Proceedings of Machine Learning Research 105:1–18, 2019 Conformal and Probabilistic Prediction and Applications

Competitive Online Regression under Continuous Ranked Probability Score

Raisa Dzhamtyrova RAISA.DZHAMTYROVA@RHUL.AC.UK
Department of Computer Science
Royal Holloway, University of London
Egham, United Kingdom
Yuri Kalnishkan YURI.KALNISHKAN@RHUL.AC.UK
Department of Computer Science
Royal Holloway, University of London
Laboratory of Advanced Combinatorics and Network Applications
Moscow Institute of Physics and Technology
Egham, United Kingdom

Editor: Alex Gammerman, Vladimir Vovk, Zhiyuan Luo and Evgueni Smirnov

Abstract

We consider the framework of competitive prediction when one provides guarantees compared to other predictive models that are called experts. We propose the algorithm that combines point predictions of an infinite pool of linear experts and outputs probability forecasts in the form of cumulative distribution functions. We evaluate the quality of probabilistic prediction by the continuous ranked probability score (CRPS), which is a widely used proper scoring rule. We provide a strategy that allows us to 'track the best expert' and derive the theoretical bound on the discounted loss of the strategy. Experimental results on synthetic data and solar power data show that the theoretical bounds of our algorithm are not violated. Also the algorithm performs close to and sometimes outperforms the retrospectively best quantile regression.

Keywords: prediction with expert advice, online learning, competitive prediction, Aggregating Algorithm, continuous ranked probability score (CRPS), probabilistic forecasting

1. Introduction

Probabilistic forecasts serve to quantify an uncertain future and provide a way to make optimal decisions. While an initial focus has been on deterministic forecasting, probabilistic forecasting found its niche in several fields including sports, finance, meteorology and energy. An overview of the state of the art methods and scoring rules in probabilistic forecasting can be found in Gneiting and Katzfuss (2014). One of the frequently used proper scoring rules that evaluates the quality of probabilistic predictions is the continuous ranked probability score (CRPS). The CRPS provides a direct way of comparing point forecasts and probabilistic forecasts. Weighted versions of the CRPS were introduced in Matheson and Winkler (1976).

In this paper we work in the framework of competitive prediction and look for performance guarantees relative to other predictive models called experts. We propose an algo-

rithm that combines point forecasts of an infinite pool of linear regressions and provides probabilistic predictions in the form of cumulative distribution functions. The proposed strategy allows us to 'track the best expert', and the theoretical bound on the discounted loss of the strategy is derived.

Our approach uses the Aggregating Algorithm (AA), which was first introduced in Vovk (1990). In case of mixable loss functions and a finitely many experts, AA gives a guarantee ensuring that the learner's loss is as small as best expert's loss plus a constant. An interesting application of the method of prediction with expert advice for the Brier loss function in forecasting of football outcomes can be found in Vovk and Zhdanov (2009); it is shown that the proposed strategy that follows AA is as good as any bookmaker.

In a recent paper V'yugin and Trunov (2019) it is shown that the CRPS is a mixable loss function, and the theoretical bound for the case of a finite number of experts is derived. In this paper we consider the same problem setting, but we choose a pool of linear regressions to be our experts. We consider the case of discounted loss along the lines of Chernov and Zhdanov (2010). Discounting allows us to give less importance to older losses, which is an important property for practical applications. In Freund and Hsu (2008) the authors noticed that in the context of prediction with expert advice, the discounted loss provides an alternative to 'tracking the best expert' framework of Herbster and Warmuth (1998). Indeed, if the best expert changes after some steps, an algorithm that competes under discounted loss will not take into account small losses of the old best expert because they are strongly discounted, and will switch to track the new best expert.

Our prediction strategy mixes an infinite pool of linear experts in a way which is similar to Aggregating Algorithm for Regression which is proposed in Vovk (2001) for the case of linear experts under the square loss. The generalisation for the case of discounted square loss for linear regression was proposed in Chernov and Zhdanov (2010). The case of generalised linear regression experts under log-loss was introduced in Kakade and Ng (2005), and the case of the square loss was considered in Zhdanov and Vovk (2010).

We perform experiments on a synthetic data set and apply our algorithm for the prediction of solar power. We compare the performance of our algorithm with linear regression and quantile regression. Quantile regression is one of the methods which models a quantile of the response variable conditional on the explanatory variables (Koenker (2005)). Quantile regression has been extensively used to produce renewable energy power quantile forecasts (Koenker and Bassett (1978)) and in probabilistic energy forecasting competitions (Nagya et al. (2016)). Our prediction algorithm uses Markov chain Monte Carlo (MCMC) method in a way which is similar to the algorithm introduced in Zhdanov and Vovk (2010), where AAR was applied to the generalised linear regression class of functions for making a prediction in a fixed interval. With the experiments provided we show that by tuning parameters online, our algorithm moves fast to the area of high values of the probability function and gives a good approximation of the prediction, and theoretical bounds are not violated. In the proposed experiments our algorithm requires some time for training, however by the end of the period the performance of our algorithm becomes close to the performance of the retrospectively best quantile regression.

2. Framework

In the framework of prediction with expert advice we need to specify a *game* $\mathcal{G}$ which contains three components: a space of outcomes Ω, a decision space Γ, and a loss function $\lambda : \Omega \times \Gamma \to \mathbb{R}$.

Suppose that F is the distribution function $F = F_X$ of some random variable X. Then

(a) $F : \mathbb{R} \to [0, 1]$, F is non-decreasing (that is $x \le y \Rightarrow F(x) \le F(y)$),

(b) $\lim_{x\to+\infty} F(x) = 1$, $\lim_{x\to-\infty} F(x) = 0$,

(c) F is right-continuous.

(See Section 3.10 in Williams (1991)).

Let us restrict these functions to a finite interval $[A, B]$, $|A|, |B| < \infty$, that is consider the class of functions $F : [A, B] \to [0, 1]$ satisfying conditions (a) and (c). It is easy to see that these functions can be extended to the whole $\mathbb{R}$ so that they satisfy condition (b). We take this class of functions to be our decision space Γ and take the space of outcomes $\Omega = [A, B] \subset \mathbb{R}$. We measure loss by the CRPS loss function:

$$\lambda(y, F) = \int_A^B (F(u) - 1_{u\ge y})^2 \, du. \tag{1}$$

CRPS loss function generalizes the absolute error; it reduces to the absolute error if F is a point forecast. Indeed, if $F_z(u) = 1_{u\ge z}$, then $\lambda(y, F_z) = |y - z|$.

