JMLR: Workshop and Conference Proceedings vol 49:1–25, 2016

Online Isotonic Regression

Wojciech Kotłowski WKOTLOWSKI@CS.PUT.POZNAN.PL
Poznań University of Technology

Wouter M. Koolen WMKOOLEN@CWI.NL
Centrum Wiskunde & Informatica

Alan Malek MALEK@BERKELEY.EDU
University of California at Berkeley

Abstract

We consider the online version of the isotonic regression problem. Given a set of linearly ordered points (e.g., on the real line), the learner must predict labels sequentially at adversarially chosen positions and is evaluated by her total squared loss compared against the best isotonic (non-decreasing) function in hindsight. We survey several standard online learning algorithms and show that none of them achieve the optimal regret exponent; in fact, most of them (including Online Gradient Descent, Follow the Leader and Exponential Weights) incur linear regret. We then prove that the Exponential Weights algorithm played over a covering net of isotonic functions has a regret bounded by $O\big(T^{1/3}\log^{2/3}(T)\big)$ and present a matching $\Omega(T^{1/3})$ lower bound on regret. We provide a computationally efficient version of this algorithm. We also analyze the noise-free case, in which the revealed labels are isotonic, and show that the bound can be improved to $O(\log T)$ or even to $O(1)$ (when the labels are revealed in isotonic order). Finally, we extend the analysis beyond squared loss and give bounds for entropic loss and absolute loss.

Keywords: online learning, isotonic regression, isotonic function, monotonic, nonparametric regression, exp-concave loss.

1. Introduction

We propose a problem of sequential prediction in the class of isotonic (non-decreasing) functions. At the start of the game, the learner is given a set of T linearly ordered points (e.g., on the real line). Then, over the course of T trials, the adversary picks a new (as of yet unlabeled) point and the learner predicts a label from $[0,1]$ for that point. Then, the true label (also from $[0,1]$) is revealed, and the learner suffers the squared error loss. After T rounds the learner is evaluated by means of the regret, which is its total squared loss minus the loss of the best isotonic function in hindsight.

Our problem is precisely the online version of *isotonic regression*, a fundamental problem in statistics, which concerns fitting a sequence of data where the prediction is an isotonic function of the covariate (Ayer et al., 1955; Brunk, 1955; Robertson et al., 1998). Isotonic constraints arise naturally in many structured problems, e.g. predicting the height of children as a function of age, autocorrelation functions, or biomedical applications such as estimating drug dose responses (Stylianou and Flournoy, 2002). Despite being simple and commonly used in practice, isotonic regression is an example of nonparametric regression where the number of parameters grows linearly with the number of data points. A natural question to ask is whether there are efficient, provably low regret algorithms for online isotonic regression.

Since online isotonic regression concerns minimizing a convex loss function over the convex set of feasible prediction strategies (isotonic functions), it can be analyzed within the framework of online convex optimization (Shalev-Shwartz, 2012). We begin by surveying popular online learning algorithms in our setting and showing that most of them (including Online Gradient Descent, Follow the Leader and Exponential Weights) suffer regret that is linear in the number of data points in the worst case. The failure of most standard approaches makes the problem particularly interesting. We also show that the Exponentiated Gradient algorithm delivers a $O(\sqrt{T \log T})$ regret guarantee which is nontrivial but suboptimal.

We then propose an algorithm which achieves the regret bound $O\big(T^{1/3} \log^{2/3}(T)\big)$. The algorithm is a simple instance of Exponential Weights that plays on a covering net (discretization) of the class of isotonic functions. Despite the exponential size of the covering net, we present a computationally efficient implementation with $O(T^{4/3})$ time per trial. We also show a lower bound $\Omega(T^{1/3})$ on the regret of any algorithm, hence proving that the proposed algorithm is optimal (up to a logarithmic factor).

We also analyze the noise-free case where the labels revealed by the adversary are isotonic and therefore the loss of the best isotonic function is 0. We show that the achievable worst-case regret in this case scales only logarithmically in T. If we additionally assume that the labels are queried in isotonic order (from left to right), the achievable worst-case regret drops to 1. In both cases, we are able to determine the minimax algorithm and the actual value of the minimax regret.

Finally, we go beyond the squared loss and adapt our discretized Exponential Weights algorithm to logarithmic loss and get the same regret guarantee. We also consider isotonic regression with absolute loss and show that the minimax regret is of order $\tilde{O}(\sqrt{T})$ and is achieved, up to a logarithmic factor, by the Exponentiated Gradient algorithm.

1.1. Related work

Isotonic regression has been extensively studied in statistics starting from work by Ayer et al. (1955); Brunk (1955). The excellent book by Robertson et al. (1998) provides a history of the subject and numerous references to the statistical literature. Isotonic regression has applications throughout statistics (e.g. nonparametric regression, estimating monotone densities, parameter estimation and statistical tests under order constraints, multidimensional scaling, see Robertson et al. 1998) and to more practical problems in biology, medicine, psychology, etc. (Kruskal, 1964; Stylianou and Flournoy, 2002; Obozinski et al., 2008; Luss et al., 2012).

The classical problem of minimizing an isotonic function under squared loss (the offline counterpart of this paper) has usually been studied in statistics under a generative model $y_i = f(x_i) + \epsilon_i$ with $f(x_i)$ being some isotonic function and ϵ_i being random i.i.d. noise variables (Van de Geer, 1990; Birgé and Massart, 1993; Zhang, 2002). It is known (see, e.g., Zhang, 2002) that the statistical risk of the isotonic regression function $\mathbb{E}[\frac{1}{T}\|\widehat{f} - f\|^2]$ converges at the rate of $O(T^{-2/3})$, where T is the sample size. Interestingly, this matches (up to a logarithmic factor) our results on online isotonic regression, showing that the online version of the problem is not fundamentally harder.

In machine learning, isotonic regression is used to calibrate class probability estimates (Zadrozny and Elkan, 2002; Niculescu-Mizil and Caruana, 2005; Menon et al., 2012; Narasimhan and Agarwal, 2013; Vovk et al., 2015), for ROC analysis (Fawcett and Niculescu-Mizil, 2007), for learning Generalized Linear Models and Single Index Models (Kalai and Sastry, 2009; Kakade et al., 2011), for data cleaning (Kotłowski and Słowiński, 2009) and for ranking (Moon et al., 2010). Recent work

by Kyng et al. (2015) proposes fast algorithms under general partial order constraints. None of these works are directly related to the subject of this paper. The one related problem we found is online learning with logarithmic loss for the class of monotone predictors as studied by Cesa-Bianchi and Lugosi (2001), who give an upper bound on the minimax regret (the bound is not tight for our case).

We also note that the problem considered here falls into a general framework of online nonparametric regression. Rakhlin and Sridharan (2014) give nonconstructive upper and lower bound on the minimax regret, but using their bounds for a particular function class requires upper and lower bounds on its sequential entropy. In turn, our upper bound is achieved by an efficient algorithm, while the lower bound follows from a simple construction. Gaillard and Gerchinovitz (2015) propose an algorithm, called Chaining Exponentially Weighted Average Forecaster, that is based on aggregation on two levels. On the first level, a multi-variable version of Exponentiated Gradient is used, while on the second level, the Exponential Weights algorithm is used. The combined algorithm works for any totally bounded (in terms of metric entropy) set of functions, which includes our case. It is, however, computationally inefficient in general (an efficient adaptation of the algorithm is given for the Hölder class of functions, to which our class of isotonic functions does not belong). In contrast, we achieve the optimal bound by using a simple and efficient Exponential Weights algorithm on a properly discretized version of our function class (interestingly, Gaillard and Gerchinovitz (2015) show that a general upper bound for Exponential Weights, which works for any totally bounded nonparametric class, is suboptimal).

2. Problem statement

Let $x_1 \le x_2 \le \ldots \le x_T$, be a set of T linearly ordered points (e.g., on the real line), denoted by X. We call a function $f\colon X \to \mathbb{R}$ *isotonic* (order-preserving) on X if $f(x_i) \le f(x_j)$ for any $x_i \le x_j$. Given data $(y_1, x_1), \ldots, (y_T, x_T)$, the isotonic regression problem is to find an isotonic f that minimizes $\sum_{t=1}^{T}(y_t - f(x_t))^2$, and the optimal such function is called the *isotonic regression function*.

We consider the online version of the isotonic regression problem. The adversary chooses $X = \{x_1, \ldots, x_T\}$ which is given in advance to the learner. In each trial $t = 1, \ldots, T$, the adversary picks a yet unlabeled point x_{i_t}, $i_t \in \{1, \ldots, T\}$ and the learner predicts with $\widehat{y}_{i_t} \in [0, 1]$. Then, the actual label $y_{i_t} \in [0, 1]$ is revealed, and the learner is penalized by the squared loss $(y_{i_t} - \widehat{y}_{i_t})^2$. Thus, the learner predicts at all points $x_1, \ldots x_T$ but in an adversarial order.

The goal of the learner is to have small regret, which is defined to be the difference of the cumulative loss and the cumulative loss of the best isotonic function in hindsight:

$$\mathrm{Reg}_T := \sum_{t=1}^{T}(y_{i_t} - \widehat{y}_{i_t})^2 - \min_{\text{isotonic } f} \sum_{t=1}^{T}(y_{i_t} - f(x_{i_t}))^2.$$

Note that neither the labels nor the learner's predictions are required to be isotonic on X. In what follows, we assume without loss of generality that $x_1 < x_2 < \ldots < x_T$, because equal consecutive points $x_i = x_{i+1}$ constrain the adversary ($f(x_i) = f(x_{i+1})$ for any function f) but not the learner.

Fixed-design. We now argue that without showing X to the learner in advance, the problem is hopeless; if the adversary can choose x_{i_t} online, any learning algorithm will suffer regret at least $\frac{1}{4}T$ (a linear regret implies very little learning is happening since playing randomly obtains linear regret). To see this, assume the adversary chooses $x_{i_1} = 0$; given learner's prediction $\widehat{y}_{i_1}$, the

At trial $t = 1 \dots T$:
Adversary chooses index i_t, such that $i_t \notin \{i_1, \dots, i_{t-1}\}$.
Learner predicts $\widehat{y}_{i_t} \in [0, 1]$.
Adversary reveals label $y_{i_t} \in [0, 1]$.
Learner suffers squared loss $(y_{i_t} - \widehat{y}_{i_t})^2$.

Figure 1: Online protocol for isotonic regression.

adversary can choose $y_{i_1} \in \{0, 1\}$ to cause loss at least $\frac{1}{4}$. Now, after playing round t, the adversary chooses $x_{i_{t+1}} = x_{i_t} - 2^{-t}$ if $y_{i_t} = 1$ or $x_{i_{t+1}} = x_{i_t} + 2^{-t}$ if $y_{i_t} = 0$. This allows the adversary to set $y_{i_{t+1}}$ to any value and still respect isotonicity. Regardless of $\widehat{y}_{i_{t+1}}$, the adversary inflicts loss at least $\frac{1}{4}$. This guarantees that if $y_{i_t} = 1$ then $x_{i_q} < x_{i_t}$ for all future points $q = t + 1, \dots, T$; similarly, if $y_{i_t} = 0$ then $x_{i_q} > x_{i_t}$ for all $q > t$. Hence, the label assignment is always isotonic on X, and the loss of the best isotonic function in hindsight is 0 (by choosing $f(x_i) = y_i$, $i = 1, \dots, T$) while the total loss of the learner is at least $\frac{1}{4}T$.

