loss $h_t = \lambda(\gamma_t, y_t)$ of the learner.

ENDFOR

The cumulative loss $L_{i,T}$ suffered by any expert i and the loss H_T suffered by the learner during T steps are defined by $L_{i,T} = \sum_{t=1}^{T} l_{i,t}$ and $H_T = \sum_{t=1}^{T} h_t$ respectively. The performance of the algorithm with respect to an expert i is measured by the regret $R_{i,T} = H_T - L_{i,T}$.

The goal of the aggregating algorithm is to minimize the regret with respect to each expert. In order to achieve this goal, at each time moment t, the aggregating algorithm evaluates performance of the experts in the form of theirs weights $\mathbf{w}_t = \{w_{i,t} : i \in \mathcal{N}\}$, where $w_{i,t} \geq 0$ for all i and t. For example, we set initial weights $w_{i,1} = \frac{1}{i(i+1)}$ for $i \in \mathcal{N}$. The weight $w_{i,t}$ of an expert i is an estimate of the quality of the expert predictions at steps $\leq t$. In the classical setting (see Freund and Schapire (1997), Vovk (1990) among others), the process of experts' weights updating is based on the method of exponential weighting with a learning rate $\eta > 0$:

$$w_{i,t+1} = w_{i,t} e^{-\eta l_{i,t}} \tag{1}$$

for every $i \in \mathcal{N}$. The normalized weights are defined as

$$w^*_{i,t} = \frac{w_{i,t}}{\sum_{j \in \mathcal{N}} w_{j,t}}.$$

2.2. Aggregating algorithm

The aggregating algorithm (AA) by Vovk (1990), Vovk (1998) is the base algorithm in our study. Let us explain the main ideas of learning with AA.

We consider the learning with a mixable loss function $\lambda(\gamma, y)$ Here y is an outcome and γ is a forecast. Each expert i presents forecast c_i. In this case, the main tool is a superprediction function

$$g(y) = -\frac{1}{\eta} \ln \sum_{i \in \mathcal{N}} e^{-\eta \lambda(c_i, y)} p_i,$$

where $\mathbf{p} = (p_i : i \in \mathcal{N})$ is a probability distribution on the set of all experts and $\mathbf{c} = (c_i : i \in \mathcal{N})$ is a sequence of the experts predictions.

A loss function λ is η-mixable if for any probability distribution $\mathbf{p}$ on the set of experts and for any set of experts predictions $\mathbf{c}$ a forecast γ exists such that

$$\lambda(\gamma, y) \le g(y) \tag{2}$$

for all y.

A closely related, but the more narrow notion, is exponentially concavity. A loss function $\lambda(\gamma, y)$ is η-exponential concave if for any y the function $e^{-\eta \lambda(\gamma, y)}$ is concave with respect to γ. By definition for any η-exponential concave function inequality (2) holds, i.e., any η-exponential concave function is η-mixable.

We fix some rule $\gamma = \mathrm{Subst}(\mathbf{c}, \mathbf{p})$ for computing a forecast satisfying (2). Subst is called a substitution function.

The square loss function $\lambda(\gamma, y) = (y - \gamma)^2$ is η-mixable for any η such that $0 < \eta \le \frac{2}{(b-a)^2}$, where y and γ are a real numbers and $y \in [a, b]$ for some $a < b$, see Vovk (1990), Vovk (1998)). By Vovk (1998) and Vovk (2001), for the square loss function, the corresponding forecast can be defined as

$$\gamma = \mathrm{Subst}(\mathbf{c}, \mathbf{p}) = \frac{a+b}{2} + \frac{1}{2\eta(b-a)} \ln \frac{\sum\limits_{i \in \mathcal{N}} p_i e^{-\eta(b-c_i)^2}}{\sum\limits_{i \in \mathcal{N}} p_i e^{-\eta(a-c_i)^2}}. \tag{3}$$

For the η-exponential concave loss function we can also use a more straightforward expression for the substitution function:

$$\gamma = \mathrm{Subst}(\mathbf{c}, \mathbf{p}) = \sum_{i \in \mathcal{N}} c_i p_i. \tag{4}$$

The square loss function is η-exponential concave for $0 < \eta \le \frac{1}{2(b-a)^2}$. However, the definition (4) results in four times more regret bound (see Kivinen and Warmuth 1999). Inequality (2) also holds for all y.

2.3. Regret analysis for AA

The performance bound of AA is given by the following theorem (see Vovk 1998).

Proposition 1 *Assume that a loss function $\lambda(f, y)$ is η-mixable. Let H_T be the cumulated loss of the learner and $L_{i,T}$ be the cumulated loss of an expert i. Then for any i,*

$$H_T \le L_{i,T} + \frac{1}{\eta} \ln \frac{1}{w_{i,1}}$$

for every T.

Proof Let $\mathbf{w}_t^* = (w_{i,t}^* : i \in \mathcal{N})$ be the normalized weights and $\mathbf{f}_t = (f_{i,t} : i \in \mathcal{N})$ be the forecasts of the experts at step t. The learner's forecast is defined $f_t = \mathrm{Subst}(\mathbf{f}_t, \mathbf{w}_t^*)$. By mixability property

$$h_t = \lambda(f_t, y_t) \le g_t(y_t) = -\frac{1}{\eta} \ln \sum_{i \in \mathcal{N}} e^{-\eta \lambda(f_t, y_t)} w_{i,t}^* = -\frac{1}{\eta} \ln \frac{W_{t+1}}{W_t}$$

for all t, where $W_t = \sum\limits_{i \in \mathcal{N}} w_{i,t}$ and $W_1 = 1$. From (1) we have $w_{i,T+1} = w_{i,1} e^{-\eta L_{i,T}}$. By telescoping, we obtain for any expert i the time-independent bound $H_T \le \sum\limits_{t=1}^{T} g_t(y_t) = -\frac{1}{\eta} \ln W_{T+1} \le L_{i,T} + \frac{1}{\eta} \ln \frac{1}{w_{i,1}}$ for all T. ■

2.4. AA for delayed feedback

In what follows, we will evaluate the predictive performance of the constructed algorithms as long-term predictors. At each step t of the game the algorithm makes a decision, and its result will be revealed only at the time point $t + d$, where d is some delay. Only at this moment will we be able to estimate the losses from this long-term prediction. To do this, we consider a generalization of online learning to handle delays in receiving feedback to the given prediction.