Learner and experts work according to the following protocol:

Protocol 1

$L_0 := 0$
$L_0^\theta := 0$
for $t = 1, 2, \ldots$
  Accountant announces $\alpha_{t-1} \in (0, 1]$
  Nature announces $x_t \in X \subseteq \mathbb{R}^n$
  Experts output $\xi_t(\theta)$, $\theta \in \Theta$
  Learner outputs $\gamma_t \in \Gamma \subseteq \mathbb{R}^d$
  Nature announces $y_t \in \Omega \subseteq \mathbb{R}^d$
  $L_t^\theta := \alpha_{t-1} L_{t-1}^\theta + \lambda(y_t, \xi_t(\theta))$, $\theta \in \Theta$
  $L_t := \alpha_{t-1} L_{t-1} + \lambda(y_t, \gamma_t)$
end for

The learner in the game of prediction plays against experts θ from some pool Θ, and also accountant and nature. The aim of the learner is to keep his total loss L_t small as compared to the total losses L_t^θ of all experts $\theta \in \Theta$.

We want to find a strategy which is capable of competing with all prediction strategies $\theta \in \mathbb{R}^n$ that at step t outputs:

$$\xi_t(\theta) = F_t^\theta(u) = 1_{u\ge x_t'\theta}, \tag{2}$$

and the loss of the strategy θ is:

$$\lambda(y, \xi_t(\theta)) = |y - x_t'\theta|. \tag{3}$$

Even though the outcome space is an interval, we do not restrict the space of prediction strategies. Note that in the case of the absolute loss the capabilities of prediction with expert advice are restricted. The absolute loss is not mixable and the regret term should have the order of $\sqrt{T}$ (see Kalnishkan and Vyugin (2008)).

In the standard framework of online learning the performance of learners is evaluated by means of cumulative loss. In this paper, we consider the generalisation where we discount the previous losses with the discount factor which is announced at each time step.

The cumulative losses of the learner are discounted with a factor $\alpha_t \in (0, 1]$ at each step. If L_{T-1} is the discounted cumulative loss of the learner at step $T-1$, then the discounted cumulative loss of the learner at step T is defined by

$$L_T := \alpha_{T-1} L_{T-1} + \lambda_T(y_T, \gamma_T) = \sum_{t=1}^{T-1} \left(\prod_{j=t}^{T-1} \alpha_j \right) \lambda_t(y_t, \gamma_t) + \lambda_T(y_T, \gamma_T). \tag{4}$$

If L_{T-1}^θ is the discounted cumulative loss of the prediction strategy θ at the step $T-1$, then the discounted cumulative loss of the prediction strategy θ at the step T is defined by

$$L_T^\theta := \alpha_{T-1} L_{T-1}^\theta + \lambda_T(y_T, \xi_T(\theta)) = \sum_{t=1}^{T-1} \left(\prod_{j=t}^{T-1} \alpha_j \right) \lambda_t(y_t, \xi_t(\theta)) + \lambda_T(y_T, \xi_T(\theta)). \tag{5}$$

In the beginning the losses L_0, L_0^θ are initialized to zero. If all the discount factors are the same, i.e. $\alpha_1 = \cdots = \alpha_T = \alpha$, then we have a case of exponential smoothing. At each step the dependence on the loss at the previous steps exponentially decreases: the initial loss is discounted by α^{T-1} at the step T.

Note that if $\alpha_t = 1$ at each time step t then we have the standard framework of undiscounted loss.

3. Theoretical Bounds

Theorem 1 *Let $a > 0$, $y \in \Omega = [A, B]$, $\gamma \in \Gamma$. There exists a prediction strategy for Learner such that for every positive integer T, every sequence of outcomes of length T, every sequence $\alpha_t \in (0, 1]$, $t = 1, \ldots, T$, and every $\theta \in \mathbb{R}^n$ the discounted cumulative losses L_T of Learner and L_T^θ of expert θ satisfy*

$$L_T \le L_T^\theta + a\|\theta\|_1 + \frac{n(B-A)}{2} \ln \left(1 + \frac{\sum_{t=1}^T w_{t,T}}{a} \max_{t=1,\ldots,T} \|x_t\|_\infty \right), \tag{6}$$

where $w_{t,T} = \prod_{j=t}^{T-1} \alpha_j$.

Note that for the undiscounted losses we have:

Corollary 2 *Let $a > 0$, $y \in \Omega = [A, B]$ and $\gamma \in \Gamma$. There exists a prediction strategy for Learner such that for every positive integer T, every sequence of outcomes of length T, and every $\theta \in \mathbb{R}^n$ the cumulative losses L_T of Learner and L_T^θ of expert θ satisfy*

$$L_T \le L_T^\theta + a\|\theta\|_1 + \frac{n(B-A)}{2} \ln\left(1 + \frac{T}{a} \max_{t=1,\dots,T} \|x_t\|_\infty\right). \tag{7}$$

4. Aggregating Algorithm

We use prediction with expert advice to create our strategy. In the framework of prediction with expert advice we have access to experts' predictions at each time step and the learner has to make a prediction based on experts' past performances. We use an approach based on the AA. The AA is given a parameter η and an initial distribution on experts $P_0(d\theta)$. After each step t it updates the experts' weights according to their losses:

$$P_t(d\theta) = e^{-\eta\lambda(y_t, \xi_t(\theta))} P_{t-1}(d\theta). \tag{8}$$

The weights of experts which suffer large loss at some step will have a smaller importance for making further predictions.

First, we introduce the Aggregating Pseudo-Algorithm (APA) which at step t outputs *generalised prediction*

$$g_t(y) = -\frac{1}{\eta} \ln \int_\Theta e^{-\eta\lambda(y, \xi_t(\theta))} P_{t-1}^*(d\theta), \tag{9}$$

where $P_{t-1}^*(d\theta)$ are normalized weights:

$$P_{t-1}^*(d\theta) = \frac{P_{t-1}(d\theta)}{P_{t-1}(\Theta)},$$

where Θ is a *parameter space*, i.e. experts $\theta \in \Theta$ can output prediction $\xi_t(\theta) \in \Gamma$ at time t.

The generalised prediction can be seen as a weighted average of the experts' predictions in a way which is similar to the Bayesian method.