Thus, the learner needs to know X in advance. On the other hand, the particular values $x_i \in X$ do not play any role in this problem; it is only the order on X that matters. Thus, we may without loss of generality assume that $x_i = i$ and represent isotonic functions by vectors $\boldsymbol{f} = (f_1, \dots, f_T)$, where $f_i := f(i)$. We denote by $\mathcal{F}$ the set of all $[0, 1]$-valued isotonic functions:

$$\mathcal{F} = \{\boldsymbol{f} = (f_1, \dots, f_T) \colon 0 \le f_1 \le f_2 \le \dots \le f_T \le 1\}.$$

Using this notation, the protocol for online isotonic regression is presented in Figure 1.

We will use $\widehat{L}_T = \sum_{t=1}^{T} (y_t - \widehat{y}_t)^2$ to denote the total loss of the algorithm and $L_T(\boldsymbol{f}) = \sum_{t=1}^{T} (y_t - f_t)^2$ to denote the total loss of the isotonic function $\boldsymbol{f} \in \mathcal{F}$. The regret of the algorithm can then be concisely expressed as $\mathrm{Reg}_T = \widehat{L}_T - \min_{\boldsymbol{f} \in \mathcal{F}} L_T(\boldsymbol{f})$.

The offline solution. The classic solution to the isotonic regression problem is computed by the *Pool Adjacent Violators Algorithm* (PAVA) (Ayer et al., 1955). The algorithm is based on the observation that if the labels of any two consecutive points $i, i + 1$ violate isotonicity, then we must have $f_i^* = f_{i+1}^*$ in the optimal solution and we may merge both points to their average. This process repeats and terminates in at most T steps with the optimal solution. Efficient $O(T)$ time implementations exist (De Leeuw et al., 2009). There are two important properties of the isotonic regression function $\boldsymbol{f}^*$ that we will need later (Robertson et al., 1998):

1. The function $\boldsymbol{f}^*$ is piecewise constant and thus its level sets partition $\{1, \dots, T\}$.
2. The value of $\boldsymbol{f}^*$ on any level set is equal to the weighted average of labels within that set.

3. Blooper reel

The online isotonic regression problem concerns minimizing a convex loss function over the convex class of isotonic functions. Hence, the problem can be analyzed with online convex optimization tools (Shalev-Shwartz, 2012). Unfortunately, we find that most of the common online learning algorithms completely fail on the isotonic regression problem in the sense of giving linear regret

Algorithm	General bound	Bound for online IR
Online GD	$G_2 D_2 \sqrt{T}$	T
EG	$G_\infty D_1 \sqrt{T \log d}$	$\sqrt{T \log T}$
FTL	$G_2 D_2 d \log T$	$T^2 \log T$
Exponential Weights	$d \log T$	$T \log T$

Table 1: Comparison of general bounds as well as bounds specialized to online isotonic regression for various standard online learning algorithms. For general bounds, d denotes the dimension of the parameter vector (equal to T for this problem), G_p is the bound on the L_p-norm of the loss gradient, and D_q is the bound on the L_q-norm of the parameter vector. Bounds for FTL and Exponential Weights exploit the fact that the square loss is $\frac{1}{2}$-exp-concave (Cesa-Bianchi and Lugosi, 2006).

guarantees or, at best, suboptimal rates of $O(\sqrt{T})$; see Table 1. We believe that the fact that most standard approaches fail makes the considered problem particularly interesting and challenging.

In the usual formulation of online convex optimization, for trials $t = 1, \dots, T$, the learner predicts with a parameter vector $\boldsymbol{w}_t \in \mathbb{R}^d$, the adversary reveals a convex loss function ℓ_t, and the learner suffers loss $\ell_t(\boldsymbol{w}_t)$. To cast our problem in this framework, we set the prediction of the learner at trial t to $\widehat{y}_{i_t} = \boldsymbol{w}_t^\intercal \boldsymbol{x}_{i_t}$ and the loss to $\ell_t(\boldsymbol{w}_t) = (y_{i_t} - \boldsymbol{w}_t^\intercal \boldsymbol{x}_{i_t})^2$. There are two natural ways to parameterize $\boldsymbol{w}_t, \boldsymbol{x}_{i_t} \in \mathbb{R}^d$:

1. The learner predicts some $\boldsymbol{f} \in \mathcal{F}$ and sets $\boldsymbol{w} = \boldsymbol{f}$. Then, $\boldsymbol{x}_i$ is the i-th unit vector (with i-th coordinate equal to 1 and the remaining coordinates equal to 0). Note that $\sup_{\boldsymbol{w}} \|\boldsymbol{w}\|_2 = \sqrt{T}$ and $\|\nabla \ell(\boldsymbol{w})\|_2 \le 2$ in this parameterization.

2. The learner predicts some $\boldsymbol{f} \in \mathcal{F}$ and sets $\boldsymbol{w} = (f_1 - f_0, f_2 - f_1, \dots, f_{T+1} - f_T) \in \mathbb{R}^{T+1}$, i.e. the vector of differences of $\boldsymbol{f}$ (we used two dummy variables $f_0 = 0$ and $f_{T+1} = 1$); then, $\boldsymbol{x}_i$ has the first i coordinates equal to 1 and the last $T - i$ coordinates equal to 0. Note that $\|\boldsymbol{w}\|_1 = 1$, $\|\nabla \ell(\boldsymbol{w})\|_\infty \le 2$, but $\sup_{y, \boldsymbol{w}} \|\nabla \ell(\boldsymbol{w})\|_2 = 2\sqrt{T}$.

Table 1 lists the general bounds and their specialization to online isotonic regression for several standard online learning algorithms: Online Gradient Descent (GD) (Zinkevich, 2003), Exponentiated Gradient (EG) Kivinen and Warmuth (1997) when applied to exp-concave losses (which include squared loss, see Cesa-Bianchi and Lugosi 2006), Follow the Leader[1], and Exponential Weights (Hazan et al., 2007). EG is assumed to be used in the second parameterization, while the bounds for the remaining algorithms apply to both parameterizations (since $G_2 D_2 = \Omega(\sqrt{T})$ in both cases).

EG is the only algorithm that provides a meaningful bound of order $O(\sqrt{T \log T})$, as shown in Appendix A. All the other bounds are vacuous (linear in T or worse). This fact does not completely rule out these algorithms since we do not know a priori whether their bounds are tight in the worst case for isotonic regression. Next we will exhibit sequences of outcomes that cause GD, FTL and Exponential Weights to incur linear regret.

1. The Online Newton algorithm introduced by Hazan et al. (2007) is equivalent to FTL for squared loss.

Theorem 1 *For any learning rate $\eta \geq 0$ and any initial parameter vector $\boldsymbol{f}_1 \in \mathcal{F}$, the Online Gradient Descent algorithm, defined as*

$$\boldsymbol{f}_t = \operatorname*{argmin}_{\boldsymbol{f} \in \mathcal{F}} \left\{ \frac{1}{2} \|\boldsymbol{f} - \boldsymbol{f}_{t-1}\|^2 + 2\eta (f_{t-1,i_{t-1}} - y_{i_{t-1}}) f_{i_{t-1}} \right\},$$

suffers at least $\frac{T}{4}$ regret in the worst case.

Proof The adversary reveals the labels in isotonic order ($i_t = t$ for all t), and all the labels are zero. Then, $\ell_t(\boldsymbol{f}_t) = \ell_t(\boldsymbol{f}_1)$, and the total loss of the algorithm $\widehat{L}_T$ is equal to the loss of the initial parameter vector: $\widehat{L}_T = L_T(\boldsymbol{f}_1) = \sum_t f_{1,t}^2$. This follows from the fact that $\boldsymbol{f}_t$ and $\boldsymbol{f}_{t-1}$ can only differ on the first $t-1$ coordinates ($f_{t,q} = f_{t-1,q}$ for $q \geq t$) so only the coordinates of the already labeled points are updated. To see this, note that the parameter update can be decomposed into the "descent" step $\widetilde{\boldsymbol{f}}_t = \boldsymbol{f}_{t-1} - 2\eta f_{t-1,t-1} \boldsymbol{e}_{t-1}$ (where $\boldsymbol{e}_i$ is the i-th unit vector), and the "projection" step $\boldsymbol{f}_t = \operatorname{argmin}_{\boldsymbol{f} \in \mathcal{F}} \|\boldsymbol{f} - \widetilde{\boldsymbol{f}}_t\|^2$ (which is actually the isotonic regression problem). The descent step decreases $(t-1)$-th coordinate by some amount and leaves the remaining coordinates intact. Since $\boldsymbol{f}_{t-1}$ is isotonic, $\widetilde{f}_{t,t} \leq \ldots \leq \widetilde{f}_{t,T}$ and $\widetilde{f}_{t,q} \leq \widetilde{f}_{t,t}$ for all $q < t$. Hence, the projection step will only affect the first $t-1$ coordinates.

By symmetry, one can show that when the adversary reveals the labels in *antitonic* order ($i_t = T - t + 1$ for all t), and all the labels are 1, then $\widehat{L}_T = \sum_t (1 - f_{1,t})^2$. Since $f_{1,t}^2 + (1 - f_{1,t})^2 \geq \frac{1}{2}$ for any $f_{1,t}$, the loss suffered by the algorithm on one of these sequences is at least $\frac{T}{4}$. ■

Theorem 2 *For any regularization parameter $\lambda > 0$ and any regularization center $\boldsymbol{f}_0 \in \mathcal{F}$, the Follow the (Regularized) Leader algorithm defined as:*

$$\boldsymbol{f}_t = \operatorname*{argmin}_{\boldsymbol{f} \in \mathcal{F}} \left\{ \lambda \|\boldsymbol{f} - \boldsymbol{f}_0\|^2 + \sum_{q=1}^{t-1} (f_{i_q} - y_{i_q})^2 \right\},$$

suffers at least $\frac{T}{4}$ regret in the worst case.

Proof The proof uses exactly the same arguments as the proof of Theorem 1: If the adversary reveals labels equal to 0 in isotonic order, or labels equal to 1 in antitonic order, then $f_{t,t} = f_{0,t}$ for all t. This is because the constraints in the minimization problem are never active (argmin over $\boldsymbol{f} \in \mathbb{R}^T$ returns an isotonic function). ■

We used a regularized version of FTL in Theorem 2, because otherwise FTL does not give unique predictions for unlabeled points.

Theorem 3 *The Exponential Weights algorithm defined as:*

$$\boldsymbol{f}_t = \int_{\mathcal{F}} \boldsymbol{f} p_t(\boldsymbol{f}) \, \mathrm{d}\mu(\boldsymbol{f}), \qquad \text{where} \quad p_t(\boldsymbol{f}) = \frac{e^{-\frac{1}{2}\sum_{q=1}^{t-1} (f_{i_q} - y_{i_q})^2}}{\int_{\mathcal{F}} e^{-\frac{1}{2}\sum_{q=1}^{t-1} (f_{i_q} - y_{i_q})^2} \, \mathrm{d}\mu(\boldsymbol{f})},$$

with μ being the uniform (Lebesgue) measure over $\mathcal{F}$, suffers regret $\Omega(T)$ in the worst case.

The proof of Theorem 3 is long and is deferred to Appendix B.

4. Optimal algorithm

We have hopefully provided a convincing case that many of the standard online approaches do not work for online isotonic regression. Is this section, we present an algorithm that does: Exponential Weights over a discretized version of $\mathcal{F}$. We show that it achieves $O(T^{1/3}(\log T)^{2/3})$ regret which matches, up to log factors, the $\Omega(T^{1/3})$ lower bound we prove in the next section.