There exists a bunch of meta-algorithms that allow to produce a version for delayed feedback setting from the basic non-delayed version, see Weinberger and Ordentlich (2002) and further developed by Langford et al. (2009), Mesterharm (2007), and Mesterharm (2009). The authors studied the setting under fixed known feedback delay d. They proved that the optimal (non-adaptive) algorithm is to run d independent versions of the optimal non-delayed algorithm on d disjoint time grids $G_k = \{t : t = dj + k, j = 0, 1, \dots\}$, where $k = 0, 1, \dots, d-1$. Thus, the optimal worst case adversarial regret for AA with respect to any expert i is bounded by the sum of the regret bounds for AA on the d grids: $H_T \le L_{i,T} + \frac{d}{\eta} \ln \frac{1}{w_{i,1}}$ for all T. We will use this approach in Section 4.

3. Learning with functional forecasts

Consider a formulation in which some device observes results of a physical process and presents at any time step t a forecasting model for this process. Let $\mathcal{X}$ be a set supplemented by the corresponding algebra of Borel sets. By a forecasting model we mean a measurable function $f : \mathcal{X} \to \mathcal{R}_+$, which, with an input value of $x \in \mathcal{X}$, yields a real number $f(x)$ (the model's output at x).

We consider the problem of online learning (according to Protocol 1) in which an each round t we get a set of current versions of models $f_{1,t}(x), f_{2,t}(x), \dots$ and build the current aggregation model $\gamma_t(x)$. After that, as a result of measurements, we obtain (maybe with some delay) more or less accurate (surrogate) approximations $y_t(x)$ of the background physical process and calculate the discrepancy $\boldsymbol{\lambda}(f_i(\cdot), y_t(\cdot))$ between any model f_i and this approximation. We compute also the discrepancy $\boldsymbol{\lambda}(\gamma_t(\cdot), y_t(\cdot))$ for the aggregated fore-

cast γ_t. Here $\boldsymbol{\lambda}$ is some (integral) loss function which can compute discrepancy between functional models.

We can construct at each step t an optimal functional model $\boldsymbol{\gamma}_t(\cdot)$ by observing the cumulative losses suffered by functional models obtained in the past. For this we will use the ideas of prediction with expert advice theory and the concept of integral loss function. This notion develops the concepts of generalized loss function for vector forecasts introduced by Adamskiy et al. (2019) and CRPS loss function considered by V'yugin and Trunov (2019).

Let Γ, Ω and $\mathcal{X}$ be measurable spaces. Assume that $\lambda : \Gamma \times \Omega \to \mathbb{R}_+$ is a measurable loss function. We refer to this loss function as to basic loss function.

Let u be some non-negative measurable function satisfying $\int_{\mathcal{X}} u(x)dx = 1$. By integral loss function we mean a function $\boldsymbol{\lambda} : \Gamma^{\mathcal{X}} \times \Omega^{\mathcal{X}} \to \mathbb{R}_+$ defined by

$$\boldsymbol{\lambda}(\gamma, y) = \int_{\mathcal{X}} \lambda(\gamma(x), y(x))u(x)dx. \tag{5}$$

Consider a countable pool $\mathcal{N}$ of experts. Assume that an η-mixable loss function $\lambda(f, y)$ be given and $\mathrm{Subst}(\mathbf{f}, \mathbf{p})$ be the corresponding substitution function, where $f \in \Gamma$ and $y \in \Omega$.

Theorem 2 *(Korotin et al. 2019) If a basic loss function is η-mixable then the corresponding integral loss function is also η-mixable. The aggregated forecast can be computed pointwise as $\gamma(x) = \mathrm{Subst}(\mathbf{f}(x), \mathbf{p})$, where $\mathbf{f}(x) = (f_i(x) : i = 1, 2, \dots)$ be a sequence of (measurable) forecast functions and $\mathbf{p}$ be a probability distribution on the set of all experts.*[2]

Proof By η-mixability property of the basic function, for a distribution $\mathbf{p} \in \mathcal{P}(\mathcal{N})$ for any $x \in \mathcal{X}$ a forecast $\gamma(x) = \mathrm{Subst}(\mathbf{f}(x), \mathbf{p})$ exists such that

$$\mathbf{e}^{-\eta\lambda(\gamma(x), y(x))} \geq \sum_{i \in \mathcal{N}} p_i e^{-\eta\lambda(f_i(x), y(x))}$$

holds for every function $y(\cdot) \in \Omega^{\mathcal{X}}$. Taking the logarithm of both sides of the inequality we obtain

$$-\eta\lambda(\gamma(x), y(x)) \geq \ln \sum_{i \in \mathcal{N}} p_i e^{-\eta\lambda(f_i(x), y(x))}$$

for any $x \in \mathcal{X}$. Multiply both sides by $u(x) \geq 0$ and integrate over $x \in \mathcal{X}$:

$$\int_{\mathcal{X}} (-\eta\lambda(\gamma(x), y(x))u(x)dx \geq \int_{\mathcal{X}} \ln \sum_{i \in \mathcal{N}} p_i e^{-\eta\lambda(f_i(x), y(x))} u(x)dx \tag{6}$$