The AA is obtained from the APA by replacing each generalised prediction g_t by *a permitted prediction* $\Sigma(g_t)$, where the *substitution function* Σ maps every generalised prediction $g : \Omega \to [0, \infty]$ into a permitted prediction $\Sigma(g) \in \Gamma$ satisfying

$$\forall y : \lambda(y, \Sigma(g)) \le g(y). \tag{10}$$

Let us define $P(\Theta)$ as the set of all probability measures over Θ. If a substitution function satisfying (10) for any distribution $P_{t-1}^*(d\theta) \in P(\Theta)$ exists, we say that the loss function is *η-mixable*. The loss function is *mixable* if it is η-mixable for some $\eta > 0$. The game is called *mixable* if the loss function of it is mixable in the setting of the game.

5. Aggregating Algorithm with Discounting

We formulate AA for the case of discounted loss. It is essentially an equivalent to the method in Chernov and Zhdanov (2010). The Aggregating Algorithm with Discounting

(AAD) differs from AA only by the use of the weights in the computation of generalised prediction g_t and the weights update.

For the AAD we denote the discounted weight of expert θ as $\tilde{P}(\theta)$. We initialize a prior distribution on experts $P_0(d\theta)$, $\theta \in \Theta$ and initial discounted weights of experts $\tilde{P}_0(\theta) = 1$.

Instead of (8) the AAD updates weights according to

$$\tilde{P}_t(\theta) = \left(\tilde{P}_{t-1}(\theta)\right)^{\alpha_{t-1}} e^{-\eta\lambda(y_t,\xi_t(\theta))}. \tag{11}$$

The generalised prediction of the AAD is

$$g_t(y) = -\frac{1}{\eta}\ln \int_{\theta\in\Theta} P_0(d\theta)\left(\tilde{P}^*_{t-1}(\theta)\right)^{\alpha_{t-1}} e^{-\eta\lambda(y,\xi_t(\theta))}, \tag{12}$$

where

$$\tilde{P}^*_{t-1}(\theta) = \frac{\tilde{P}_{t-1}(\theta)}{\int_{\theta\in\Theta} P_0(d\theta)\tilde{P}_{t-1}(\theta)}. \tag{13}$$

Lemma 3 *For any learning rate $\eta > 0$, initial prior P_0 and $T = 1, 2, \ldots,$*

$$L_T(AAD(\eta, P_0)) \le -\frac{1}{\eta}\ln \int_\Theta e^{-\eta L^\theta_T} P_0(d\theta). \tag{14}$$

Proof The weights update for AAD is

$$\tilde{P}_t(\theta) = \left(\tilde{P}_{t-1}(\theta)\right)^{\alpha_{t-1}} e^{-\eta\lambda(y_t,\xi_t(\theta))} = e^{-\eta L^\theta_t}. \tag{15}$$

We will prove (14) by induction. At step $t+1$ we can re-write inequality (10) as follows

$$\begin{aligned} e^{-\eta\lambda(y_{t+1},\gamma_{t+1})} &\ge \int_\Theta P_0(d\theta)\left(\tilde{P}^*_t(\theta)\right)^{\alpha_t} e^{-\eta\lambda(y_{t+1},\xi_{t+1}(\theta))} \\ &= \int_\Theta P_0(d\theta)\frac{e^{-\eta\alpha_t L^\theta_t}}{\left(\int_\Theta P_0(d\theta)e^{-\eta L^\theta_t}\right)^{\alpha_t}} e^{-\eta\lambda(y_{t+1},\xi_{t+1}(\theta))}. \end{aligned} \tag{16}$$

Suppose that (14) is true for the step t. By putting the inequality (14) for step t in the power $0 < \alpha_t \le 1$ we obtain

$$e^{-\eta\alpha_t L_t} \ge \left(\int_\Theta P_0(d\theta)e^{-\eta L^\theta_t}\right)^{\alpha_t}.$$

By putting the last inequality in the denominator of (16) we obtain

$$e^{-\eta\lambda(y_{t+1},\gamma_{t+1})} \ge \frac{\int_\Theta e^{-\eta\lambda(y_{t+1},\xi_{t+1}(\theta))-\eta\alpha_t L^\theta_t} P_0(d\theta)}{e^{-\eta L_t\alpha_t}}.$$

By multiplying by the denominator we have

$$e^{-\eta L_{t+1}} \ge \int_\Theta e^{-\eta L^\theta_{t+1}} P_0(d\theta).$$

By taking a natural logarithm of both parts and multiplying by $-\frac{1}{\eta}$ we obtain (14). ■

6. Prediction Strategy

Let $\tilde{\mathcal{G}}$ be the square-loss game with the outcome space $\tilde{\Omega} = \{0, 1\}$, prediction space $\tilde{\Gamma} = [0, 1]$, and the square loss function $\tilde{\lambda}(\omega, \gamma) = (\omega - \gamma)^2$. We consider the game $\mathcal{G}$ as the 'limit' of a sequence of games $\tilde{\mathcal{G}}$ with the vector-valued forecasts. For $d \in \mathcal{N}$ we take points $z_i = A + i\frac{B-A}{d}$, $i = 0, 1, \ldots, d$ and approximate any function $F \in \Gamma$ by a piecewise-constant function F_d defined by $F_d(u) = F(z_i)$ for any $u \in [z_i, z_{i+1})$, $i = 0, 1, \ldots, d-1$. For the game $\tilde{\mathcal{G}}$ the learner's prediction is defined by (Section 2.3.2 in Zhdanov (2011)):

$$\gamma_t = \frac{1}{2} - \frac{g_t(1) - g_t(0)}{2}, \ t = 1, \ldots, T. \tag{17}$$

Let $F_t^\theta \in \Gamma$ be a set of predictions parameterised by $\theta \in \Theta$ at time t. Since the game $\tilde{\mathcal{G}}$ is 2-mixable (Lemma 2.5 in Zhdanov (2011)), we obtain the learner's prediction by putting the expression for generalised prediction of AAD (12):

$$F_t(z_i) = \frac{1}{2} - \frac{1}{4} \ln \frac{\int_{\theta \in \Theta} P_0(d\theta) \left(\tilde{P}_{t-1}(\theta)\right)^{\alpha_{t-1}} e^{-2(F_t^\theta(z_i))^2}}{\int_{\theta \in \Theta} P_0(d\theta) \left(\tilde{P}_{t-1}(\theta)\right)^{\alpha_{t-1}} e^{-2(1-F_t^\theta(z_i))^2}}, \ i = 0, 1, \ldots, d-1. \tag{18}$$

By letting $d \to +\infty$ in (18), we obtain the expression for computing the learner's forecast:

$$\gamma_t = F_t(u) = \frac{1}{2} - \frac{1}{4} \ln \frac{\int_{\theta \in \Theta} P_0(d\theta) \left(\tilde{P}_{t-1}(\theta)\right)^{\alpha_{t-1}} e^{-2(F_t^\theta(u))^2}}{\int_{\theta \in \Theta} P_0(d\theta) \left(\tilde{P}_{t-1}(\theta)\right)^{\alpha_{t-1}} e^{-2(1-F_t^\theta(u))^2}}, \ u \in [A, B]. \tag{19}$$

We choose the initial distribution of the parameters for some $a > 0$:

$$P_0(d\theta) = \left(\frac{a\eta}{2}\right)^n e^{-a\eta\|\theta\|_1} d\theta, \tag{20}$$

where $\theta \in \mathbb{R}^n$.