The basic idea is to form a covering net of all isotonic functions by discretizing $\mathcal{F}$ with resolution $\frac{1}{K}$, to then play Exponential Weights on this covering net with a uniform prior, and to tune K to get the best bound. We take as our covering net $\mathcal{F}_K \subset \mathcal{F}$ the set of isotonic functions which take values of the form $\frac{k}{K}$, $k = 0, \dots, K$, i.e.

$$\mathcal{F}_K := \left\{ \boldsymbol{f} \in \mathcal{F} \colon f_t = \frac{k_t}{K} \text{ for some } k_t \in \{0, \dots, K\},\ k_1 \le \dots \le k_T \right\}.$$

Note that $\mathcal{F}_K$ is finite. In fact $|\mathcal{F}_K| = \binom{T+K}{K}$, since the enumeration of all isotonic function in $\mathcal{F}_K$ is equal to the number of ways to distribute the K possible increments among bins $[0,1), \dots, [T-1, T), [T, T+1)$. The first and last bin are to allow for isotonic functions that start and end at arbitrary values. It is a well known fact from combinatorics that there are $\binom{T+K}{K}$ ways to allocate K items into $T+1$ bins, (see, e.g., DeTemple and Webb, 2014, section 2.4).

The algorithm we propose is the Exponential Weights algorithm over this covering net; at round t, each $\boldsymbol{f}$ in $\mathcal{F}_K$ is given weight $e^{-\frac{1}{2}\sum_{q=1}^{t-1}(f_{i_q} - y_{i_q})^2}$ and we play the weighted average of f_{i_t}. An efficient implementation is given in Algorithm 1.

Theorem 4 *Using* $K = \left\lceil \left(\frac{T}{4\log(T+1)} \right)^{1/3} \right\rceil$, *the regret of Exponential Weights with the uniform prior on the covering net* $\mathcal{F}_K$ *has regret bounded by:*

$$\mathrm{Reg}_T \le \frac{3}{2^{2/3}} T^{1/3} \left(\log(T+1)\right)^{2/3} + 2\log(T+1).$$

Proof Due to exp-concavity of the squared loss, running Exponential Weights with $\eta = 1/2$ guarantees that:

$$\widehat{L}_T - \min_{\boldsymbol{f} \in \mathcal{F}_K} L_T(\boldsymbol{f}) \le \frac{\log |\mathcal{F}_K|}{\eta} = 2\log |\mathcal{F}_K|,$$

(see, e.g., Cesa-Bianchi and Lugosi, 2006, Proposition 3.1).

Let $\boldsymbol{f}^* = \mathrm{argmin}_{\boldsymbol{f} \in \mathcal{F}} L_T(\boldsymbol{f})$ be the isotonic regression function. The regret is

$$\begin{aligned} \mathrm{Reg} &= \widehat{L}_T - L_T(\boldsymbol{f}^*) \\ &= \widehat{L}_T - \min_{\boldsymbol{f} \in \mathcal{F}_K} L_T(\boldsymbol{f}) + \underbrace{\min_{\boldsymbol{f} \in \mathcal{F}_K} L_T(\boldsymbol{f}) - L_T(\boldsymbol{f}^*)}_{:= \Delta_K}. \end{aligned}$$

Let us start with bounding Δ_K. Let $\boldsymbol{f}^+$ be a function obtained from $\boldsymbol{f}^*$ by rounding each value f_t^* to the nearest number of the form $\frac{k_t}{K}$ for some $k_t \in \{0, \dots, K\}$. It follows that $\boldsymbol{f}^+ \in \mathcal{F}_K$ and $\Delta_K \le L_T(\boldsymbol{f}^+) - L_T(\boldsymbol{f}^*)$. Using $\ell_t(x) := (y_t - x)^2$, we have

$$\ell_t(f_t^+) - \ell_t(f_t^*) = (y_t - f_t^+)^2 - (y_t - f_t^*)^2 = (f_t^+ - f_t^*)(f_t^+ + f_t^* - 2y_t). \tag{1}$$

Let $\mathcal{T}_c = \{t\colon f^*_t = c\}$ be the level set of the isotonic regression function. It is known (Robertson et al., 1998, see also Section 2) that (as long as $|\mathcal{T}_c| > 0$):

$$\frac{1}{|\mathcal{T}_c|}\sum_{t\in\mathcal{T}_c} y_t = f^*_t = c, \tag{2}$$

i.e., the isotonic regression function is equal to the average over all labels within each level set. Now, choose any level set $\mathcal{T}_c$ with $|\mathcal{T}_c| > 0$. Note that $\boldsymbol{f}^+$ is also constant on $\mathcal{T}_c$ and denote its value by c^+. Summing (1) over $\mathcal{T}_c$ gives:

$$\begin{aligned}
\sum_{t\in\mathcal{T}_c} \ell_t(f^+_t) - \ell_t(f^*_t) &= \sum_{t\in\mathcal{T}_c} (c^+ - c)(c^+ + c - 2y_t) \\
&= |\mathcal{T}_c|(c^+ - c)(c^+ + c) - 2(c^+ - c)\sum_{t\in\mathcal{T}_c} y_t \\
\text{(from (2))} \qquad &= |\mathcal{T}_c|(c^+ - c)(c^+ + c) - 2|\mathcal{T}_c|(c^+ - c)c \\
&= |\mathcal{T}_c|(c^+ - c)^2 \\
&= \sum_{t\in\mathcal{T}_c} (f^+_t - f^*_t)^2.
\end{aligned}$$

Since for any t, $|f^+_t - f^*_t| \le \frac{1}{2K}$, we can sum over the level sets of $\boldsymbol{f}^*$ to bound Δ_K:

$$\Delta_K \le L_T(\boldsymbol{f}^+) - L_T(\boldsymbol{f}^*) \le \sum_{t=1}^{T} \ell_t(f^+_t) - \ell_t(f^*_f) = \sum_{t=1}^{T} (f^+_t - f^*_t)^2 \le \frac{T}{4K^2}.$$

Combining these two bounds, we get:

$$\mathrm{Reg}_T \le 2\log|\mathcal{F}_K| + \frac{T}{4K^2} \le 2K\log(T+1) + \frac{T}{4K^2},$$

where we used $|\mathcal{F}_K| = \binom{T+K}{K} \le (T+1)^K$.[2] Optimizing the bound over K by setting the derivative to 0 gives $K^* = \left(\frac{T}{4\log(T+1)}\right)^{1/3}$. Taking $K = \lceil K^* \rceil$ and plugging it in into the bound gives:

$$\mathrm{Reg}_T \;\le\; 2(K^*+1)\log(T+1) + \frac{T}{4(K^*)^2} \;\le\; \frac{3}{2^{2/3}} T^{1/3}\left(\log(T+1)\right)^{2/3} + 2\log(T+1),$$

where we used $K^* \le K \le K^* + 1$. ■

We note that instead of predicting with weighted average over the discretized functions, one can make use of the fact that the squared loss is 2-mixable and apply the prediction rule of the Aggregating Forecaster (Vovk, 1990; Cesa-Bianchi and Lugosi, 2006, Section 3.6). This would let us run the algorithm with $\eta = 2$ and improve the leading constant in the regret bound to $\frac{3}{4}$.

The importance of being discrete. Surprisingly, playing weighted averages over $\mathcal{F}$ does not work (Theorem 3), but playing over a covering net does. Indeed, the uniform prior exhibits wild behavior by concentrating all mass around the "diagonal" monotonic function with constant slope $1/T$, whereas the discretized version with the suggested tuning for K still has non-negligible mass everywhere.

2. $\binom{T+K}{K} = \frac{(T+1)\cdot\ldots\cdot(T+K)}{1\cdot\ldots\cdot K}$; we get the bound by noticing that $\frac{T+k}{k} \le T+1$ for $k \ge 1$.

Algorithm 1: Efficient Exponential Weights on the covering net

Input: Game length T, discretization K
Initialize $\beta_s^j = 1$ for all $s = 1, \dots, T,\ j = 0, \dots, K$;
for $t = 1, \dots, T$ **do**
  Receive i_t;
  Initialize $w_1^k = \beta_1^k$ and $v_T^k = \beta_T^k$ for all $k = 0, \dots, K$;
  for $s = 2, \dots, i_t$ **do**
    $w_s^k \leftarrow [\![k > 0]\!] w_s^{k-1} + \beta_{s-1}^k w_{s-1}^k$ for all $k = 0, \dots, K$;
  end
  for $s = T-1, \dots, i_t$ **do**
    $v_s^k \leftarrow [\![k < K]\!] w_s^{k+1} + \beta_{s+1}^k v_{s+1}^k$ for all $k = K, \dots, 0$;
  end
  $\hat{y}_{i_t} \leftarrow \frac{\sum_{k=0}^K \frac{k}{K} w_{i_t}^k v_{i_t}^k}{\sum_{k=0}^K w_{i_t}^k v_{i_t}^k}$;
  Receive y_{i_t} and update $\beta_{i_t}^j = e^{-\frac{1}{2}(\frac{j}{K} - y_{i_t})^2}$ for all $j = 0, \dots, K$;
end

Comparison with online nonparametric regression. We compare our approach to the work of Rakhlin and Sridharan (2014) and Gaillard and Gerchinovitz (2015), which provide general upper bounds on the minimax regret expressed by means of the sequential and metric entropies of the function class under study. It turns out that we can use our covering net to show that the metric entropy $\log \mathcal{N}_2(\beta, \mathcal{F}, T)$, as well as the sequential entropy $\log \mathcal{N}_\infty(\beta, \mathcal{F}, T)$, of the class of isotonic functions are bounded by $O(\beta^{-1} \log T)$; this implies (by following the proof of Theorem 2 of Rakhlin and Sridharan, 2014, and by Theorem 2 of Gaillard and Gerchinovitz, 2015) that the minimax regret is bounded by $O(T^{1/3}(\log T)^{2/3})$, which matches our result up to a constant. Note, however,that the bound of Rakhlin and Sridharan (2014) is nonconstructive, while ours is achieved by an efficient algorithm. The bound of Gaillard and Gerchinovitz (2015) follows from applying the Chaining Exponentially Weighted Average Forecaster, that is based on aggregation on two levels: On the first level a multi-variable version of Exponentiated Gradient is used, while on the second level the Exponential Weights algorithm is used. The algorithm is, however, computationally inefficient in general, and it is not clear whether an efficient adaptation to the class of isotonic functions can easily be constructed. In contrast, we achieve the optimal bound by using a simple and efficient Exponential Weights algorithm on a properly discretized version of our function class; the chaining step turns out to be unnecessary for the class of isotonic functions due to the averaging property (2) of the isotonic regression function.

4.1. An Efficient implementation

A naïve implementation of exponential averaging has an intractable complexity of $O(|\mathcal{F}_k|)$ per round. Fortunately, one can use dynamic programming to derive an efficient implicit weight update that is able to predict in $O(TK)$ time per round for arbitrary prediction orders and $O(K)$ per round when predicting in isotonic order. See Algorithm 1 for pseudocode.