The left part of inequality (6) equals to $-\eta\boldsymbol{\lambda}(\gamma, y)$. Next, for $x \in \mathcal{X}$ and $i \in \mathcal{N}$ define

$$f(x, i) = e^{-\eta\lambda(f_i(x), y(x))}.$$

By applying the notation change and taking the exponent of both sides of (6), we obtain

$$e^{-\eta\boldsymbol{\lambda}(\gamma, y)} \geq e^{\int_{\mathcal{X}} \ln \sum_{i \in \mathcal{N}} p_i \cdot f(x,i)u(x)dx} \tag{7}$$

2. We assume that the function $\mathrm{Subst}(\mathbf{f}(x), \mathbf{p})$ is measurable.

The final step is to apply Continuous Form of Holder Inequality (8) by Dunford and Schwartz (1958) (see also Nikolova et al. 2017)

$$\int_{\mathcal{Y}} e^{\int_{\mathcal{X}} \ln f(x,y)u(x)dx} v(y)dy \le e^{\int_{\mathcal{X}} \ln\left(\int_{\mathcal{Y}} f(x,y)v(y)dy\right)u(x)dx}, \tag{8}$$

where $f(x,y)$ is positive and measurable on $\mathcal{X} \times \mathcal{Y}$ function, and $u(x)$ and $v(y)$ are weight functions and $\int_{\mathcal{X}} u(x)dx = 1$.

We set in (8) $\mathcal{Y} = \mathcal{N} = \{1, 2, \dots\}$, $v(i) = p_i$ and obtain

$$e^{\int_{\mathcal{X}} \ln \sum_{i\in\mathcal{N}} p_i \cdot f(x,i)u(x)dx} \ge \sum_{i\in\mathcal{N}} p_i e^{\int_{\mathcal{X}} \ln f(x,i)u(x)dx} =$$
$$\sum_{i\in\mathcal{N}} p_i e^{-\eta \int_{\mathcal{X}} \lambda(f_i(x),y(x))u(x)dx} = \sum_{i\in\mathcal{N}} p_i e^{-\eta\boldsymbol{\lambda}(f_i,y)} \tag{9}$$

Now we combine (9) with (7) and obtain the desired inequality

$$e^{-\eta\boldsymbol{\lambda}(\gamma,y)} \ge \sum_{i\in\mathcal{N}} p_i e^{-\eta\boldsymbol{\lambda}(f_i,y)}. \tag{10}$$

for every $y \in \Omega^{\mathcal{X}}$. ■

Let us specify Protocol 1 for the case of learning with integral loss function, which is based on an η-mixable loss function $\lambda(f,y)$, where $f \in \Gamma$ and $y \in \Omega$.

Protocol 1a

Define the initial weights $\mathbf{w}_1 = (w_{i,1} : i \in \mathcal{N})$ of the experts.

FOR $t = 1, \dots, T$

1. Receive the experts' predictions $f_{i,t}(x)$ for all $i \in \mathcal{N}$ and a density function $u_t(x)$.
2. Present the learner's forecast $\gamma_t(x) = \text{Subst}(\mathbf{f}_t(x), \mathbf{w}_t^*)$, where $\text{Subst}(\mathbf{f}, \mathbf{p})$ is a substitution function for $\lambda(f,y)$ and $\mathbf{f}_t(x) = (f_{i,t}(x) : i \in \mathcal{N})$, $\mathbf{w}_t^* = (w_{i,t}^* : i \in \mathcal{N})$ are normalized weights.
3. Observe the true outcome $y_t(x)$ and compute the losses $l_{i,t} = \int \lambda(f_{i,t}(x), y_t(x))u_t(x)dx$ of the experts and the loss $h_t = \int \lambda(\gamma_t(x), y_t(x))u_t(x)dx$ of the learner.
4. Update weights: $w_{i,t+1} = w_{i,t}e^{-\eta l_{i,t}}$ for $i \in \mathcal{N}$.

ENDFOR

The corresponding performance bound is given by Proposition 1.

3.1. Continuous Ranked Probability Score (CRPS)

A typical application of learning with functional forecasts is learning a probability distribution function using Continuous Ranked Probability Score (CRPS) as a loss function (Epstein 1969, Matheson and Winkler 1976, Bröcker 2012, etc).

Let in Protocol 1a the set of outcomes be the real line for some $a < b$ and the set of forecasts Γ be a set of all probability distribution functions F. [3]

The quality of the prediction F in view of the actual outcome y is often measured by the continuous ranked probability score (loss function)

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{+\infty} (F(x) - H(x - y))^2 u(x) dx, \tag{11}$$

where $H(x)$ is the Heaviside function: $H(x) = 0$ for $x < 0$ and $H(x) = 1$ for $x \geq 0$.

The CRPS score measures the difference between the forecast $F(x)$ and a perfect forecast $H(x - y)$ which puts all mass on the verification y. The lowest possible value 0 is attained when F is concentrated at y, and in all other cases $\mathrm{CRPS}(F, y)$ will be positive.

For simplicity, we consider integration over a finite interval $[a, b]$, where $a < b$, with the uniform density $u(x) = \frac{1}{b-a}$ if $x \in [a, b]$ and $u(x) = 0$ otherwise. Also, $F(a) = 0$ and $F(b) = 1$ for every F.

$$\mathrm{CRPS}(F, y) = \frac{1}{b - a} \int_a^b (F(u) - H(u - y))^2 du. \tag{12}$$

The definition (12) is a special case of definition (11) (up to a factor), where $u(x) = \frac{1}{b-a}$ for $x \in [a, b]$ and $u(x) = 0$ otherwise.

By Theorem 2 and analysis of Section 2 the function (12) is η-mixable for $0 < \eta \leq 2$ and η-exponentially concave for $0 < \eta \leq \frac{1}{2}$.