Then the learner's prediction (19) can be re-written as follows:

$$\gamma_t = F_t(u) = \frac{1}{2} - \frac{1}{4} \ln \frac{\int_{\theta \in \Theta} q_t^*(\theta) e^{-2(F_t^\theta(u))^2} d\theta}{\int_{\theta \in \Theta} q_t^*(\theta) e^{-2(1-F_t^\theta(u))^2} d\theta}, \tag{21}$$

where

$$q_T^*(\theta) = C q_T(\theta) = C \exp\left(-\sum_{t=1}^{T-1} \left(\prod_{j=t}^{T-1} \alpha_j\right) |y_t - x_t'\theta| - a\|\theta\|^2\right), \tag{22}$$

and C is the normalising constant ensuring that $\int_\Theta q_T^*(\theta) d\theta = 1$.

Since

$$e^{-2} \le \int_{\theta \in \Theta} e^{-2(F^\theta(u))^2} q^*(\theta) d\theta, \int_{\theta \in \Theta} e^{-2(1-F^\theta(u))^2} q^*(\theta) d\theta \le 1 \tag{23}$$

we get $0 \le F(u) \le 1$. Since $F^\theta(u)$ is non-decreasing in u, our $F(u)$ is non-decreasing too. By the Monotone Convergence Theorem (Theorem 5.3 in Williams (1991)) if $u \downarrow u_0$ then

$$\int_{\theta\in\Theta} e^{-2(F^\theta(u))^2} q^*(\theta)d\theta \downarrow \int_{\theta\in\Theta} e^{-2(F^\theta(u_0))^2} q^*(\theta)d\theta$$
$$\int_{\theta\in\Theta} e^{-2(1-F^\theta(u))^2} q^*(\theta)d\theta \uparrow \int_{\theta\in\Theta} e^{-2(1-F_t^\theta(u_0))^2} q^*(\theta)d\theta.$$

Therefore, F is right-continuous. We have shown that $F \in \Gamma$.

For completeness, we include the following lemma from V'yugin and Trunov (2019) and go through the details of the proof.

Lemma 4 *Game $\mathcal{G}$ where the space of outcomes $\Omega = [A, B]$ and decision space Γ contains probability distribution functions $F : [A, B] \to [0, 1]$, and CRPS loss function (1) is $\frac{2}{B-A}$-mixable.*

Proof We need to show that prediction (21) satisfies (10), that is

$$\lambda(y, F) \le -\frac{1}{\eta} \ln \int_\Theta e^{-\eta\lambda(y,F^\theta)} P(d\theta) \tag{24}$$

for all $y \in [A, B]$.

The CRPS loss function can be represented as:

$$\lambda(y, F_d) = \sum_{i=0}^{j-1} \int_{z_i}^{z_{i+1}} F_d^2(u)du + \int_{z_j}^{y} F_d^2(u)du + \int_y^{z_{j+1}} (1 - F_d(u))^2 \, du$$
$$+ \sum_{i=j+1}^{d-1} \int_{z_i}^{z_{i+1}} (1 - F_d(u))^2 \, du = \frac{B-A}{d} \sum_{i=0}^{j-1} F^2(z_i) + (y - z_j)F^2(z_j)$$
$$+ (z_{j+1} - y) \, (1 - F(z_j))^2 + \frac{B-A}{d} \sum_{i=j+1}^{d-1} (1 - F(z_i))^2 \,, \tag{25}$$

where $y \in [z_j, z_{j+1})$.

An outcome y can be identified with a vector $\omega = (\omega_0^y, \omega_1^y, \ldots, \omega_{d-1}^y)$, where $\omega_i^y = 1_{z_{i+1} \ge y} \in \{0, 1\}$ for $i = 0, 1, \ldots, d-1$. Let us define the loss function $\hat{\lambda}(y, F_d)$ by

$$\hat{\lambda}(y, F_d) = \frac{B-A}{d} \sum_{i=0}^{d-1} (\omega_i^y - F(z_i))^2 = \frac{B-A}{d} \left(\sum_{i=0}^{j-1} F^2(z_i) + \sum_{i=j}^{d-1} (1 - F(z_i))^2 \right), \tag{26}$$

where $y \in [z_j, z_{j+1})$. We get:

$$|\lambda(y, F_d) - \hat{\lambda}(y, F_d)| = (y - z_j)|F^2(z_j) - (1 - F(z_j))^2| = (y - z_j)|2F(z_j) - 1| \le \frac{B-A}{d}, \tag{27}$$

where $y \in [z_j, z_{j+1}),\ j = 0, 1, \ldots, d-1$.

Consider the game $\hat{\mathcal{G}}$ with the outcome and prediction spaces given by the Cartesian products $\tilde{\Omega}^d$ and $\tilde{\Gamma}^d$ and the loss function $\frac{1}{d}\sum_{i=1}^d \tilde{\lambda}(\omega_i, \gamma_i)$. By Lemma 1 in Adamskiy

et al. (2019), the game $\hat{\mathcal{G}}$ is 2-mixable. For the experts' predictions $(\gamma_0^\theta, \dots, \gamma_{d-1}^\theta) = (F^\theta(z_0), \dots, F^\theta(z_{d-1}))$, $\theta \in \Theta$, the learner's predictions $(F(z_0), \dots, F(z_{d-1}))$ satisfy

$$\frac{1}{d}\sum_{i=0}^{d-1}(\omega_i - F(z_i))^2 \le -\frac{1}{2}\ln\int_\Theta e^{-\frac{2}{d}\left(\sum_{i=0}^{d-1}(\omega_i - F^\theta(z_i))^2\right)}P(d\theta)$$

for all $\omega_i \in [0,1]$, $i = 0, 1, \dots, d-1$ including ω_i^y, $y \in [A, B]$. In other terms, we get

$$\frac{1}{B-A}\hat{\lambda}(y, F_d) \le -\frac{1}{2}\ln\int_\Theta e^{-\frac{2}{B-A}\hat{\lambda}(y, F_d^\theta)}P(d\theta).$$