Say we currently need to predict at i_t. We can compute the Exponential Weights prediction by dynamic programming: for each $k = 0, \dots, K$, let

$$w_s^k \;=\; \sum_{0 \le f_1 \le \dots \le f_s = \frac{k}{K}} e^{-\frac{1}{2}\sum_{q<t:i_q<s}(f_{i_q}-y_{i_q})^2} \quad \text{and} \quad v_s^k \;=\; \sum_{\frac{k}{K} = f_s \le \dots \le f_T \le 1} e^{-\frac{1}{2}\sum_{q<t:i_q>s}(f_{i_q}-y_{i_q})^2},$$

so that the exponentially weighted average prediction is

$$\hat{y}_{i_t} \;=\; \frac{\sum_{\boldsymbol{f}\in\mathcal{F}_K} f_{i_t} e^{-\frac{1}{2}\sum_{q<t}(f_{i_q}-y_{i_q})^2}}{\sum_{\boldsymbol{f}\in\mathcal{F}_K} e^{-\frac{1}{2}\sum_{q<t}(f_{i_q}-y_{i_q})^2}} \;=\; \frac{\sum_{k=0}^K \frac{k}{K} w_{i_t}^k v_{i_t}^k}{\sum_{k=0}^K w_{i_t}^k v_{i_t}^k}.$$

Now we can compute the w_s^k in one sweep from $s = 1$ to $s = i_t$ for all $k = 0, \dots, K$. If we define $\beta_s^j \;=\; e^{-\frac{1}{2}(\frac{j}{K}-y_s)^2}$ if $s \in \{i_1, \dots, i_{t-1}\}$ and 1 otherwise, we can calculate w_s^k by starting with $w_1^k = \beta_1^k$ and then sweeping right:

$$\begin{aligned} w_{s+1}^k \;&=\; \sum_{0 \le f_1 \le \dots \le f_{s+1} = \frac{k}{K}} e^{-\frac{1}{2}\sum_{q<t:i_q\le s}(f_{i_q}-y_{i_q})^2} \\ &=\; \sum_{0\le j\le k} \beta_s^j \sum_{0 \le f_1 \le \dots \le f_s = \frac{j}{K}} e^{-\frac{1}{2}\sum_{q<t:i_q<s}(f_{i_q}-y_{i_q})^2} \\ &=\; \sum_{0\le j\le k} \beta_s^j w_s^j. \end{aligned}$$

The equations for v_s^k are updated symmetrically right-to-left, which gives an $O(TK^2)$ per round algorithm. We can speed it up to $O(TK)$ by using

$$w_{s+1}^{k+1} \;=\; \sum_{0\le j\le k} \beta_s^j w_s^j + \beta_s^{k+1} w_s^{k+1} \;=\; w_{s+1}^k + \beta_s^{k+1} w_s^{k+1},$$

and similarly for v_{s+1}^{k+1}.

Acceleration for predicting in isotonic order. When working in isotonic order (meaning $i_t = t$), we can speed up the computation to $O(K)$ per round (independent of T) by the following tricks. First, we do not need to spend work maintaining v_t^k as they satisfy $v_t^k = \binom{T-t+K-k}{K-k}$. Moreover, between rounds $t-1$ and t the w_s^k do not change for $s < t$, and we only need to compute w_t^k for all k, hence speeding up the computation to $O(K)$ per round.

5. Lower bound

Theorem 5 *All algorithms must suffer*

$$\mathrm{Reg}_T = \Omega(T^{1/3}).$$

The full proof is given in Appendix C.

Proof (sketch) We proceed by constructing the difficult sequence explicitly. Split the T points $(1, \dots, T)$ into K consecutive segments $(1, \dots, m), (m+1, \dots, 2m), \dots, (m(K-1)+1, \dots, T)$,

where in each segment there are $m = \frac{T}{K}$ consecutive points (for simplicity assume T is divisible by K). Let $t \in k$ mean that t is in the k-th segment, $k = 1, \ldots, K$. Now, suppose the adversary generates the labels i.i.d. with $y_t \sim \text{Bernoulli}(p_k)$ when $t \in k$, and $p_1 \leq \ldots \leq p_K$. The total loss of the best isotonic function is then bounded above by the total loss of the constant function equal to p_k in each segment, hence the expected regret of any algorithm can be lower-bounded by $\mathbb{E}[\text{Reg}_T] \geq \sum_{k=1}^{K} \mathbb{E}\Big[\sum_{t \in k} (\widehat{y}_t - p_k)^2\Big]$. In each segment, the adversary picks $p_k \in \{p_{k,0}, p_{k,1}\}$, where $p_{k,0} = \frac{1}{4} + \frac{k-1}{2K}$ and $p_{k,1} = \frac{1}{4} + \frac{k}{2K}$, which guarantees that for any choice of the the adversary $p_1 \leq \ldots \leq p_K$. We then show that the expected regret can be lower-bounded by:

$$\mathbb{E}[\text{Reg}_T] = \sum_{k=1}^{K} \sum_{t \in k} \mathbb{E}[(\widehat{y}_t - p_k)^2] \geq \frac{m}{4} \sum_{k=1}^{K} \mathbb{E}[(\widehat{p}_k - p_k)^2],$$

where $\widehat{p}_k \in \{p_{k,0}, p_{k,1}\}$ depends on the predictions $\{\widehat{y}_t\}_{t \in k}$ (and hence on the data), but not on the probabilities p_k. We use Assouads lemma (Yu, 1997; Tsybakov, 2009) to bound the sum on the right-hand side:

$$\max_{p_1, \ldots, p_K \,:\, p_k \in \{p_{k,0}, p_{k,1}\}} \sum_{k=1}^{K} \mathbb{E}[(\widehat{p}_k - p_k)^2] \geq \frac{1}{8K} \left(1 - \frac{\sqrt{m}}{\sqrt{3}K}\right).$$

Using $m = \frac{T}{K}$ and tuning the number of segments to $K = \Theta(T^{1/3})$ to optimize the bound, gives $\Omega(T^{1/3})$ lower bound on the worst-case regret. ■

We note that an analogous lower bound of the form $\Omega(T^{-2/3})$ is known in the statistical literature on isotonic regression as a lower bound for the statistical risk $\mathbb{E}[\frac{1}{T}\|\widehat{f} - f\|^2]$ of any estimator $\widehat{f}$ in the fixed-design setup under the standard i.i.d. noise assumption (see Zhang, 2002, for a brief overview of the lower and upper bounds in this setting). This shows that the online version of the problem is not fundamentally harder (up to a logarithmic factor) than the batch (statistical) version.

6. Noise-free case

In this section, we are concerned with a particular case of "easy data", when the labels revealed by the adversary are actually isotonic: $y_1 \leq y_2 \leq \ldots \leq y_T$, so that the loss of the best isotonic function is zero. We show that the achievable worst-case regret in this case scales only logarithmically in T. Furthermore, if we additionally assume that the labels are revealed in isotonic order, the achievable worst-case regret is bounded by 1. Interestingly, we were able to determine the minimax algorithm, and the exact value of the minimax regret in both cases. Our findings are summarized in the two theorems below. The proofs and the minimax predictions are given in Appendix D.

Theorem 6 *Assume the labels revealed by the adversary are isotonic. Then, the regret of the minimax algorithm is bounded above by:*

$$\text{Reg}_T \leq \frac{1}{4} \log_2(T+1).$$

Furthermore, when $T = 2^k - 1$ for some positive integer k, any algorithm suffers regret at least $\frac{1}{4} \log_2(T+1)$.

Theorem 7 *Assume the labels are isotonic, and they are revealed in isotonic order ($i_t = t$ for all t). Then, the regret of the minimax algorithm is bounded above by:*

$$\mathrm{Reg}_T \le \alpha_T \le 1,$$

where α_T is defined recursively as: $\alpha_1 = \frac{1}{4}$ and $\alpha_t = \left(\frac{\alpha_{t-1}+1}{2}\right)^2$. Furthermore, any algorithm suffers regret at least α_T.

Finally, we note that the logarithmic regret can also be obtained by using the Exponentiated Gradient algorithm with its learning rate tuned for the noise-free case (see Appendix A and Kivinen and Warmuth, 1997, for details).

7. Other loss functions

We discuss extensions of the isotonic regression problem where the squared loss is replaced by the entropic loss and the absolute loss respectively.

7.1. Entropic loss

The entropic loss, defined for $y, \widehat{y} \in [0,1]$ by $\ell(y,\widehat{y}) = -y\log\widehat{y} - (1-y)\log(1-\widehat{y})$, plays an important role in isotonic regression, as its minimization is equivalent to maximum likelihood estimation for Bernoulli distributions under isotonic constraints (Robertson et al., 1998). It is convenient to replace the entropic loss by the relative entropy $D_\phi(y\|\widehat{y}) = \phi(y) - \phi(\widehat{y}) - (y-\widehat{y})^\intercal\phi'(\widehat{y})$, which is the Bregman divergence generated by $\phi(y) = -y\log y - (1-y)\log(1-y)$, the binary entropy. A surprising fact in isotonic regression is that minimizing the sum of Bregman divergences $\sum_t D_\phi(y_t\|f_t)$ in the class of isotonic functions $\boldsymbol{f} \in \mathcal{F}$ leads to the same optimal solution, no matter what ϕ is: the isotonic regression function $\boldsymbol{f}^*$ (Robertson et al., 1998).

Since the entropic loss is 1-exp-concave (Cesa-Bianchi and Lugosi, 2006, page 46), we may use the Exponential Weights algorithm on the discretized class of functions:

$$\mathcal{F}_K = \left\{\boldsymbol{f} \in \mathcal{F} \colon \forall t, f_t \in \{z_0, z_1, \ldots, z_K\}\right\}$$

(we now use a non-uniform discretization $\{z_0, z_1, \ldots, z_K\}$, to be specified later). Following the steps of the proof of Theorem 4, we obtain a regret bound:

$$\mathrm{Reg}_T \le \log\binom{T+K}{K} + L_T(\boldsymbol{f}^+) - L_T(\boldsymbol{f}^*),$$

where $L_T(\boldsymbol{f}) = \sum_{t=1}^T D_\phi(y_t\|f_t)$, $\boldsymbol{f}^+$ is defined by: $f_t^+ = \operatorname{argmin}_{z\in\{z_0,z_1,\ldots,z_K\}} D_\phi(f_t^*\|z)$, and we used the fact that the isotonic regression function $\boldsymbol{f}^*$ minimizes $L_T(\boldsymbol{f})$ over $\mathcal{F}$ (see above). Let $\mathcal{T}_c = \{t\colon f_t^* = c\}$ be a non-empty level set of $\boldsymbol{f}^*$. Using the averaging property (2) of $\boldsymbol{f}^*$, and the

fact that $\boldsymbol{f}^+$ is constant on $\mathcal{T}_c$ (denote its value by c^+), we have:

$$\begin{aligned}\sum_{t\in\mathcal{T}_c} D_\phi(y_t\|f_t^+) - D_\phi(y_t\|f_t^*) &= \sum_{t\in\mathcal{T}_c} \phi(c) - \phi(c^+) - (y_t - c^+)\phi'(c^+) + (y_t - c)\phi(c) \\ &= |\mathcal{T}_c| D_\phi(c\|c^+) + (\phi'(c) - \phi'(c^+)) \sum_{t\in\mathcal{T}_c} (y_t - c) \\ \text{(from (2))} \qquad &= |\mathcal{T}_c| D_\phi(c\|c^+) \\ &= \sum_{t\in\mathcal{T}_c} D_\phi(f_t^*\|f_t^+).\end{aligned}$$

Summing over the level sets gives $L_T(\boldsymbol{f}^+) - L_T(\boldsymbol{f}^*) = \sum_t D_\phi(f_t^*\|f_t^+)$. To introduce the appropriate discretization points, we follow (De Rooij and Van Erven, 2009). For any $y \in [0,1]$ and $\psi \in [0, \pi/2]$, we let $\psi(y) = \arcsin\sqrt{y}$, so that $y = \sin^2(\psi)$. The parameterization ψ has a nice property, that the values of D_ϕ on uniformly located neighboring points are also close to uniform. We discretize the interval $[0, \pi/2]$ into $K+1$ points $\{\psi_0, \dots, \psi_K\} = \left\{\frac{\pi}{2K}, \dots, \frac{\pi(K-1)}{2K}\right\} \cup \left\{\frac{\pi}{4K}, \frac{\pi}{2} - \frac{\pi}{4K}\right\}$, which is almost uniform, with two additional points on the boundaries. Then, we define $z_k = y(\psi_k) = \sin^2(\psi_k)$, $k = 0, \dots, K$. Using Lemma 4 from De Rooij and Van Erven (2009):

$$D_\phi(f_t^*\|f_t^+) \le \frac{(2-\sqrt{2})\pi^2}{K^2},$$

which bounds $L_T(\boldsymbol{f}^+) - L_T(\boldsymbol{f}^*)$ by $(2-\sqrt{2})\pi^2 \frac{T}{K^2}$. From now on we proceed as in the proof of Theorem 4, to get $O(T^{1/3}\log^{2/3}(T))$ bound. Thus, we showed:

Theorem 8 *Using* $K = \left\lceil \left(\frac{2(2-\sqrt{2})\pi^2 T}{\log(T+1)} \right)^{1/3} \right\rceil$, *the entropic loss regret of discretized Exponential Wights on the covering net:*

$$\mathcal{F}_K = \left\{ \boldsymbol{f} \in \mathcal{F} \colon \forall t, f_t \in \{z_0, z_1, \dots, z_K\} \right\},$$

where $z_0 = \sin^2(\frac{\pi}{4K})$, $z_K = \cos^2(\frac{\pi}{4K})$, *and* $z_k = \sin^2(\frac{\pi k}{2K})$ *for* $k = 1, \dots, K-1$, *has the following upper bound:*

$$\mathrm{Reg}_T \le \frac{3(2-\sqrt{2})^{1/3}\pi^{2/3}}{2^{2/3}} T^{1/3} \left(\log(T+1)\right)^{2/3} + 2\log(T+1).$$

7.2. Absolute loss

Absolute loss $|\widehat{y}_{i_t} - y_{i_t}|$ is a popular loss function in modeling data with isotonic functions, especially in the context of isotonic discrimination/classification (Dykstra et al., 1999; Kotłowski and Słowinski, 2013). However, the online version of this problem turns out to be rather uninteresting for us, since it can be solved in an essentially optimal way (up to an $O(\sqrt{\log T})$ factor) by using the vanilla Exponentiated Gradient algorithm. Applying the standard EG regret bound (Kivinen and Warmuth, 1997; Cesa-Bianchi and Lugosi, 2006, also c.f. Section 3) results in a $O(\sqrt{T\log T})$ bound, whereas a lower bound of $\Omega(\sqrt{T})$ comes from the setting of prediction with expert advice (Cesa-Bianchi and Lugosi, 2006): we constrain the adversary to only play with one of the two constant (isotonic) functions $f_t \equiv 0$ or $f_t \equiv 1$, and apply the standard lower bound for the 2-experts case.

8. Conclusions and open problem

We introduced the online version of the isotonic regression problem, in which the learner must sequentially predict the labels as well as the best isotonic function. We gave a computationally efficient version of the Exponential Weights algorithm which plays on a covering net for the set of isotonic functions and proved that its regret is bounded by $O(T^{1/3}\log^{2/3}(T))$. We also showed an $\Omega(T^{1/3})$ lower bound on the regret of any algorithm, essentially closing the gap.

There are some interesting directions for future research. First, we believe that the discretization (covering net) is not needed in the algorithm, and a carefully devised continuous prior would work as well. We were, however, unable to find a prior that would produce the optimal regret bound and remain computationally efficient. Second, we are interested to see whether some regularized version of FTL (e.g., by means of relative entropy), or the *forward algorithm* (Vovk-Azoury-Warmuth) (Azoury and Warmuth, 2001) could work for this problem. However, the most interesting research direction is the extension to the partial order case. In this setting, the learner is given a set of points $X = \{x_1, \ldots, x_T\}$, together with a partial order relation $\preceq$ on X. The goal of the learner is to sequentially predict the labels not much worse than the best function which respects the isotonic constraints: $x_i \preceq x_j \rightarrow f(x_i) \leq f(x_j)$. A typical application would be nonparametric data modeling with multiple features, where domain knowledge may tell us that increasing the value of any of the features is likely to increase the value of the label. The off-line counterpart has been extensively studied in the statistics literature (Robertson et al., 1998), and the optimal isotonic function shares many properties (e.g., averaging within level sets) with the linear order case. The discretized Exponential Weights algorithm, which was presented in this paper, can be extended to deal with partial orders. The analysis closely follows the proof of Theorem 4 except that the size of the covering net $\mathcal{F}_K$ is no longer $O(T^K)$ but now depends on the structure of $\preceq$. We believe that $|\mathcal{F}_K|$ is the right quantity to measure the complexity of the problem and the algorithm will remain competitive in this more general setting. Unfortunately, the algorithm is no longer efficiently implementable and suffers from the same problems that plague inference in graphical models on general graphs. It thus remains an open problem to find an efficient algorithm for the partial order case.

Acknowledgments

We thank the anonymous reviewers for suggestions which improved the quality of our work. Wouter Koolen acknowledges support from the Netherlands Organization for Scientific Research (NWO, Veni grant 639.021.439), Wojciech Kotłowski acknowledges support from the Polish National Science Centre (grant no. 2013/11/D/ST6/03050), and Alan Malek acknowledges support from Adobe through a Digital Marketing Research Award.

References

M. Ayer, H. D. Brunk, G. M. Ewing, W. T. Reid, and E. Silverman. An empirical distribution function for sampling with incomplete information. *Annals of Mathematical Statistics*, 26(4): 641–647, 1955.

K. Azoury and M. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Journal of Machine Learning*, 43(3):211–246, 2001.

L. Birgé and P. Massart. Rates of convergence for minimum contrast estimators. *Probability Theory and Related Fields*, 97:113–150, 1993.

H. D. Brunk. Maximum likelihood estimates of monotone parameters. *Annals of Mathematical Statistics*, 26(4):607–616, 1955.

N. Cesa-Bianchi and G. Lugosi. Worst-case bounds for the logarithmic loss of predictors. *Machine Learning*, 43(3):247–264, 2001.

N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

D. DeTemple and W. Webb. *Combinatorial reasoning: An introduction to the art of counting*. John Wiley & Sons, 2014.

R. Dykstra, J. Hewett, and T. Robertson. Nonparametric, isotonic discriminant procedures. *Biometrica*, 86(2):429–438, 1999.

T. Fawcett and A. Niculescu-Mizil. PAV and the ROC convex hull. *Machine Learning*, 68(1): 97–106, 2007.

P. Gaillard and S. Gerchinovitz. A chaining algorithm for online nonparametric regression. In *Conference on Learning Theory (COLT)*, pages 764–796, 2015.

S. van de Geer. Estimating a regression function. *Annals of Statistics*, 18:907–924, 1990.

E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2–3):169–192, 2007.

S. M. Kakade, V. Kanade, O. Shamir, and A. Kalai. Efficient learning of generalized linear and single index models with isotonic regression. In *NIPS*, pages 927–935. 2011.

A. T. Kalai and R. Sastry. The isotron algorithm: High-dimensional isotonic regression. In *COLT*, 2009.

J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.

W. Kotłowski and R. Słowiński. Rule learning with monotonicity constraints. In *ICML*, pages 537–544, 2009.

W. Kotłowski and R. Słowinski. On nonparametric ordinal classification with monotonicity constraints. *IEEE Transactions on Knowledge and Data Engineering*, 25(11):2576–2589, 2013.

J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

R. Kyng, A. Rao, and S. Sachdeva. Fast, provable algorithms for isotonic regression in all ℓ_p-norms. In *NIPS*, 2015.

J. de Leeuw, K. Hornik, and P. Mair. Isotone optimization in R: Pool-adjacent-violators algorithm (PAVA) and active set methods. *Journal of Statistical Software*, 32:1–24, 2009.

R. Luss, S. Rosset, and M. Shahar. Efficient regularized isotonic regression with application to gene–gene interaction search. *Annals of Applied Statistics*, 6(1):253–283, 2012.

A. K. Menon, X. Jiang, S. Vembu, C. Elkan, and L. Ohno-Machado. Predicting accurate probabilities with a ranking loss. In *ICML*, 2012.

T. Moon, A. Smola, Y. Chang, and Z. Zheng. Intervalrank: Isotonic regression with listwise and pairwise constraint. In *WSDM*, pages 151–160. ACM, 2010.

H. Narasimhan and S. Agarwal. On the relationship between binary classification, bipartite ranking, and binary class probability estimation. In *NIPS*, pages 2913–2921. 2013.

A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *ICML*, pages 625–632, 2005.

G. Obozinski, C. E. Grant, G. R. G. Lanckriet, M. I. Jordan, and W. W. Noble. Consistent probabilistic outputs for protein function prediction. *Genome Biology*, 2008 2008.

A. Rakhlin and K. Sridharan. Online nonparametric regression. In *COLT*, pages 1232–1264, 2014.

T. Robertson, F. T. Wright, and R. L. Dykstra. *Order Restricted Statistical Inference*. John Wiley & Sons, 1998.

S. de Rooij and T. van Erven. Learning the switching rate by discretising Bernoulli sources online. In *AISTATS*, pages 432–439, 2009.

A. Saumard and J. A. Wellner. Log-concavity and strong log-concavity: A review. *Statistics Surveys*, 8:45–114, 2014.

S. Shalev-Shwartz. Online learning and online convex optimization. In *Foundations and Trends in Machine Learning*, volume 4, pages 107–194. 2012.

M. Stylianou and N. Flournoy. Dose finding using the biased coin up-and-down design and isotonic regression. *Biometrics*, 58(1):171–177, 2002.

A. B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer-Verlag, 2009.

V. Vovk. Aggregating strategies. In *COLT*, pages 371–386, 1990.

V. Vovk, I. Petej, and V. Fedorova. Large-scale probabilistic predictors with and without guarantees of validity. In *NIPS*, pages 892–900. 2015.

Bin Yu. Assouad, fano, and le cam. In *Festschrift for Lucien Le Cam*, pages 423–435. Springer-Verlag, 1997.

B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *KDD*, pages 694–699, 2002.

C.-H. Zhang. Risk bounds in isotonic regression. *The Annals of Statistics*, 30(2):528–555, 2002.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.

Appendix A. The Exponentiated Gradient (EG) bound

We will first cast the online isotonic regression problem to the equivalent problem of minimizing square loss over $(T+1)$-dimensional probability simplex Δ^{T+1}.

Given $\boldsymbol{f} \in \mathcal{F}$, define the $(T+1)$-dimensional vector of increments of $\boldsymbol{f}$ by $\boldsymbol{p}(\boldsymbol{f}) = (f_1 - f_0, f_2 - f_1, \ldots, f_{T+1} - f_T)$, where we used two dummy variables $f_0 = 0$ and $f_{T+1} = 1$. Note that $\boldsymbol{p}(\boldsymbol{f}) \in \Delta^{T+1}$, and there is one-to-one mapping between elements from $\mathcal{F}$ and the corresponding elements from Δ^{T+1}, with the inverse mapping $\boldsymbol{f}(\boldsymbol{p})$ given by $f_t(\boldsymbol{p}) = \sum_{q=1}^{t} p_t$. The loss in the simplex parameterization is given by:

$$\ell_t(\boldsymbol{p}_t) = \Big(y_{i_t} - \sum_{j \le i_t} p_{t,j}\Big)^2 = \Big(y_{i_t} - \boldsymbol{p}_t^\intercal \boldsymbol{x}_{i_t}\Big)^2,$$

where $\boldsymbol{x}_{i_t}$ is the vector with the first i_t coordinates equal to 1. The Exponentiated Gradient (EG) algorithm (Kivinen and Warmuth, 1997) is defined through the update:

$$p_{t,j} = \frac{p_{t-1,j} e^{-\eta \left(\nabla \ell_{t-1}(\boldsymbol{p}_{t-1})\right)_j}}{\sum_{k=1}^{T+1} p_{t-1,k} e^{-\eta \left(\nabla \ell_{t-1}(\boldsymbol{p}_{t-1})\right)_k}},$$

with $\boldsymbol{p}_0$ being some initial distribution. The prediction of the algorithm is then $\widehat{y}_{i_t} = \sum_{j \le i_t} p_{t,j}$. We now use the standard upper bound for the regret of EG:

Theorem 9 (Theorem 5.10 by Kivinen and Warmuth 1997) *Let $\{(\boldsymbol{x}_t, y_t)\}_{t=1}^T$ be a sequence of outcomes such that for all t, $\max_i x_{t,i} - \min_i x_{t,i} \le R$. For any $\boldsymbol{p} \in \Delta^{T+1}$ with $L_T(\boldsymbol{p}) \le K$ and $D(\boldsymbol{p}\|\boldsymbol{p}_0) \le D$ for some $\boldsymbol{p}_0$, the EG algorithm with initial distribution $\boldsymbol{p}_0$ and learning rate η tuned as:*

$$\eta = \frac{2\sqrt{D}}{R(\sqrt{2K} + R\sqrt{D})},$$

have the following bound on its cumulative loss:

$$\widehat{L}_T \le L_T(\boldsymbol{p}) + \sqrt{2KD} + \frac{R^2 D(\boldsymbol{p}\|\boldsymbol{p}_0)}{2}.$$

We apply this theorem to our problem with the sequence permuted by $(i_1, \ldots, i_t)$ and $R = 1$. We choose $\boldsymbol{p}_0$ to be a uniform distribution on Δ^{T+1}, which means $D(\boldsymbol{p}\|\boldsymbol{p}_0) \le \log(T+1) = D$. We also use a crude bound on the loss of best comparator $\boldsymbol{p}^* = \operatorname{argmin}_{\boldsymbol{p}} L_T(\boldsymbol{p})$, $L_T(\boldsymbol{p}^*) \le \frac{1}{4}T = K$ (this is because the loss of the best comparator is lower than the loss of the constant function $\boldsymbol{f}$ equal to the arithmetic mean of the data). This suggests tuning the learning rate to:

$$\eta = \frac{2\sqrt{\log(T+1)}}{\sqrt{\frac{T}{2}} + \sqrt{\log(T+1)}},$$

to get the following regret bound:

$$\mathrm{Reg}_T \le \sqrt{\frac{T\log(T+1)}{2}} + \frac{\log(T+1)}{2}.$$

Appendix B. Proof of Theorem 3 (bound for the Exponential Weights algorithm)

Let the adversary reveal the labels in isotonic order ($i_t = t$ for all t), and they are all equal to 0. At trial t, the prediction of the algorithm $\widehat{y}_t = \boldsymbol{f}_t^\intercal \boldsymbol{x}_t = f_{t,t}$ is given by:

$$\widehat{y}_t = \int_{\mathcal{F}} f_t p_t(\boldsymbol{f}) \,\mathrm{d}f_1 \dots \mathrm{d}f_T, \qquad \text{where} \quad p_t(\boldsymbol{f}) = \frac{e^{-\frac{1}{2}\sum_{q<t} f_q^2}}{\underbrace{\int_{\mathcal{F}} e^{-\frac{1}{2}\sum_{q<t} f_q^2} \,\mathrm{d}f_1 \dots \mathrm{d}f_T}_{=Z}},$$

We calculate the marginal distribution $p_t(f_t = z)$:

$$\begin{aligned} p_t(f_t = z) &= \int_{0 \le f_1 \le f_{t-1} \le z \le f_{t+1} \dots \le f_T \le 1} p_t(\boldsymbol{f}) \,\mathrm{d}f_1 \dots \mathrm{d}f_{t-1} \,\mathrm{d}f_{t+1} \dots \mathrm{d}f_T \\ &= \frac{1}{Z} \left(\int_{0 \le f_1 \le \dots \le f_{t-1} \le z} e^{-\frac{1}{2}\sum_{q<t} f_q^2} \,\mathrm{d}f_1 \dots \mathrm{d}f_{t-1} \right) \left(\int_{z \le f_{t+1} \le \dots \le f_T \le 1} \mathrm{d}f_{t+1} \dots \mathrm{d}f_T \right) \\ &= \frac{1}{Z} G(z, t-1) \frac{(1-z)^{T-t}}{(T-t)!}, \end{aligned}$$

where:

$$G(z, n) = \int_{0 \le f_1 \le \dots \le f_n \le z} e^{-\frac{1}{2}\sum_{t=1}^{n} f_t^2} \,\mathrm{d}f_1 \dots \mathrm{d}f_n.$$

We now calculate $G(z, n)$. Let $F(x) = \int e^{-\frac{1}{2}x^2} \,\mathrm{d}x$ denote the antiderivative of the Gaussian. Recursively applying the relation:

$$\int_{f_{t-1}}^{z} e^{-\frac{1}{2}f_t^2} \frac{1}{k!} (F(z) - F(f_t))^k = \frac{(F(z) - F(f_{t-1}))^{k+1}}{(k+1)!}.$$

we get:

$$G(z, n) = \frac{(F(z) - F(0))^n}{n!},$$

so that:

$$p_t(f_t = z) = \frac{1}{Z'} (1-z)^{T-t} (F(z) - F(0))^{t-1}, \qquad \text{where} \quad Z' = Z(t-1)!(T-t)!\,.$$

Denote $p_t(f_t = z)$ concisely as $\phi(z)$. Then, we have:

$$\widehat{y}_t = \int_0^1 z\phi(z) \,\mathrm{d}z.$$

Assume $t > 1$ and let $\alpha = \frac{t-1}{T-1}$; note that $0 < \alpha \le 1$. Define:

$$g(z) = (1-\alpha)\log(1-z) + \alpha \log(F(z) - F(0)).$$

Note that $\phi(z) = \frac{1}{Z'} e^{(T-1)g(z)}$. We have:

$$g'(z) = -\frac{1-\alpha}{1-z} + \frac{\alpha}{F(z) - F(0)} e^{-\frac{1}{2}z^2},$$

and:

$$g''(z) = -\frac{1-\alpha}{(1-z)^2} - \frac{\alpha}{(F(z)-F(0))^2}e^{-z^2} - \frac{\alpha}{F(z)-F(0)}ze^{-\frac{1}{2}z^2} < 0.$$

Thus, g is (strictly) concave, which implies that ϕ is log-concave. Furthermore, due to strict concavity of $F(z)$ for $z > 0$, we have: $F(0) < F(z) - ze^{-\frac{1}{2}z^2}$ for $z > 0$, which implies:

$$g'(z) < -\frac{1-\alpha}{1-z} + \frac{\alpha}{z} \qquad \text{for } z > 0,$$

so that $g'(\alpha) < 0$. On the other hand, also from the concavity of $F(z)$, $F(z) \le F(0) + z$, which together with $e^{-\frac{1}{2}z^2} \ge 1 - \frac{1}{2}z^2$ implies:

$$g'(z) \ge -\frac{1-\alpha}{1-z} + \frac{\alpha\left(1-\frac{1}{2}z^2\right)}{z}.$$

This means that:

$$g'\left(\frac{\alpha}{2}\right) \ge -\frac{1-\alpha}{1-\frac{\alpha}{2}} + 2\left(1-\frac{1}{8}\alpha^2\right) = \frac{1}{1-\frac{\alpha}{2}} - \frac{1}{4}\alpha^2 \ge 1 > 0.$$

Thus $g'(z)$ switches the sign between $\frac{\alpha}{2}$ and α, which means that the (unique) maximizer $z^* = \operatorname{argmax} g(z)$ is in the range $\left(\frac{\alpha}{2}, \alpha\right)$.

We now use (Saumard and Wellner, 2014, Proposition 5.2) which states that for log-concave densities, the density at the mean is not much smaller than the density at the mode:

$$\frac{1}{\sqrt{3}e}\sup_z \phi(z) \le \phi(\widehat{y}_t) \le \sup_z \phi(z),$$

which means that after taking logarithms, dividing by $T-1$ and using the definition of z^*,

$$g(z^*) - \frac{1}{T-1}(1+\log\sqrt{3}) \le g(\widehat{y}_t) \le g(z^*).$$

From concavity of g,

$$g(\widehat{y}_t) \le g\left(\frac{\alpha}{2}\right) + g'\left(\frac{\alpha}{2}\right)\left(\widehat{y}_t - \frac{\alpha}{2}\right) \le g(z^*) + g'\left(\frac{\alpha}{2}\right)\left(\widehat{y}_t - \frac{\alpha}{2}\right),$$

which, together with $g'\left(\frac{\alpha}{2}\right) \ge 1$, implies:

$$\widehat{y}_t \ge \frac{\alpha}{2} + \frac{g(\widehat{y}_t) - g(z^*)}{g'\left(\frac{\alpha}{2}\right)} \ge \frac{\alpha}{2} - \frac{\frac{1}{T-1}(1+\log\sqrt{3})}{g'\left(\frac{\alpha}{2}\right)} \ge \frac{\alpha}{2} - \frac{1}{T-1}(1+\log\sqrt{3}).$$

Hence, $\widehat{y}_t \ge \frac{\alpha}{4}$ when:

$$\frac{\alpha}{2} - \frac{1}{T-1}(1+\log\sqrt{3}) \ge \frac{\alpha}{4} \qquad \Longrightarrow \qquad T \ge 1 + \frac{4(1+\log\sqrt{3})}{\alpha}.$$

Note, that this holds when $\alpha \ge \frac{1}{2}$ and when $T \ge 14$. Therefore, when $T \ge 14$, for all $\alpha \ge \frac{1}{2}$, we have $\widehat{y}_t \ge \frac{\alpha}{4} \ge \frac{1}{8}$, which means $\ell_t(\boldsymbol{f}_t) = (\widehat{y}_t - 0)^2 \ge \frac{1}{64}$. Since $\alpha \ge \frac{1}{2}$ is implied by $t \ge \lfloor T/2 \rfloor + 1$, we conclude that when $T \ge 14$,

$$\mathrm{Reg}_T = \widehat{L}_T - \min_{f\in\mathcal{F}} L_T(\boldsymbol{f}) = \widehat{L}_T \ge \sum_{t=\lfloor \frac{T}{2} \rfloor + 1}^{T} \frac{1}{64} \ge \frac{T}{128}.$$

Appendix C. Full proof of Theorem 5

We proceed by constructing the difficult sequence explicitly. First, split the T points $(1,\dots,T)$ into K consecutive segments $(1,\dots,m),(m+1,\dots,2m),\dots,(m(K-1)+1,\dots,T)$, where in each segment there are $m=\frac{T}{K}$ consecutive points (assume for now that T is divisible by K). Let $t\in k$ mean that t is in the k-th segment, $k=1,\dots,K$, i.e. $t\in k$ whenever $k=\lceil\frac{t}{m}\rceil$. Now, suppose the adversary picks a K-vector $\boldsymbol{p}=(p_1,\dots,p_K)\in[0,1]^K$ such that $p_1\le p_2\le\ldots\le p_K$ and generates the labels in isotonic order (any order would work as well) such that $y_t\sim\text{Bernoulli}(p_k)$ when $t\in k$. Let $\boldsymbol{f_p}\in\mathcal{F}$ denote an isotonic function such that $f_t=p_k$ when $t\in k$. We lower bound the expected regret:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{p}}[\text{Reg}_T] &= \mathbb{E}_{\boldsymbol{p}}\left[\widehat{L}_T-\inf_{\boldsymbol{f}\in\mathcal{F}}L_T(\boldsymbol{f})\right]\\
&\ge \mathbb{E}_{\boldsymbol{p}}\left[\widehat{L}_T-L_T(\boldsymbol{f_p})\right]\\
&= \sum_{k=1}^{K}\mathbb{E}_{\boldsymbol{p}}\left[\sum_{t\in k}(\widehat{y}_t-y_t)^2-(p_k-y_t)^2\right]\\
&= \sum_{k=1}^{K}\mathbb{E}_{\boldsymbol{p}}\left[\sum_{t\in k}(\widehat{y}_t-p_k)(\widehat{y}_t+p_k-2y_t)^2\right]\\
&= \sum_{k=1}^{K}\mathbb{E}_{\boldsymbol{p}}\left[\sum_{t\in k}(\widehat{y}_t-p_k)^2\right],
\end{aligned}$$

where the last equality is from $\mathbb{E}_{p_k}[y_t]=p_k$. Now we assume the adversary picks $\boldsymbol{p}$ from the following set:

$$\mathcal{P}=\left\{\boldsymbol{p}=(p_1,\dots,p_K)\colon p_k\in\left\{\frac{1}{4}+\frac{k-1}{2K},\frac{1}{4}+\frac{k}{2K}\right\}\right\}.$$

There are 2^K vectors in $\mathcal{P}$, all satisfying $p_1\le p_2\le\ldots\le p_K$, and $\frac{1}{4}\le p_k\le\frac{3}{4}$ for all k. For instance, when $K=2$, $\mathcal{P}=\left\{(\frac{1}{4},\frac{1}{2}),(\frac{1}{4},\frac{3}{4}),(\frac{1}{2},\frac{1}{2}),(\frac{1}{2},\frac{3}{4})\right\}$.