By (3), where $\eta = 2$ and $[a, b] = [0, 1]$, the corresponding learner's forecast – the probability distribution function $F_t(x)$ given the probability distribution functions $F_{i,t}(x)$ presented by the experts $1 \leq i \leq N$ can computed in the closed form:

$$F_t(x) = \frac{1}{2} - \frac{1}{4} \ln \frac{\sum_{i=1}^N w_{i,t}^* e^{-2(F_{i,t}(x))^2}}{\sum_{i=1}^N w_{i,t}^* e^{-2(1-F_{i,t}(u))^2}}, \tag{13}$$

where $\eta = 2$ and $w_{i,t}^* = \frac{w_{i,t}}{\sum_{j=1}^N w_{j,t}}$ – the normalized weights of the experts (see V'yugin and Trunov 2019).[4]

A natural generalization of CRPS arises if we replace the Heaviside function with an empirical probability distribution function. Several other examples of mixable integral loss functions are given by Korotin et al. (2019).

3. A probability distribution function is a non-decreasing function $F(y)$ defined on this interval such that $\lim_{x \to -\infty} F(x) = 0$ and $\lim_{x \to \infty} F(x) = 1$. Also, it is left-continuous and has the right limit at each point.
4. It is easy to verify that $F_t(x)$ is indeed a probability distribution function.

4. Mixing long-term forecasts with confidences

In this section, we present an algorithm for mixing past long-term predictions. We use the delayed feedback approach and the proposed integral generalization of the loss function to evaluate the predictive ability of long-term forecasting methods.

The learning process is as follows. There is some unknown source generating sequentially the outcomes $y_1, y_2, \dots$. At each moment of time t, some (basic) forecasting algorithm observing the outcomes $y_1, \dots, y_t$ outputs a function f_t whose values $f_t(s)$ are predictions of the future outputs of this source at time moments $s = t+1, t+2, \dots$; we set $f_t(s) = y_s$ for $s \le t$.

At any step t we observe also the forecasting functions $f_1(s), \dots, f_{t-1}(s)$ that were issued by the basic algorithm at the previous steps. Any such function f_i, $i < t$, outputs a forecast $f_i(s)$ at each point $s \ge i+1$ and $f_i(s) = y_s$ for $s \le i$.

Assume that a loss function $\lambda(\gamma, y)$ be given; we suppose that this function is η-mixable for some $\eta > 0$.

The goal of the learner is to aggregate these long-term forecasts $f_1(s), \dots, f_t(s)$ into one long-term forecast $\gamma_t(s)$ for $s = t+1, t+2, \dots$; we set also $\gamma_t(s) = y_s$ for $s \le t$.

4.1. Confidence values

Let at any step t, each prediction function $f_i(s)$ be supplemented by a confidence value $p_{i,t}$, where $0 \le p_{i,t} \le 1$ if $i \le t$ and $p_{i,t} = 0$ for $i > t$. If $p_{i,t} < 1$, then this means that at step t we use the forecast $f_i(s)$ only partially (e.g. it may become obsolete with time). If $p_{i,t} = 0$ then the corresponding forecasting function is not taken into account at all.[5] Confidence values can be set by the basic algorithm or by the learner.[6]

Assume that the learner's forecast $\gamma_t(s)$ is known. Consider the auxiliary experts $i = 1, 2, \dots$ and define theirs forecasts at any step t: $f_{i,t}(s) = f_i(s)$ if $i \le t$, and $f_{i,t}(s) = \gamma_t(s)$ for every s if $i > t$. For any t and s, define the random forecasts of these experts

$$\tilde{f}_{i,t}(s) = \begin{cases} f_{i,t}(s) \text{ with probability } p_{i,t}, \\ \gamma_t(s) \text{ with probability } 1 - p_{i,t}. \end{cases}$$

At any time moment t, the level of confidence can be interpreted as the probability distribution $\mathbf{p}_{i,t} = (p_{i,t}, 1 - p_{i,t})$ on a two element set, where $p_{i,t}$ is the probability for the expert i to follow its own prediction and $1 - p_{i,t}$ is the probability to follow the prediction of the aggregating algorithm. We will consider the mean loss

$$E_{\mathbf{p}_{i,t}}[\lambda(\tilde{f}_{i,t}(s), y)] = p_{i,t}\lambda(f_{i,t}(s), y) + (1 - p_{i,t})\lambda(\gamma_t(s), y), \tag{14}$$

where y is an outcome and $E_{\mathbf{p}_{i,t}}$ denotes the mathematical expectation with respect to the probability distribution $\mathbf{p}_{i,t}$.

5. For example, in applications, it is convenient for some k to set $p_{i,t} = 0$ if $t > i + k$, since too old predictions become obsolete.

6. The setting of prediction with experts that use the confidences as numbers in the interval $[0, 1]$ was first studied by Blum and Mansour (2007) and further developed by Cesa-Bianchi et al. (2007) and Gaillard et al. (2014).

Let a positive integer number d be given. At each time moment t we want to evaluate the ability of the learner and of the experts i, where $1 \le i \le t-d$, to make forecasts over d time points $t-d+1, \dots, t$ ahead. To do this, we define the corresponding integral losses.

Define the outcome function $y_{t-d}(s)$, where $y_{t-d}(s) = y_{t-d+s}$ if $1 \le s \le d$ and $y_{t-d}(s)$ be arbitrary otherwise.