By using inequality (27), we have:

$$\lambda(y, F_d) \le -\frac{B-A}{2}\ln\int_\Theta e^{-\frac{2}{B-A}\lambda(y, F_d^\theta)}P(d\theta) + 2\frac{B-A}{d}. \tag{28}$$

Now it remains to show that losses $\lambda(y, F_d)$ and $\lambda(y, F)$ do not differ much. Since, by construction, $F(u) \le F_d(u)$, we get

$$|\lambda(y, F) - \lambda(y, F_d)| \le \int_A^y \left(F^2(u) - F_d^2(u)\right)du + \int_y^B \left((1 - F_d(u))^2 - (1 - F(u))^2\right)du.$$

The first integral can be upper bounded as

$$\begin{aligned}\int_A^y \left(F^2(u) - F_d^2(u)\right)du &= \sum_{i=0}^{j-1}\int_{z_i}^{z_{i+1}}\left(F^2(u) - F_d^2(u)\right)du + \int_{z_j}^{y}\left(F^2(u) - F_d^2(u)\right)du \\ &\le \frac{B-A}{d}\sum_{i=0}^{j}\left(F^2(z_{i+1}) - F^2(z_i)\right) = \frac{B-A}{d}\left(F^2(z_{j+1}) - F^2(A)\right) \\ &\le \frac{B-A}{d}\left(F^2(B) - F^2(A)\right) \le \frac{B-A}{d}.\end{aligned}$$

By doing the same for the second integral, we get

$$|\lambda(y, F) - \lambda(y, F_d)| \le 2\frac{B-A}{d}. \tag{29}$$

Now inequality (24) follows from (28) by letting $d \to +\infty$. ■

Integrals in (21) is a Bayesian mixture, where predictions γ_T needs to be integrated with respect to the normalized distribution $q_T^*(\theta)$. It is possible to avoid the calculation of normalization constant C as it is a computationally inefficient operation, and integrate function γ_T from the unnormalized distribution $q_T(\theta)$. In order to calculate the integral (21), it is possible to use MCMC algorithms. MCMC techniques are often applied to solve integration and optimisation problems in large dimensional spaces. MCMC is a strategy for generating samples while exploring the state space using a Markov chain mechanism. This mechanism is constructed so that the chain spends more time in the most important regions. The good introduction of MCMC for Machine Learning is in Andrieu et al. (2003).

We will use Metropolis-Hastings algorithm for sampling parameters θ from the posterior distribution $\mathcal{P}$. As a proposal distribution we chose Gaussian distribution $\mathcal{N}(0, \sigma^2)$ with some parameter σ. We start with some initial parameter θ_0 and at each step m we update:

$$\theta^m = \theta^{m-1} + \mathcal{N}(0, \sigma^2),\ m = 1, \ldots, M,$$

where M is a maximum number of iterations in MCMC method.

The update parameter θ^m at step m is accepted with probability $\min\left(1, \frac{f_{\mathcal{P}}(\theta^m)}{f_{\mathcal{P}}(\theta^{m-1})}\right)$, where $f_{\mathcal{P}}(\theta)$ is the density function for the distribution $\mathcal{P}$ at point θ. At each step by accepting and rejecting the updates of parameters θ we move closer to the maximum of the density function. At the beginning it is common to use 'burn-in' stage when the integral is not calculated till we will not reach the area of high values of density function $f_{\mathcal{P}}$. Thus, we perform integration only from the area with high density of $\mathcal{P}$. Some values of θ are accepted even when the calculated probability is less than 1, it allows the algorithm to move away from local minimum of the density function. Because we are interested only in the ratio of density functions of generated parameters, we can generate new parameters θ from the unnormalized posterior distribution $q_T(\theta)$ and avoid the weights normalization at each step which is more computationally efficient.

At time $t = 0$ the algorithm starts with the initial estimate of the parameters $\theta_0 = 0$. At each iteration $t > 0$ we start with parameter θ^M calculated at the previous step $t - 1$. It allows the algorithm to converge faster to the correct location of the main mass of the distribution.

Algorithm

Parameters: number $M > 0$ of MCMC iterations,
standard deviation $\sigma > 0$,
regularization coefficient $a > 0$,
prediction interval $[A, B]$.

$\eta := \frac{2}{B-A}$
initialize $\theta_0^M := 0 \in \Theta$
define $q_0(\theta) := \exp(-a\eta\|\theta\|_1)$
for $t = 1, 2, \ldots$ do
  $\gamma_t^0 := 0,\ \gamma_t^1 := 0$
  read $x_t \in \mathbb{R}^n$
  initialize $\theta_t^0 = \theta_{t-1}^M$
  for $m = 1, 2, \ldots, M$ do
    $\theta^* := \theta_t^{m-1} + \mathcal{N}(0, \sigma^2 I)$
    flip a coin with success probability
      $\min\left(1, q_{t-1}(\theta^*)/q_{t-1}(\theta_t^{m-1})\right)$
    if success then
      $\theta_t^m := \theta^*$
    else
      $\theta_t^m := \theta_{t-1}^m$
    end if

$$\gamma_t^0 := \gamma_t^0 + \exp\left(-2\left(F_t^{\theta_t^m}(u)\right)^2\right)$$
$$\gamma_t^1 := \gamma_t^1 + \exp\left(-2\left(1 - F_t^{\theta_t^m}(u)\right)^2\right)$$
end for
output predictions $\gamma_t = \frac{1}{2} - \frac{1}{4}\ln\frac{\gamma_t^0}{\gamma_t^1}$
end for

7. Proof of Theoretical Bounds

In this section we provide the proof of Theorem 1. Applying Lemma 3 for initial distribution (20) we obtain

$$L_T \le -\frac{1}{\eta}\ln\left(\left(\frac{a\eta}{2}\right)^n \int_\Theta e^{-\eta J(\theta)} d\theta\right), \tag{30}$$

where

$$J(\theta) := \sum_{t=1}^T w_{t,T}|x_t'\theta - y_t| + a\|\theta\|_1$$

and

$$w_{t,T} = \prod_{j=t}^{T-1} \alpha_j,\ w_{T,T} = 1.$$

For all $\theta, \theta_0 \in \mathbb{R}^n$ we have:

$$\begin{aligned}\sum_{t=1}^T w_{t,T}|x_t'\theta - y_t| &\le \sum_{t=1}^T w_{t,T}|x_t'\theta_0 - y_t| + \sum_{t=1}^T w_{t,T}|x_t'\theta - x_t'\theta_0| \\ &\le \sum_{t=1}^T w_{t,T}|x_t'\theta_0 - y_t| + \sum_{t=1}^T w_{t,T}\max_{t=1,\dots,T}\|x_t\|_\infty\|\theta - \theta_0\|_1. \end{aligned} \tag{31}$$