Fix k and denote $p_{k,0}=\frac{1}{4}+\frac{k-1}{2K}$ and $p_{k,1}=\frac{1}{4}+\frac{k}{2K}$, i.e. $p_k\in\{p_{k,0},p_{k,1}\}$. Define:

$$\widehat{p}_k=\operatorname*{argmin}_{p_{k,i},i=0,1}\left\{|\overline{y}_k-p_{k,i}|\right\},$$

where $\overline{y}_k=\frac{1}{m}\sum_{t\in k}\widehat{y}_t$. We now show that:

$$\sum_{t\in k}(\widehat{y}_t-p_k)^2\ge\frac{m}{4}(\widehat{p}_k-p_k)^2. \tag{3}$$

Without loss of generality, assume $p_k=p_{k,0}$. Then, if $\widehat{p}_k=p_{k,0}$, the inequality clearly holds because the right-hand side is 0. On the other hand, if $\widehat{p}_k=p_{k,1}$, then from the definition of $\widehat{p}_k$ we

have $|\overline{y}_k - p_{k,1}| \le |\overline{y}_k - p_{k,0}|$, which means $\overline{y}_k \ge \frac{1}{2}(p_{k,0} + p_{k,1})$. This implies:

$$\begin{aligned}\sum_{t\in k}(\widehat{y}_t - p_{k,0})^2 &= \sum_{t\in k}(\widehat{y}_t - \overline{y}_k + \overline{y}_k - p_{k,0})^2 \\ &= \left(\sum_{t\in k}(\widehat{y}_t - \overline{y}_k)^2 + 2(\widehat{y}_t - \overline{y}_k)(\overline{y}_k - p_{k,0})\right) + m(\overline{y}_k - p_{k,0})^2 \\ &= \left(\sum_{t\in k}(\widehat{y}_t - \overline{y}_k)^2\right) + m(\overline{y}_k - p_{k,0})^2 \\ &\ge m(\overline{y}_k - p_{k,0})^2 \ge \frac{m}{4}(p_{k,1} - p_{k,0})^2.\end{aligned}$$

Thus, (3) holds. Note that $\widehat{p}_k$ depends on $\{\widehat{y}_t, t \in k\}$ (which in turn depend on the labels), but it does not depend on p_k. Hence, the worst-case regret can be lower bounded by:

$$\max_{y_1,\dots,y_T} \mathrm{Reg}_T \ge \max_{\boldsymbol{p}\in\mathcal{P}} \sum_{k=1}^{K} \frac{m}{4}\mathbb{E}_{\boldsymbol{p}}[(\widehat{p}_k - p_k)^2].$$

We will now use Assouads lemma (Yu, 1997; Tsybakov, 2009) to bound the sum on the right-hand side. Let $\Omega = \{0,1\}^K$ be the set of all 2^K binary sequences of length K. The sequences from Ω will index the distributions from $\mathcal{P}$ by denoting $\boldsymbol{p}_{\boldsymbol{\omega}} = (p_{1,\omega_1}, \dots, p_{K,\omega_K}) \in \mathcal{P}$ for any $\boldsymbol{\omega} \in \Omega$. We also define $\widehat{\boldsymbol{\omega}} \in \Omega$ as $\boldsymbol{p}_{\widehat{\boldsymbol{\omega}}} = (\widehat{p}_1, \dots, \widehat{p}_K)$, i.e. $p_{k,\widehat{\omega}_k} = \widehat{p}_k$ for any k. In this notation:

$$\max_{\boldsymbol{p}\in\mathcal{P}} \sum_{k=1}^{K} \frac{m}{4}\mathbb{E}_{\boldsymbol{p}}[(\widehat{p}_k - p_k)^2] \ge \max_{\boldsymbol{\omega}\in\Omega} \frac{m}{4}\mathbb{E}_{\boldsymbol{p}_{\boldsymbol{\omega}}}\|\boldsymbol{p}_{\boldsymbol{\omega}} - \boldsymbol{p}_{\widehat{\boldsymbol{\omega}}}\|^2 = \max_{\boldsymbol{\omega}\in\Omega} \frac{m}{4}\frac{1}{4K^2}\mathbb{E}_{\boldsymbol{p}_{\boldsymbol{\omega}}}\rho(\boldsymbol{\omega}, \widehat{\boldsymbol{\omega}}),$$

where $\rho(\cdot,\cdot)$ denotes the Hamming distance between two binary sequences. Using (Tsybakov, 2009, Theorem 2.12.ii):

$$\min_{\widehat{\boldsymbol{\omega}}\in\Omega} \max_{\boldsymbol{\omega}\in\Omega} \mathbb{E}_{\boldsymbol{p}_{\boldsymbol{\omega}}}\rho(\boldsymbol{\omega}, \widehat{\boldsymbol{\omega}}) \ge \frac{K}{2}\left(1 - \max_{\boldsymbol{\omega},\boldsymbol{\omega}'\in\Omega:\ \rho(\boldsymbol{\omega},\boldsymbol{\omega}')=1} \mathrm{TV}(\boldsymbol{p}_{\boldsymbol{\omega}}, \boldsymbol{p}_{\boldsymbol{\omega}'})\right),$$

where $\mathrm{TV}(\cdot,\cdot)$ is the total variation distance between distributions over $(y_1, \dots, y_T)$. From Pinsker's inequality:

$$\mathrm{TV}^2(\boldsymbol{p}_{\boldsymbol{\omega}}, \boldsymbol{p}_{\boldsymbol{\omega}'}) \le \frac{1}{2}D(\boldsymbol{p}_{\boldsymbol{\omega}}\|\boldsymbol{p}_{\boldsymbol{\omega}'}) = \frac{1}{2}\sum_{k=1}^{K} D(p_{k,\omega_k}\|p_{k,\omega'_k}),$$

where $D(\cdot\|\cdot)$ is the Kullback-Leibler divergence and we used the fact the $\boldsymbol{p} \in \mathcal{P}$ are product distributions over segments. Since $\rho(\boldsymbol{\omega}, \boldsymbol{\omega}') = 1$, $p_{k,\omega_k} = p_{k,\omega'_k}$ for all but one k, hence all but one terms in the sum disappear and we have $\mathrm{TV}^2(\boldsymbol{p}_{\boldsymbol{\omega}}, \boldsymbol{p}_{\boldsymbol{\omega}'}) \le \frac{1}{2}D(p_{k,0}\|p_{k,1})$ for some k. Using the Taylor approximation of the KL-divergence with respect to $p_{k,1}$ around $p_{k,0}$, we get:

$$D(p_{k,0}\|p_{k,1}) = \frac{m}{2}\frac{(p_{k,1} - p_{k,0})^2}{\widetilde{p}(1-\widetilde{p})},$$

where $\widetilde{p}$ is some convex combination of $p_{k,0}$ and $p_{k,1}$. Since $p_{k,0}, p_{k,1} \in [\frac{1}{4}, \frac{3}{4}]$, $\frac{1}{\widetilde{p}(1-\widetilde{p})}$ is maximized for $\widetilde{p} \in \{\frac{1}{4}, \frac{3}{4}\}$, so that:

$$\mathrm{TV}^2(p_{k,0}, p_{k,1}) \le \frac{4m}{3}(p_{k,1} - p_{k,0})^2 = \frac{m}{3K^2}.$$

Plugging this into our bound gives:

$$\min_{\widehat{\boldsymbol{\omega}}\in\Omega}\max_{\boldsymbol{\omega}\in\Omega}\mathbb{E}_{\boldsymbol{p}_{\boldsymbol{\omega}}}\rho(\boldsymbol{\omega},\widehat{\boldsymbol{\omega}}) \geq \frac{K}{2}\left(1-\frac{\sqrt{m}}{\sqrt{3}K}\right),$$

which implies:

$$\max_{y_1,\ldots,y_T}\mathrm{Reg}_T \geq \frac{m}{32K}\left(1-\frac{\sqrt{m}}{\sqrt{3}K}\right) = \frac{T}{32K^2}\left(1-\frac{\sqrt{T}}{\sqrt{3}K^{3/2}}\right),$$

where we used the fact that $m=\frac{T}{K}$. Choosing $K=cT^{1/3}$ for some $c>1$ gives:

$$\max_{y_1,\ldots,y_T}\mathrm{Reg}_T \geq \frac{c^{3/2}-3^{-1/2}}{32c^{7/2}}T^{1/3}.$$

Choosing any $c>3^{-1/3}$, $c=O(1)$, such that K divides T finishes the proof.

Appendix D. Proofs for Section 6 (noise-free case)

D.1. Proof of Theorem 6 (arbitrary order of outcomes)

We first give a sequence of outcomes such that when $T=2^k-1$ for some positive integer k, any algorithm will suffer exactly $\frac{1}{4}\log_2(T+1)$ loss. The adversary picks a point in the middle of the range, $i_1=2^{k-1}$. After the algorithm predicts, the adversary chooses y_{i_1} to be 0 or 1, depending which of these two incurs more loss to the algorithm. Hence, no matter what the algorithm predicts, the loss is at least $\frac{1}{4}$. If $y_{i_1}=0$, then $2^{k-1}-1$ points on the left-hand side of y_{i_1} are labeled to 0 in the next trials (which is required due to noise-free regime), and the algorithm will possibly suffer no loss on these points. Then, the adversary repeats the above procedure of choosing the middle point on the remaining $2^{k-1}-1$ points on the right-hand side of y_{i_1}. Analogously, when $y_{i_1}=1$, the points on the right-hand side are all labeled to 1, and the adversary recursively play on remaining the left-hand side points. This procedure can be carried out k times, until no more points remains. Hence, the total loss incurred by the algorithm is at least $\frac{1}{4}k=\frac{1}{4}\log_2(n+1)$.

Next, we determine the upper bound on the value of the minimax regret:

$$V=\min_{\widehat{y}_{i_1}}\max_{y_{i_1}}\ldots\min_{y_{i_T}}\max_{y_{i_T}}\sum_{t=1}^{T}(y_{i_t}-\widehat{y}_{i_t})^2 \leq \frac{1}{4}\log_2(T+1),$$

where the labels are constrained to be isotonic, $y_1\leq\ldots\leq y_T$. We will get the predictions of the minimax algorithm as a by-product of the calculations. This implies that the minimax algorithm suffers regret at most $\frac{1}{4}\log_2(T+1)$, and the bound on V is tight whenever $T=2^k-1$.