Let $u_{t-d}(s)$ be a density on the time line such that $u_{t-d}(s) = 0$ for $s \le t-d$ and $s > t$.[7] At any step t, the learner suffers the (integral) loss

$$h_t = \int \lambda(\gamma_{t-d}(s), y_{t-d}(s)) u_{t-d}(s) ds \tag{15}$$

and each expert $1 \le i \le t-d$ suffers the loss

$$l_{i,t} = \int E_{\mathbf{p}_{i,t}}[\lambda(\tilde{f}_{i,t}(s), y_{t-d})] u_{t-d}(s) ds$$

on the time interval $[t-d+1, t]$. Define also, $l_{i,t} = h_t$ for $i > t-d$. By (14) this quantity can be represented as

$$l_{i,t} = \int (p_{i,t}\lambda(f_i(s), y_{t-d}(s)) + (1-p_{i,t})\lambda(\gamma_{t-d}(s), y_{t-d}(s))) u_{t-d}(s) ds. \tag{16}$$

By (15) and (16) at any step $t > d$ the regret $r_{i,t} = h_t - l_{i,t}$ with respect to the ith prediction for $1 \le i \le t-d$ and $t > d$ can be represented as

$$r_{i,t} = \int p_{i,t}(\lambda(\gamma_{t-d}(s), y_{t-d}(s)) - \lambda(f_i(s), y_{t-d}(s))) u_{t-d}(s) ds. \tag{17}$$

By definition $h_t = l_{i,t} = 0$ for $t \le d$; set $r_{i,t} = 0$ for every $t \le d$ and all i. For $t > d$, $r_{i,t} = 0$ if $i > t-d$, since $l_{i,t} = h_t$ for these i by the definition.

The performance of Algorithm 2, which will be presented below, is measured by the cumulated regret $R_{i,T} = \sum_{t=1}^{T} r_{i,t}$ with respect to a long term forecast (an expert) i.

First, we provide justification of the auxiliary weights update rule in Algorithm 2. To exit the logical circle in the definition of $\tilde{f}_{i,t}(s)$, we will use the fixed point method by Chernov and Vovk (2009).

Our goal is to define for any $s > t$ a forecast $\gamma_t(s)$ such that

$$e^{-\eta\lambda(\gamma_t(s), y(s))} \ge \sum_{i=1}^{\infty} E_{\mathbf{p}_{i,t}}[e^{-\eta\lambda(\tilde{f}_{i,t}(s), y(s))}] w^*_{i,t} \tag{18}$$

for each function $y(s)$, where $w^*_{i,t}$ is the normalized weight of the ith forecast accumulated at previous steps.[8]

Since $f_{i,t}(s) = f_i(s)$ if $i \le t$ and $f_{i,t}(s) = \gamma_t(s)$, $p_{i,t} = 0$ if $i > t$, we can rewrite inequality (18) in a more detailed form: for any s,

$$e^{-\eta\lambda(\gamma_t(s), y(s))} \ge \sum_{i=1}^{t} p_{i,t} w^*_{i,t} e^{-\eta\lambda(f_i(s), y(s))} + e^{-\eta\lambda(\gamma_t(s), y(s))} \left(1 - \sum_{i=1}^{t} p_{i,t} w^*_{i,t}\right). \tag{19}$$

7. A natural example of such a density is $u_{t-d}(s) = \frac{1}{d}$ for $t-d+1 \le s \le t$ and $u_{t-d}(s) = 0$ for all other s.
8. The weight updating rule will be given by item 2 of Algorithm 2 below.

Therefore, inequality (18) is equivalent to the inequality

$$e^{-\eta\lambda(\gamma_t(s),y(s))} \geq \sum_{i=1}^{t} w_{i,t}^{p} e^{-\eta\lambda(f_i(s),y(s))}, \tag{20}$$

where

$$w_{i,t}^{p} = \frac{p_{i,t} w_{i,t}}{\sum_{j=1}^{t} p_{j,t} w_{j,t}}. \tag{21}$$

According to the AA rule we can define $\gamma_t(s) = \mathrm{Subst}(\mathbf{f}_t(s), \mathbf{w}_t^p)$. Then (20) and its equivalent (18) are valid. Here Subst is the substitution function, $\mathbf{w}_t^p = (w_{1,t}^p, \dots, w_{t,t}^p)$, and $\mathbf{f}_t(s) = (f_1(s), \dots, f_t(s))$ for all s.

4.2. Algorithm for mixing past predictions

Now, we present the protocol of the algorithm for mixing past predictions with confidence values.

Algorithm 2

Set initial weights $w_{i,t} = \frac{1}{i(i+1)}$ for $1 \leq t \leq d$ and $i = 1, 2, \dots$. Fix a positive integer number d.
FOR $t = 1, \dots, T$

IF $t \leq d$ **THEN** put $l_{i,t} = h_t = 0$ for all i. **ELSE**

Suffer losses and update weights of the experts

1. Observe the outcome function $y_{t-d}(s)$, where $y_{t-d}(s) = y_{t-d+s}$ if $t-d+1 \leq s \leq t$ and $y_{t-d}(s)$ be an arbitrary otherwise. Compute the loss of Algorithm 2 suffered at step t

$$h_t = \int \lambda(\gamma_{t-d}(s), y_{t-d}(s)) u_{t-d}(s) ds,$$

where $u_{t-d}(s)$ is a density on the time line such that $u_{t-d}(s) = 0$ for $s \leq t-d$ or $s > t$.[9] Compute the losses incurred at step t by the experts $1 \leq i \leq t-d$

$$l_{i,t} = \int (p_{i,t}\lambda(f_i(s), y_{t-d}(s)) + (1-p_{i,t})\lambda(\gamma_{t-d}(s), y_{t-d}(s))) u_{t-d}(s) ds.$$

Define also $l_{i,t} = h_t$ for $i > t-d$.

2. Update weights $w_{i,t} = w_{i,t-d} e^{-\eta l_{i,t}}$ for $1 \leq i < \infty$.

Compute the aggregated long-term forecast

3. Observe the long-term forecasts $\mathbf{f}_t(s) = (f_1(s), \dots, f_t(s))$ and the corresponding confidences $p_{i,t}$ for $1 \leq i \leq t$.

4. Compute the auxiliary weights of the experts $1 \leq i \leq t$:

$$w_{i,t}^{p} = \frac{p_{i,t} w_{i,t}}{\sum_{j=1}^{t} p_{j,t} w_{j,t}}. \tag{22}$$

9. An example of such a density is $u_{t-d}(s) = \frac{1}{d}$ for $t-d+1 \leq s \leq t$ and $u_{t-d}(s) = 0$ for all other s.

