Random Permutation Online Isotonic Regression

Authored by:

Wouter M. Koolen
Wojciech Kotlowski
Alan Malek

Abstract

We revisit isotonic regression on linear orders, the problem of fitting monotonic functions to best explain the data, in an online setting. It was previously shown that online isotonic regression is unlearnable in a fully adversarial model, which lead to its study in the fixed design model. Here, we instead develop the more practical random permutation model. We show that the regret is bounded above by the excess leave-one-out loss for which we develop efficient algorithms and matching lower bounds. We also analyze the class of simple and popular forward algorithms and recommend where to look for algorithms for online isotonic regression on partial orders.

1 Paper Body

A function f : R ? R is called isotonic (non-decreasing) if x ? y implies f (x) ? f (y). Isotonic functions model monotonic relationships between input and output variables, like those between drug dose and response [25] or lymph node condition and survival time [24]. The problem of isotonic regression is to find the isotonic function that best explains a given data set or population distribution. The isotonic regression problem has been extensively studied in statistics [1, 24], which resulted in efficient optimization algorithms for fitting isotonic functions to the data [7, 16] and sharp convergence rates of estimation under various model assumptions [26, 29]. In online learning problems, the data arrive sequentially, and the learner is tasked with predicting each subsequent data point as it arrives [6]. In online isotonic regression, the natural goal is to predict the incoming data points as well as the best isotonic function in hindsight. Specifically, for time steps t = 1, . . . , T , the learner observes an instance xi ? R, makes a prediction ybi of the true label yi , which is assumed to lie in [0, 1]. There is no restriction that the labels or predictions be isotonic. We evaluate a prediction ybi by its squared loss (b yi ? yi)2 . The quality of an algorithm is measured by its PT yi ? yi)2 ? L?T , where L?T is the loss of the best isotonic function on the entire data regret, t=1 (b sequence. Isotonic regression is nonparametric:

the number of parameters grows linearly with the number of data points. It is thus natural to ask whether there are efficient, provably low regret algorithms for online isotonic regression. As of yet, the picture is still very incomplete in the online setting. The first online results were obtained in the recent paper [14] which considered linearly ordered domains in the adversarial fixed design model, i.e. a model in which all the inputs x1 , . . . , xT are given to the learner before the start of prediction. The authors show that, due to the nonparametric nature of the problem, many textbook online learning algorithms fail to learn at all (including Online Gradient Descent, Follow the Leader and Exponential Weights) in the sense that their worst-case regret grows 1 linearly with the number of data points. They prove a ?(T 3) worst case regret lower bound, and 1 ? 3) regret. Unfortunately, the fixed design develop a matching algorithm that achieves the optimal O(T assumption is often unrealistic. This leads us to our main question: Can we design methods for online isotonic regression that are practical (do not hinge on fixed design)? 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Our contributions Our long-term goal is to design practical and efficient methods for online isotonic regression, and in this work we move beyond the fixed design model and study algorithms that do not depend on future instances. Unfortunately, the completely adversarial design model (in which the instances are selected by an adaptive adversary) is impossibly hard: every learner can suffer linear regret in this model [14]. So in order to drop the fixed design assumption, we need to constrain the adversary in some other way. In this paper we consider the natural random permutation model, in which all T instances and labels are chosen adversarially before the game begins but then are presented to the learner in a random order. This model corresponds with the intuition that the data gathering process (which fixes the order) is independent of the underlying data generation mechanism (which fixes instances and labels). We will show that learning is possible in the random permutation model (in fact we present a reduction ? 13) upper bound on showing that it is not harder than adversarial fixed design) by proving an O(T regret for an online-to-batch conversion of the optimal fixed design algorithm from [14] (Section 3). Our main tool for analyzing the random permutation model is the leave-one-out loss, drawing interesting connections with cross-validation and calibration. The leave-one-out loss on a set of t labeled instances is the error of the learner predicting the i-th label after seeing all remaining t ? 1 labels, averaged uniformly over i = 1, . . . , t. We begin by proving a general correspondence between regret and leave-one-out loss for the random permutation model in Section 2.1, which allows us to use excess leave-one-out loss as a proxy for regret. We then describe a version of online-to-batch conversion that relates the fixed design model with the random permutation model, resulting in an ? 31) regret. algorithm that attains the optimal O(T Section 4 then turns to the computationally efficient and natural class of forward algorithms that use an offline optimization oracle to form their prediction. This class contains most common online 1 isotonic regression algorithms. We then show a O(T 2) upper bound on the regret for the entire class, 1 which improves to O(T 3) for the well-specified case where the data are in fact generated from an isotonic function plus i.i.d. noise (the most common model in the statistics literature). 1