In the first trial, the adversary reveals outcome i_1, which splits the set of unknown labels into two disjoint sets $(y_1,\ldots,y_{i_t-1})$ and $(y_{i_t+1},\ldots,y_T)$. The minimax algorithm knows that $0\leq y_1,\ldots,y_{i_t-1}\leq y_{i_t}$ and $y_{i_t}\leq y_{i_t+1},\ldots,y_T\leq 1$ (due to noise-free case). Then, in the future trials, each of these sets of unknown consecutive labels will be recursively split into smaller sets. At any moment of time, for any set of unknown labels $(y_i,\ldots,y_j)$ with $j\geq i$, we know that $y_{i-1}\leq y_i,\ldots,y_j\leq y_{j+1}$, and y_{i-1} and y_{j+1} has already been revealed (we use $y_0=0$ and

$y_{T+1} = 1$). Hence, the minimax algorithm will play a separate minimax game for each set of unknown labels $(y_i, \dots, y_j)$, bounded in the range $[y_{i-1}, y_{j+1}]$. We use this observation as a basis for the recursion. Let $V(u, v, n)$ denote the minimax value of the game for a set of n not yet revealed consecutive labels lower-bounded by u, and upper-bounded by v. We get the recursion:

$$V(u, v, n+1) = \max_{k \in \{0,\dots,n\}} V(u, v, n+1, k), \tag{4}$$

where:

$$V(u, v, n+1, k) = \min_{\widehat{y} \in [u,v]} \max_{y \in [u,v]} \left\{ (y - \widehat{y})^2 + V(u, y, k) + V(y, v, n-k) \right\},$$

which follows from the fact that first the adversary reveals $(k+1)$-th point, then the algorithm predicts with $\widehat{y}$ for that point, and finally the outcome y is revealed, while the set is split into two sets of smaller size. The minimax regret can be read out from $V(0, 1, T)$. To start the recursion, we define $V(u, v, 0) = 0$.

We now prove by induction on n that:

$$V(u, v, n) = \beta_n (v - u)^2, \tag{5}$$

where β_n is some coefficient independent of u and v. Assume $n+1$ unknown labels, lower-bounded by u, and upper-bounded by v. We fix $k \in \{0, \dots, n\}$ and calculate the optimal prediction of the algorithm for $V(u, v, n+1, k)$:

$$\widehat{y} = \operatorname*{argmin}_{\widehat{y} \in [u,v]} \max_{y \in [u,v]} \left\{ (y - \widehat{y})^2 + \beta_k (y - u)^2 + \beta_{n-k} (v - y)^2 \right\},$$

where we used the inductive assumption. The function inside $\max$ is convex in y, hence the solution w.r.t. y is $y \in \{u, v\}$. First note that if $\beta_k - \beta_{n-k} > 1$, the function inside $\max$ is increasing in y for any choice of $\widehat{y} \in [u, v]$, hence the optimal choice for the adversary is $y = v$, and the optimal choice for the algorithm is $\widehat{y} = v$. Similarly, if $\beta_k - \beta_{n-k} < -1$, the function inside $\max$ is decreasing in y, which results in the optimal choice $y = u$ and $\widehat{y} = u$. When $-1 \le \beta_k - \beta_{n-k} \le 1$, it is easy to check that the optimal prediction is obtained by setting the function inside $\max$ equal for both choices of $y \in \{u, v\}$. This gives:

$$\widehat{y} = \frac{u+v}{2} + \frac{u-v}{2} (\beta_k - \beta_{n-k}) \in [u, v].$$

Thus, depending on the value of $\beta_k - \beta_{n-k}$, $V(u, v, n+1, k)$ is given by:

$$V(u, v, n+1, k) = (u - v)^2 \beta_{n,k},$$

where

$$\beta_{n,k} = \begin{cases} \beta_k & \text{if } \beta_k - \beta_{n-k} > 1, \\ \frac{1}{4}(\beta_k - \beta_{n-k})^2 + \frac{1}{2}(\beta_k + \beta_{n-k}) + \frac{1}{4} & \text{if } -1 \le \beta_k - \beta_{n-k} \le 1, \\ \beta_{n-k} & \text{if } \beta_k - \beta_{n-k} < -1. \end{cases} \tag{6}$$

From (4), we have:

$$\beta_{n+1} = \max_{k \in \{0,\dots,n\}} \beta_{n,k},$$

which proves our inductive hypothesis (5).

What is left to show is that $\beta_n \leq \frac{1}{4}\log_2(n+1)$. We will prove it by induction on n. For $n = 0$, $\beta_0 = 0$ and thus the bound trivially holds. We now show that $\beta_{n,k}$, as defined in (6), is nondecreasing in β_k and β_{n-k}. We fix β_{n-k}, and calculate the derivative with respect to β_k:

$$\frac{\mathrm{d}\beta_{n,k}}{\mathrm{d}\beta_k} = \begin{cases} 1 & \text{if } \beta_k - \beta_{n-k} > 1, \\ \frac{1}{2}(\beta_k - \beta_{n-k} + 1) & \text{if } -1 \leq \beta_k - \beta_{n-k} \leq 1, \\ 0 & \text{if } \beta_k - \beta_{n-k} < -1, \end{cases}$$

which is nonnegative. Hence, $\beta_{n,k}$ is nondecreasing with β_k for any fixed β_{n-k}. An analogous arguments shows that $\beta_{n,k}$ is nondecreasing with β_{n-k} for any fixed β_k. Hence, we can replace β_k and β_{n-k} by their upper bounds from the inductive argument, and then $\beta_{n,k}$ (as well as β_{n+1}) will not decrease. Thus, to show that $\frac{1}{4}\log_2((n+1)+1)$ is the upper bound on β_{n+1}, it suffices to show that for any n, k, $\beta_{n,k} \leq \frac{1}{4}\log_2(n+2)$ after substituting $\beta_k = \frac{1}{4}\log_2(k+1)$ and $\beta_{n-k} = \frac{1}{4}\log_2(n-k+1)$ in (6).

We proceed by cases in (6). When $\beta_k - \beta_{n-k} > 1$, $\beta_{n,k} = \beta_k = \frac{1}{4}\log_2(k+1) \leq \frac{1}{4}\log_2(n+2)$, because $k \leq n$. Case $\beta_k - \beta_{n-k} < -1$ is covered in an analogous way. We are left with the case $-1 \leq \beta_k - \beta_{n-k} \leq 1$. It suffices to show that:

$$\underbrace{(\beta_k - \beta_{n-k})^2 + \frac{1}{2}(\beta_k + \beta_{n-k}) + \frac{1}{4}}_{=g(\beta_k, \beta_{n-k})} \leq \underbrace{\frac{1}{4}\log_2(2^{4\beta_k} + 2^{4\beta_{n-k}})}_{=f(\beta_k, \beta_{n-k})}, \tag{7}$$

because the right-hand side is equal to $\frac{1}{4}\log_2(n+2)$ when $\beta_k = \frac{1}{4}\log_2(1+k)$ and $\beta_{n-k} = \frac{1}{4}\log_2(1+n-k)$. Assume w.l.o.g. that $\beta_k \leq \beta_{n-k}$ (because both $f(\cdot,\cdot)$ and $g(\cdot,\cdot)$ are symmetric in their arguments). For any δ, $g(x+\delta, y+\delta) = g(x,y)+\delta$, and similarly $f(x+\delta, y+\delta) = f(x,y)+\delta$. Thus, proving $f(x,y) \geq g(x,y)$ is equivalent to proving $f(0, y-x) \geq g(0, y-x)$. Given the condition $-1 \leq \beta_{n-k} - \beta_k \leq 1$, we thus need to show that: $f(0,y) \geq g(0,y)$ for any $0 \leq y \leq 1$, which translates to:

$$\log_2\left(1 + 2^{4y}\right) \geq (1+y)^2, \qquad \text{for } 0 \leq y \leq 1.$$

This inequality can be shown by splitting the range $[0,1]$ into $[0, \frac{1}{4}]$, $[\frac{1}{4}, \frac{3}{4}]$ and $[\frac{3}{4}, 1]$, lower-bounding the left-hand side by its Taylor expansion up to the second order around points 0, $\frac{1}{2}$, and $\frac{3}{4}$, respectively (with the second derivative replaced by its lower bound in a given range), and showing that the corresponding quadratic inequality always holds within its range. We omit the details here. Unfortunately, we were unable to find a more elegant proof of this inequality.

D.2. Proof of Theorem 7 (isotonic order of outcomes)

We determine the value of the minimax regret:

$$V = \min_{\widehat{y}_1} \max_{y_1 \in [0,1]} \min_{\widehat{y}_1} \max_{y_2 \in [y_1, 1]} \dots \min_{\widehat{y}_T} \max_{y_T \in [y_{T-1}, 1]} \sum_{t=1}^{T} (y_t - \widehat{y}_t)^2,$$

getting the predictions of the minimax algorithm as a by-product of the calculations. Note that any algorithm will suffer regret at least V for some sequences of labels, while the minimax algorithm

will suffer regret at most V for any sequence of labels. Let:

$$V_{T-t}(y_t) = \min_{\widehat{y}_{t+1}} \max_{y_{t+1}\in[y_t,1]} \dots \min_{\widehat{y}_T} \max_{y_T\in[y_{T-1},1]} \sum_{q=t+1}^{T} (y_q - \widehat{y}_q)^2$$

be the value-to-go function, which is the worst-case loss suffered by the minimax algorithm in $T-t$ trials $t+1,\dots,T$, given the last revealed label was y_t. The minimax regret V can be read out from $V_T(0)$. We use the following recursion:

$$V_n(c) = \min_{\widehat{y}} \max_{y\in[c,1]} \left\{ (y-\widehat{y})^2 + V_{n-1}(y) \right\},$$

where we used $V_0(y) = 0$. We start with calculating $V_1(c)$ (which corresponds to the last trial $t = T$). The minimax prediction is given by:

$$\operatorname*{argmin}_{\widehat{y}} \max_{y\in[c,1]} (\widehat{y}-y)^2 = \operatorname*{argmin}_{\widehat{y}} \max\{(\widehat{y}-c)^2, (\widehat{y}-1)^2\} = \frac{c+1}{2},$$

and the value-to-go function is $V_1(c) = \frac{1}{4}(1-c)^2$. We now prove by induction that $V_n(c) = \alpha_n(1-c)^2$ for some $\alpha_n > 0$ (which clearly holds for $n = 1$ with $\alpha_1 = \frac{1}{4}$, as shown above). By the induction argument,

$$V_n(c) = \min_{\widehat{y}} \max_{y\in[c,1]} \left\{ (\widehat{y}-y)^2 + \alpha_{n-1}(1-y)^2 \right\} = \min_{\widehat{y}} \max_{y\in\{c,1\}} \left\{ (\widehat{y}-y)^2 + \alpha_{n-1}(1-y)^2 \right\}.$$

The last equality is due to the fact that the function inside the $\min\max$ is convex in y, therefore the optimal y is on the boundary of the feasible range $[c,1]$. It is easy to check that the optimal $\widehat{y}$ makes the expression inside $\min\max$ equal for both choices of y, so that:

$$\widehat{y}_t = \frac{c+1}{2} + \alpha_{n-1}\frac{c-1}{2},$$

The expression inside max is a convex function of y_t, therefore the optimal y_t is on the boundary of feasible range $\{y_{t-1}, b\}$. The algorithm predicts with $\widehat{y}_t$, such that the expression inside max has the same value for $y_t = y_{t-1}$ and $y_t = b$. This gives:

$$\widehat{y}_t = \frac{b+y_{t-1}}{2} - \alpha_{t+1}^T \frac{b-y_{t-1}}{2},$$

and:

$$V_n(c) = (\widehat{y}-1)^2 = \underbrace{\left(\frac{\alpha_{n-1}+1}{2}\right)^2}_{=\alpha_n}(1-c)^2.$$

This finishes the inductive proof for $V_n(c)$. The value of the minimax regret is given by $V_T(0) = \alpha_T$. Now, given that $\alpha_1 = \frac{1}{4} < 1$, we have inductively for all n:

$$\alpha_n = \left(\frac{\alpha_{n-1}+1}{2}\right)^2 \le \left(\frac{1+1}{2}\right)^2 = 1,$$