5. Compute pointwise the aggregated long-term forecast:

$$\gamma_t(s) = \mathrm{Subst}(\mathbf{f}_t(s), \mathbf{w}_t^p) \tag{23}$$

for $s = t+1, t+2, \ldots$, where $\mathbf{w}_t^p = (w_{1,t}^p, \ldots, w_{t,t}^p)$.
ENDFOR

We measure the performance of our algorithm by the regret $R_{i,T}$ with respect to any ith forecast.

Theorem 3 *For any i,*

$$\sum_{t=1}^{T} r_{i,t} \le \frac{d}{\eta} \ln(i(i+1)) \tag{24}$$

for all T, where $r_{i,t}$ is defined by (17) and conventions below it.

Proof . The inequality (18) holds for outcomes $y_t(s)$ and and for the forecasts $\gamma_t(s)$ for all $s \ge t+1$.

Let $u_t(s)$ be a density on the time interval $[t+1, t+d]$.

By convexity of the exponent the inequality (18) implies

$$e^{-\eta\lambda(\gamma_t(s), y_t(s))} \ge \sum_{i=1}^{\infty} e^{-\eta E_{\mathbf{p}_{i,t}}[\lambda(y_t(s), \tilde{f}_{i,t}(s))]} w_{i,t}^*. \tag{25}$$

for all s. We apply the generalized Hölder inequality (10) with the density $u_t(s)$ and obtain

$$e^{-\eta \int \lambda(\gamma_t(s), y_t(s)) u_t(s) ds} \ge \sum_{i=1}^{\infty} e^{-\eta \int E_{\mathbf{p}_{i,t}}[\lambda(\tilde{f}_{i,t}(s), y_t(s))] u_t(s) ds} w_{i,t}^*. \tag{26}$$

The inequality (26) can be rewritten as

$$e^{-\eta h_{t+d}} \ge \sum_{i=1}^{\infty} e^{-\eta l_{i,t+d}} w_{i,t}^*, \tag{27}$$

where $h_{t+d} = \int \lambda(\gamma_t(s), y_t(s)) u_t(s) ds$ is the (integral) loss of the aggregating algorithm and for any $i \le t$

$$l_{i,t+d} = \int E_{\mathbf{p}_{i,t}}[\lambda(\tilde{f}_{i,t}(s), y_t(s))] u_t(s) ds =$$

$$\int (p_{i,t} \lambda(f_i(s), y_t(s)) + (1 - p_{i,t}) \lambda(\gamma_t(s), y_t(s))) u_t(s) ds$$

is the loss of the expert i suffered on the time interval $[t+1, t+d]$.

The sum from the inequality (24) can be split into d sums of regrets on pairwise disjoint grids: $\sum_{t=1}^{T} r_{i,t} = \sum_{k=0}^{d-1} \sum_{t \in G_k, t \le T} r_{i,t}$, where $G_k = \{t : t \equiv k (\mathrm{mod}\ d)\}$. Applying Proposition 1 and (27) to each grid separately, we obtain the bound (24). ■

The bound (24) implies that $\sup_i \sum_{t=1}^{T} r_{i,t} \le \frac{d}{\eta} \ln((T-d)(T-d+1))$ for each $T > d$.

5. Supervised setting: Mixing past regressors

In this section we apply Algorithm 2 to the online learning scenario within the supervised setting (that is, data are pairs $(\mathbf{x}, y)$ of predictor-response variables). At any step i a forecaster presents a regression function f defined on a set X of objects $\mathbf{x}$, which are called signals.

An example is a linear regression, where $X \subseteq \mathcal{R}^k$ is a set of k-dimensional vectors and a regression function is a linear function $f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x})$, where $\mathbf{w} \in \mathcal{R}^k$ is a weight vector and $\lambda(\gamma, y) = (\gamma - y)^2$ is the square loss. At any step t a forecasting function $f_t(\mathbf{x}) = (\mathbf{w}_t \cdot \mathbf{x})$ is constructed. So, at any step t we have a collection $\mathbf{f}_t = (f_1, \ldots, f_t)$ of past $t-1$ and the tth current forecasting functions (experts). We compute the forecasting function $\gamma_t(\mathbf{x})$ of the learner by the rule (23).

In the online mode with delayed feedback, at any step t, in item 1 of Algorithm 2 the learner observes the set of pairs of signals and outcomes $D_t = \{\mathbf{x}_{t-d+1}, y_{t-d+1}), \ldots (\mathbf{x}_t, y_t)\}$ and suffers the loss $h_t = \int \lambda(\gamma_{t-d}(\mathbf{x}), y_{t-d}(\mathbf{x}))u_{t-d}(\mathbf{x})d\mathbf{x}$, where $y_{t-d}(\mathbf{x}) = y$ if $(\mathbf{x}, y) \in D_t$ and is arbitrary for all other $\mathbf{x}$. Each prediction $f_i(\mathbf{x})$, where $1 \le i \le t$, also suffers the loss defined by (16). The corresponding density can be defined as $u_{t-d}(\mathbf{x}) = \frac{1}{|D_t|}$ if $\mathbf{x} \in D_t^1$ and $u_{t-d}(\mathbf{x}) = 0$ otherwise, where $D_t^1 = \{\mathbf{x}_{t-d+1}, \ldots, \mathbf{x}_t\}$.

The performance bound is presented by Theorem 3 and the inequality (24).

Experiment. Some time series show a strong dependence on the latest information instead of all the data. In this case, it is useful to apply regression with a rolling window. In this regard, we consider the application of Algorithm 2 for the case of online regression with a rolling window. The corresponding forecast represents some type of dependence between input and output data. If this relationship is relatively regular the corresponding forecast based on past data can successfully compete with forecasts based on the latest data. Therefore, it may be useful to aggregate the predictions of all forecasts based on past data.