Then, we have:

$$\begin{aligned} J(\theta) &\le J(\theta_0) + \sum_{t=1}^T w_{t,T}\max_{t=1,\dots,T}\|x_t\|_\infty|\theta - \theta_0| + a(\|\theta\|_1 - \|\theta_0\|_1) \\ &\le J(\theta_0) + \max_{t=1,\dots,T}\|x_t\|_\infty \sum_{t=1}^T w_{t,T}\|\theta - \theta_0\|_1 + a\|\theta - \theta_0\|_1 \\ &= J(\theta_0) + \left(\max_{t=1,\dots,T}\|x_t\|_\infty \sum_{t=1}^T w_{t,T} + a\right)\|\theta - \theta_0\|_1. \end{aligned} \tag{32}$$

Let us denote $b_T = \max_{t=1,\dots,T} \|x_t\|_\infty \sum_{t=1}^T w_{t,T} + a$. We evaluate the integral

$$\int_\Theta e^{-\eta J(\theta)} d\theta \geq \int_{\mathbb{R}^n} e^{-\eta(J(\theta_0)+b_T\|\theta-\theta_0\|_1)} d\theta$$
$$= e^{-\eta J(\theta_0)} \int_{\mathbb{R}} \dots \int_{\mathbb{R}} e^{-\eta b_T \sum_{i=1}^n |\theta_i-\theta_{i,0}|} d\theta_i = e^{-\eta J(\theta_0)} \int_{\mathbb{R}} \dots \int_{\mathbb{R}} \prod_{i=1}^n e^{-\eta b_T|\theta_i-\theta_{i,0}|} d\theta_i$$
$$= e^{-\eta J(\theta_0)} \prod_{i=1}^n \int_{\mathbb{R}} e^{-\eta b_T|\theta_i-\theta_{i,0}|} d\theta_i = e^{-\eta J(\theta_0)} \left(\frac{2}{\eta b_T}\right)^n.$$

By putting this expression in (7) we have

$$L_T \leq J(\theta_0) - \frac{1}{\eta} \ln\left(\left(\frac{a\eta}{2}\right)^n \left(\frac{2}{\eta b_T}\right)^n\right) = L_T^{\theta_0} + a\|\theta_0\|_1 + \frac{n}{\eta} \ln\left(1 + \frac{\sum_{t=1}^T w_{t,T}}{a} \max_t \|x_t\|_\infty\right).$$

By putting $\eta = \frac{2}{B-A}$ from Lemma 4 we obtain the theoretical bound (6).

8. Experiments

In this section we apply our proposed algorithm on synthetic data and solar power data, and compare its performance with other predictive models. The solar power data set is downloaded from Open Power System Data which provides free and open data platform for power system modelling. The platform contains hourly measurements of geographically aggregated weather data across Europe and time-series of solar power. Our training data are measurements in Austria from January to December 2015, test set contains data from January to April 2016. [1]

8.1. Synthetic data

We apply our algorithm on synthetic data sets. The first data set is generated from the linear model $y = 2x - 1 + \epsilon$, where $\epsilon \in \mathcal{N}(0, 0.001)$ and feature x is generated from normal distribution $\mathcal{N}(0.75, 0.05)$. Figure 1 illustrates the generated data set which contains 1000 observations. We divide our data in a way that it has half of its observations in training and test data sets. First, we will run our algorithm and train the linear regression on training data set and compare their performance. From Figure 1 we can see that the data set is almost perfectly linear; and the linear regression model, trained on training data set, has $R^2 = 0.9999$ on the test data. We run our algorithm for the number of MCMC iterations $M = 1500$ and 'burn-in' period $M_0 = 300$ for different parameters of regularization a and standard deviation σ. For this example, we pick our parameters of regularization $a = 0.5$, standard deviation $\sigma = 0.1$, and we do not discount our losses $\alpha_t = 1$ for $t = 1, \dots, T$. Figure 3 shows the difference between cumulative losses of the linear regression and our algorithm $L_T^{\theta^*} - L_T$ on test data set, where θ^* was obtained by linear regression model on training data set. We also plot the theoretical bound for our algorithm. The initial large gap corresponds to the value $-a\|\theta^*\|_1$, which gives the initial start to Learner on expert θ^*.

1. The code written in R is available at https://github.com/RaisaDZ/CRPS.

As time increases, we add an additional value $-\frac{n(B-A)}{2}\ln\left(1+\frac{T}{a}\max\|x_t\|_\infty\right)$ to the bound. We can see from the graph that initially the loss difference is decreasing fast which means that loss of our algorithm becomes larger compared to the loss of linear regression model. The initial start $-a\|\theta^*\|_1$ gives us some time for training. After the initial training time passes, the difference between cumulative losses becomes smoother and behaves in a similar way with the theoretical bound of our algorithm which is decreasing logarithmically with the number of steps. Figure 4 illustrates the difference between cumulative losses of the quantile regression and our algorithm which behaves in a similar way with the previous graph. Total loss of our algorithm on the test data set is 3.05. It is much larger than the loss of the best linear regression model which is equal to 0.42, and the loss of the quantile regression which is equal to 0.30. It is not surprising as the data set is almost perfectly linear, and our algorithm makes a large loss during the initial training. However, the theoretical bound of our algorithm is not violated.

The second synthetic data set is similar with the previous one, but the slope of the model slowly changes with time $y_t = (2 + 0.00005t)x_t - 1 + \epsilon,\ t = 1, \dots, T$. Figure 2 illustrates the generated data set. We use the same parameters of our algorithm as in the previous example, but we add exponential discounting $\alpha_t = 0.999,\ t = 1, \dots, T$. The data set still looks linear; and the linear regression model, trained on training data set, has $R^2 = 0.9681$ on the test data. Figure 5 shows the difference between different competitors and our algorithm. We can see from the graph that after around 50 iterations the loss difference starts to increase which means that our algorithm starts to perform better than other models. At the end of the period total loss of the best linear regression (LM) is 9.32, loss of the quantile regression trained on training set (QR) is 7.75, loss of the quantile regression trained online (QR online) is 4.66. Total loss of our algorithm is equal 4.55, which is slightly lower than the total loss of the quantile regression trained online.

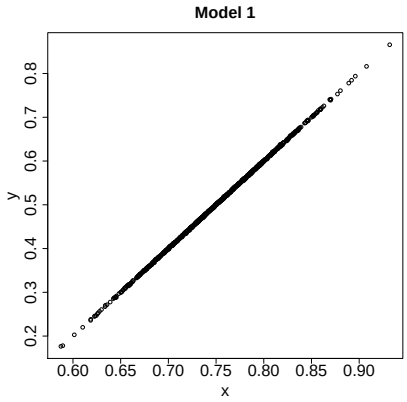

Figure 1: First data set

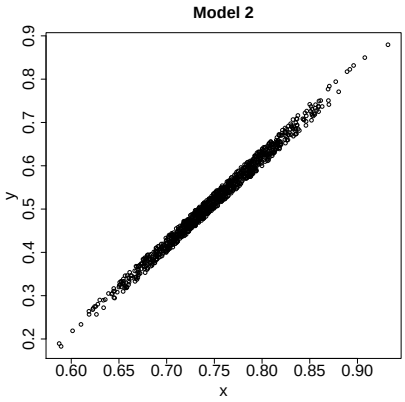

Figure 2: Second data set

8.2. Solar power data

We perform similar experiments for prediction of solar power. We choose measurements of direct and diffuse radiations to be our explanatory variables. Figures 6, 7 show the

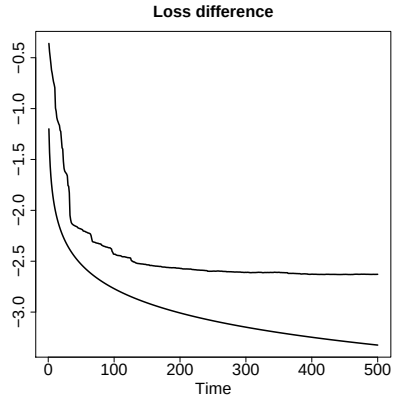

Figure 3: Loss difference between the best linear regression and our algorithm

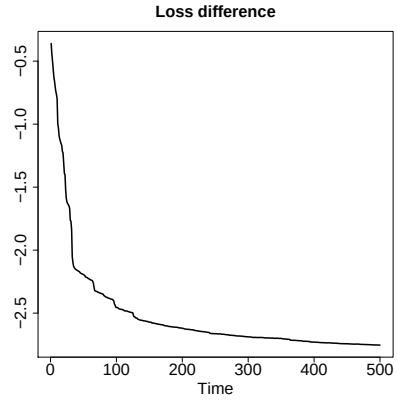

Figure 4: Loss difference between the best quantile regression and our algorithm

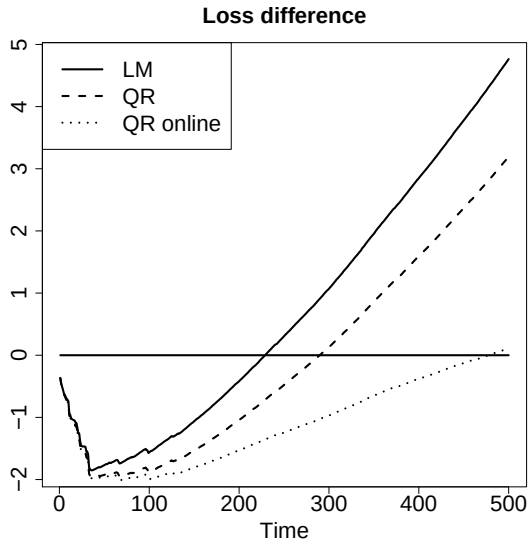

Figure 5: Loss difference between competitors and our algorithm

dependence of solar power on explanatory variables on the training set. We can see that there is a linear dependence between predicted and explanatory variables. The correlation between solar power and direct radiation is equal to 0.88, whereas the correlation between solar power and diffuse radiation is equal to 0.74. First, similar to the previous experiments, we fit linear regression on the training set. The linear regression seems to perform well on this data set; it has $R^2 = 0.8929$ on the test set.

Now we run our algorithm for the number of MCMC iterations $M = 1500$ and 'burn-in' period $M_0 = 300$ for different parameters of regularization a and standard deviation σ.

Table 1 shows the acceptance ratio of new generated parameters θ for different parameters a and σ. We can notice from the table that standard deviation σ affects the acceptance ratio quite a lot, whereas regularization parameter a has a little affect. If no prior knowledge is available, one can start with some reasonable values of input parameters and keep track of the acceptance ratio of new generated θ. If the acceptance ratio is too high it might indicate that the algorithm moves too slowly to the area of high values of the probability function of θ, and standard deviation σ should be increased. Another option is to take very large number of steps and larger 'burn-in' period. For this example, we pick our parameters of regularization $a = 0.1$, standard deviation $\sigma = 0.03$, and we do not discount our losses $\alpha_t = 1$ for $t = 1, \ldots, t$. Figure 8 shows the difference between cumulative losses of the best linear regression trained on the training set and our algorithm, and Figure 9 shows the difference between cumulative losses of the best quantile regression trained on the training set and our algorithm. We can see from the graphs that we need a little time to outperform the linear regression model, but our algorithm performs much worse than the quantile regression as the difference of cumulative losses decreases fast. However, after around 2000 steps the difference of cumulative losses stabilizes and becomes more 'flattened' which indicates that the performance of our algorithm becomes close to the performance of the quantile regression.

Figures 10, 11 show predictions of our algorithm and quantile regression (QR) with $[25\%, 75\%]$ confidence interval for the first 100 steps and after 1000 steps respectively. We can see from the graph, that initially predictions of our algorithm are very different from predictions of QR. However, after the initial training of our algorithm, predictions of both methods become very close to each other.