Let $f_t(\mathbf{x}) = (\mathbf{w}_t \cdot \mathbf{x})$ be the ridge regression function, where $\mathbf{w}_t = (\sigma I + X_t' X_t)^{-1} X_t' \mathbf{y}_t$ for $t > h$. Here X_t is the matrix in which rows are formed by vectors $\mathbf{x}_{t-h}, \ldots, \mathbf{x}_{t-1} \in R^k$ (X_t' is the transposition of the matrix X_t), I is unit matrix, σ is a parameter, and $\mathbf{y}_t = (y_{t-h}, \ldots, y_{t-1})$. For $t \le h$ define $\mathbf{w}_t$ be equal to some fixed value.

We use the square loss function and assume that $y_t \in [-b, b]$ for all t. For each t we define the aggregating regression function γ_{t+1} (the learner forecast) by (28) using the regression functions f_i for $h < i \le t$, where each such a function is defined using a learning sample (a window) $(\mathbf{x}_{i-h}, y_{i-h}), \ldots, (\mathbf{x}_{i-1}, y_{i-1})$.

For the square loss $\lambda(\gamma, y) = (\gamma - y)^2$, where $y \in [-b, b]$, by (3) the learner forecast can be defined in the closed form:

$$\gamma_t(\mathbf{x}) = \frac{1}{4\eta b} \ln \frac{\sum_{i=1}^{t} w_{i,t}^p e^{-\eta((\mathbf{w}_i \cdot \mathbf{x}) - b)^2}}{\sum_{i=1}^{t} w_{i,t}^p e^{-\eta((\mathbf{w}_i \cdot \mathbf{x}) + b)^2}} \tag{28}$$

for each x or by the rule

$$\gamma_t(\mathbf{x}) = \left(\left(\sum_{i=1}^{t} w_{i,t}^p \mathbf{w}_t \right) \cdot \mathbf{x} \right), \tag{29}$$

where the weights $w^p_{i,t}$ are defined by (22).[10]

Let us discuss the details and results of experiments for online regression with a rolling window which were performed on synthetic data. The initial data was obtained as a result of sampling from a data generative model.

We start from a sequence $\mathbf{x}_1, \dots, \mathbf{x}_T$ of 20-dimensional signals sampled i.i.d from the multidimensional standard normal distribution. The signals are revealed online and $T = 3000$.

The target variable y is generated as follows. First, three random linear dependencies are generated, i.e. three weights vectors $\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, \hat{\mathbf{w}}_3$ are generated (so $y_t = (\hat{\mathbf{w}}_i \cdot \mathbf{x}_t)$ for $i = 1, 2, 3$ on the corresponding time intervals). The time scale $[1, T]$ is divided into $K = 7$ random consecutive plots, at each site data is generated based on one of these three random regressions $y_t = (\hat{\mathbf{w}}_i \cdot \mathbf{x}_t) + \epsilon$, where $i = 1$ or $i = 2$ or $i = 3$ and ϵ is a standard normal noise. That is, the dependence of y on x is switched 7 times.

Each forecast $f_i(\mathbf{x})$ corresponds to a linear regression trained in a rolling data window $(\mathbf{x}_{i-h}, y_{i-h}), \dots, (\mathbf{x}_i, y_i)$ of length $h = 40$. There are a total of such $T - h + 1$ functional forecasts.

The result of an experiment is shown on Figure 1, where the graphs of $H_t - L_{i,t}$ present the regret of Algorithm 2 with respect to the predictions starting at time moments $i \le t$.[11]

The regret with respect to the ridge regression performed on all data interval is also presented. In most experiments, Algorithm 2 outperforms linear regression in all realizations of the random experiment.

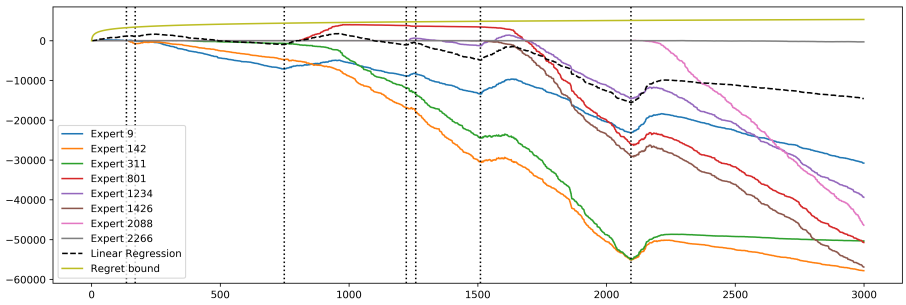

Figure 1: The graphs of the regret with respect to the predictions starting at $i = 9, 142, 311, 801, 1234, 1426, 2088, 2266$ time moments (chosen randomly). The theoretical upper bound for the regret is represented by the line located above all the lines in the graph. The regret with respect to the simple ridge regression is presented by the dotted line.

10. The most appropriate $\eta = \frac{1}{2b^2}$ for the rule (28) and $\eta = \frac{1}{8b^2}$ for (29). A a more straightforward definition (29) results in four times more regret but is easier to compute. Since we set $p_{i,t} = 1$ in the experiment, the quantities $w^p_{i,t}(\mathbf{x}) = w^*_{i,t}$ are simply normalized weights.
11. The forecast $\gamma_t(\mathbf{x})$ is computed by (28), where $\eta = \frac{1}{2b^2}$.

6. Conclusion

In this paper, we have presented a generalization of the prediction with expert advice (PEA) approach for the case when experts present functions as their forecasts instead of point forecasts. For the case of functional forecasts, we use the concept of the integral loss function and of integral mixability.