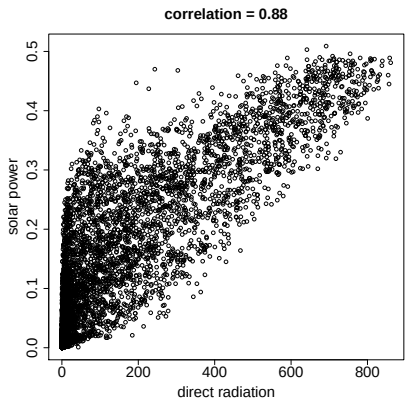

Figure 6: Dependence of solar power on direct radiation

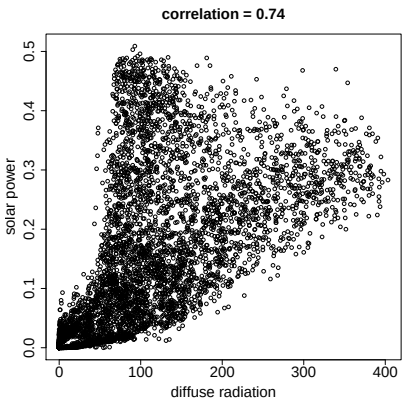

Figure 7: Dependence of solar power on diffuse radiation

Table 1: Acceptance ratio of new generated parameters

$a \setminus \sigma$	0.01	0.02	0.03	0.05	0.10
0.1	0.858	0.602	0.410	0.214	0.070
0.5	0.857	0.602	0.410	0.214	0.069
1.0	0.858	0.601	0.409	0.215	0.069

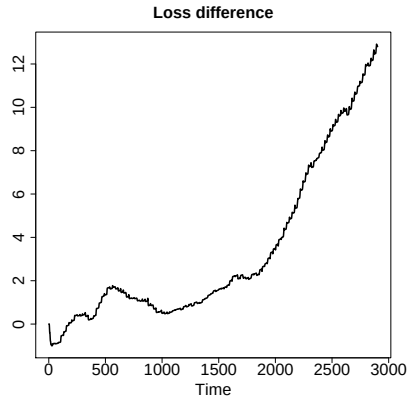

Figure 8: Loss difference between the best linear regression and our algorithm

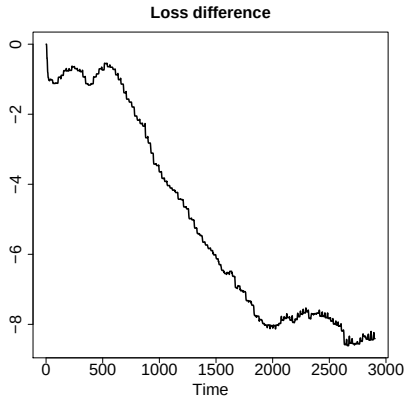

Figure 9: Loss difference between the best quantile regression and our algorithm

9. Conclusions

We propose an algorithm that combines deterministic predictions of an infinite pool of linear regressions and outputs probability forecasts in the form of cumulative distribution functions. The proposed strategy allows us to 'track the best expert'. The theoretical bound on the discounted cumulative CRPS loss function of the algorithm is derived.

We perform experiments to evaluate the performance of our algorithm on synthetic and solar power data sets. The first experiment shows that the theoretical bound of our algorithm is not violated. The second experiment on the synthetic data set show that the loss discounting helps in situations when the underlying nature of data changes with time; and our algorithm can outperform the best online quantile regression. The experiment with prediction of solar power shows that our algorithm needs some time for training, however after an initial time passes, the performance of our algorithm becomes close to the performance of the quantile regression.

References

D. Adamskiy, A. Bellotti, R. Dzhamtyrova, and Y. Kalnishkan. Aggregating algorithm for prediction of packs. *Machine Learning*, 2019. doi: 10.1007/s10994-018-5769-2.

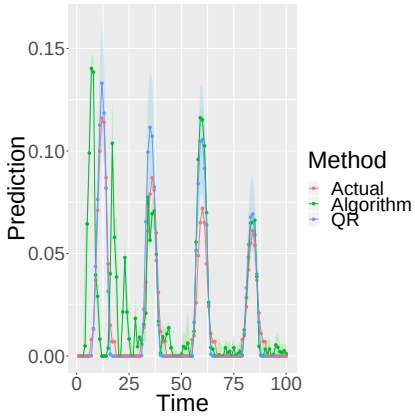

Figure 10: Predictions with $[25\%, 75\%]$ confidence interval of our Algorithm and QR, first 100 steps

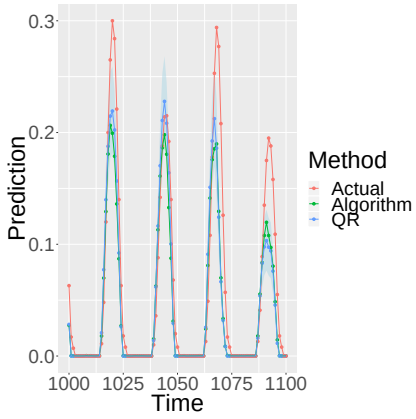

Figure 11: Predictions with $[25\%, 75\%]$ confidence interval of our Algorithm and QR, after 1000 steps

C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine Learning Journal*, page 50:5–43, 2003.

A. Chernov and F. Zhdanov. Prediction with expert advice under discounted loss. In *Proccedings of ALT 2010*, volume LNAI 6331, pages 255–269. Springer, 2010. See also arXiv:1005.1918 [cs.LG].

Y. Freund and D. Hsu. A new hedging algorithm and its application to inferring latent random variables. *Technical report, arXiv:0806.4802 [cs.GT], arXiv.org e-Print archive*, 2008.

T. Gneiting and M. Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, pages 1: 125–151, 2014.

M. Herbster and M. K. Warmuth. Tracking the best expert. *Machine Learning*, 32:151–178, 1998.

S. M. Kakade and A. Y. Ng. Online bounds for bayesian algorithms. *Advances in Neural Information Processing Systems 17*, pages 641–648, 2005.

Y. Kalnishkan and M. V. Vyugin. The weak aggregating algorithm and weak mixability. *Journal of Computer and System Sciences*, 74(8):1228–1244, 2008.

R. Koenker. Quantile regression. *Cambridge, UK: Cambridge Univ. Press*, 2005.

R. Koenker and G. Bassett. Regression quantiles. *Econometrica*, pages 46: 33–50, 1978.

J. E. Matheson and R. L. Winkler. Scoring rules for continuous probability distributions. *Management Science*, page 22:1087–96, 1976.

G. I. Nagya, G. Bartaa, S. Kazia, G. Borbelyb, and G. Simon. Gefcom2014: Probabilistic solar and wind power forecasting using a generalized additive tree ensemble approach. *International Journal of Forecasting*, pages 32: 1087–1093, 2016.

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

V. Vovk. Competitive on-line statistics. *International Statistical Review*, 69(2):213–248, 2001.

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

V. V'yugin and V. Trunov. Online learning with continuous ranked probability score. 2019. doi: https://arxiv.org/abs/1902.10173.

D. Williams. *Probability with Martingales*. Cambridge University Press, 1991.

F. Zhdanov. Theory and applications of competitive prediction. *PhD thesis*, 2011.

F. Zhdanov and V. Vovk. Competitive online generalized linear regression under square loss. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 531–546, 2010.