We use PEA approach for mixing long-term forecasts. We consider any long-term forecast as a functional forecast and apply the delayed feedback approach and the proposed integral generalization of the loss function to evaluate the predictive ability of long-term forecasting methods.

Combining past and current long-term forecasts allows to protects the algorithm from temporary changes in the trend of the time series, noise and outliers. Our mechanism can be applied to the time series forecasting models that are capable of predicting for the infinitely many time moments ahead, e.g. widespread ARMA-like models. For the developed algorithm we proved the time independent regret bound. We have applied PEA approach for the case of online supervised learning, the experts and the learner present predictions in the form of regression functions. The method for smoothing regression using expert advice was presented. Experiments conducted on synthetic data show that the proposed regression algorithm outperforms the ridge regression on the whole data. It would be helpful to compare the performance of this algorithm against AAR by Vovk (2001) and other online regression algorithms.

Acknowledgements

This research was partially supported by Russian foundation for fundamental research: project 20-01-00203.

References

D. Adamskiy, T. Bellotti, R. Dzhamtyrova, Y. Kalnishkan. Aggregating Algorithm for Prediction of Packs. *Machine Learning* 108 (8-9): 1231–1260, 2019.

O. Anava, E. Hazan, S. Mannor. Online Learning for Adversaries with Memory: Price of Past Mistakes, Proceedings of the 28th International Conference on Neural Information Processing Systems, NIPS'15. V.1, 784–792, 2015. http://dl.acm.org/citation.cfm?id=2969239.2969327

A. Blum, Y. Mansour. From external to internal regret. *Journal of Machine Learning Research*. 8:1307–1324, 2007.

G.E.P. Box, D. Pierce. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association*, 65 (332): 1509–1526, 1970.

G.E.P. Box, G. Jenkins. Time Series Analysis. Wiley 2013.

G.E.P. Box, G.Jenkins, G.Reinsel, G. Ljung. Time series analysis: forecasting and control, Wiley & Sons, 2015.

O. Bousquet, M. Warmuth. Tracking a small set of experts by mixing past posteriors. *Journal of Machine Learning Research*. 3: 363–396, 2002.

J. Bröcker. Evaluating raw ensembles with the continuous ranked probability score. *Q. J. R. Meteorol. Soc.*, 138 B: 1611–1617, July 2012.

A. Chernov and V. Vovk. Prediction with expert evaluators advice. In Algorithmic Learning Theory, ALT 2009, Proceedings, volume 5809 of LNCS, pages 8-22. Springer, 2009.

N. Cesa-Bianchi, G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

N. Cesa-Bianchi, Y. Mansour, and G. Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*. 66(2/3): 321–352, 2007.

N. Dunford, J. Schwartz. Linear operators part I: general theory. 243, 1958, Interscience publishers New York.

R. Dzhamtyrova, Y. Kalnishkan. Competitive Online Regression under Continuous Ranked Probability Score. *Proceeding of Mashine Learning Research*. 105: 178–195, 2019.

E.S. Epstein. A scoring system for probability forecasts of ranked categories. *J. Appl. Meteorol. Climatol.*. 8: 985–987, 1969.

P. Gaillard, G. Stoltz, T. van Erven. A Second-order Bound with Excess Losses. *JMLR: Workshop and Conference Proceedings* 35: 1–21, 2014.

Y. Freund, R.E. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*. 55: 119–139, 1997.

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

J. Kivinen, M.K. Warmuth. Averaging expert prediction. In Paul Fisher and Hans Ulrich Simon, editors, *Computational Learning Theory: 4th European Conference (EuroColt '99)*. 153–167. Springer, 1999.

A. Korotin, V. V'yugin, E. Burnaev. Integral Mixabilty: a Tool for Efficient Online Aggregation of Functional and Probabilistic Forecasts. arXiv:1912.07048 [cs.LG], 2019 https://arxiv.org/abs/1912.07048

N. Littlestone, M. Warmuth. The weighted majority algorithm. *Information and Computation* 108: 212–261, 1994.

J. Langford, A. Smola, M. Zinkevich. Slow learners are fast. In Y. Bengio, D. Schuurmans, J. Laderty, C. K. I. Williams, and A. Culotta (eds.), *Advances in Neural Information Processing Systems*. 22: 2331–2339, 2009.

J.E. Matheson, R.L. Winkler. Scoring Rules for Continuous Probability Distributions. *Management Science*. 22(10): 1087-1096, 1976. doi:10.1287/mnsc.22.10.1087

C.J. Mesterharm. On-line learning with delayed label feedback. In Jain, Sanjay, Simon, HansUlrich, and Tomita, Etsuji (eds.), Algorithmic Learning Theory. *Lecture Notes in Computer Science*. 3734: 399–413. Springer. Berlin, Heidelberg, 2005.

C.J. Mesterharm. Improving on-line learning. PhD thesis, Department of Computer Science, Rutgers University, New Brunswick, NJ, 2007.

N. Nikolova, L.E. Persson, S. Varo**v**sanec. A new look at classical inequalities involving Banach lattice norms. *Journal of inequalities and applications*. 2017(1): 302, 2017.

M.J. Weinberger, E. Ordentlich. On delayed prediction of individual sequences. *IEEE Transactions on Information Theory*. 48(7): 1959–1976, 2002.

V. Vovk. Aggregating strategies. In M. Fulk and J. Case, editors, *Proceedings of the 3rd Annual Workshop on Computational Learning Theory*. 371–383. San Mateo, CA, Morgan Kaufmann, 1990.

V. Vovk. A game of prediction with expert advice. *Journal of Computer and System Sciences*. 56(2): 153–173, 1998.

V. Vovk. Competitive on-line statistics. *International Statistical Review* 69: 213–248, 2001.

V. V'yugin, V. Trunov. Online Learning with Continuous Ranked Probability Score, *Proceedings of Machine Learning Research*. 105: 163–177, 2019.