While forward algorithms match the lower bound for the well-specified case, there is a factor T 6 gap in the random permutation case. Section 4.6 proposes a new algorithm that calls a weighted offline oracle with a large weight on the current instance. This algorithm can be efficiently computed via [16]. We prove necessary bounds on the weight. Related work Offline isotonic regression has been extensively studied in statistics starting from work by [1, 4]. Applications range across statistics, biology, medicine, psychology, etc. [24, 15, 25, 22, 17]. In statistics, isotonic regression is studied in generative models [26, 3, 29]. In machine learning, isotonic regression is used for calibrating class probability estimates [28, 21, 18, 20, 27], ROC analysis [8], training Generalized Linear Models and Single Index Models[12, 11], data cleaning [13], and ranking [19]. Fast algorithms for partial ordering are developed in [16]. In the online setting, [5] bound the minimax regret for monotone predictors under logarithmic loss and [23, 10] study online nonparametric regression in general. Efficient algorithms and worst-cases regret bounds for fixed design online isotonic regression are studied in [14]. Finally, the relation between regret and leave-one-out loss was pioneered by [9] for linear regression.

2

Problem Setup

Given a finite set of instances {x1 , . . . , xt } ? R, a function f : {x1 , . . . , xt } ? [0, 1] is isotonic (non-decreasing) if xi ? xj implies f (xi) ? f (xj) for all i, j ? {1, . . . , t}. Given a set of labeled instances D = {(x1 , y1), . . . , (xt , yt)} ? R ? [0, 1], let L? (D) denote the total squared loss of the best isotonic function on D, t X L? (D) := min (yi ? f (xi))2 . isotonic f

i=1

This convex optimization problem can be solved by the celebrated Pool Adjacent Violators Algorithm (PAVA) in time linear in t [1, 7]. The optimal solution, called the isotonic regression function, is piecewise constant and its value on any of its levels sets equals the average of labels within that set [24]. 2

Online isotonic regression in the random permutation model is defined as follows. At the beginning of the game, the adversary chooses data instances x1 ¡ . . . ¡ xT 1 and labels y1 , . . . , yT . A permutation ? = (?1 , . . . , ?T) of {1, . . . , T } is then drawn uniformly at random and used to determine the order in which the data will be revealed. In round t, the instance x?t is revealed to the learner who then predicts yb?t . Next, the learner observes the true label y?t and incurs the squared loss (b y?t ? y?t)2 . ? ? ? For a fixed permutation ?, we use the shorthand notation Lt = L ({(x?1 , y?1), . . . , (x?t , y?t)}) to denote the optimal isotonic regression loss of the first t labeled instances (L?t will clearly depend on ?, except for the case t = T). The goal of the learner is to minimize the expected regret, X

T T X RT := E? (y?t ? yb?t)2 ? L?T = rt , t=1

t=1

where we have decomposed the regret into its per-round increase, h i rt := E? (y?t ? yb?t)2 ? L?t + L?t?1 ,

(1)

L?0

:= 0. To simplify the analysis, let us assume that the prediction strategy does not depend with on the order in which the past data were revealed (which is true for all algorithms considered in this paper). Fix t and define D = {(x?1 , y?1), . . . , (x?t , y?t)} to be the set of first t labeled instances. Furthermore, let D?i = D{(x?i , y?i)} denote the set D with the instance from round i removed. Using this notation, the expression under the expectation in (1) can be written as 2 y?t ? yb?t (D?t) ? L? (D) + L? (D?t), where we made the dependence of yb?t on D?t explicit (and used the fact that it only depends on the set of instances, not on their order). By symmetry of the expectation over permutations with respect to the indices, we have h h 2 i 2 i

E? y?t ? yb?t (D?t) = E? y?i ? yb?i (D?i) , and E? L? (D?t) = E? L? (D?i) , for all i = 1, . . . , t. Thus, (1) can as well be rewritten as:

X t

2 1 y?i ? yb?i (D?i) + L? (D?i) ? L? (D) . rt = E? t i=1 Let us denote the expression inside the expectation by rt (D) to stress its dependence on the set of instances D, but not on the order in which they were revealed. If we can show that rt (D) ? Bt holds PT for all t, then its expectation has the same bound, so RT ? t=1 Bt . 2.1

Excess Leave-One-Out Loss and Regret

Our main tool for analyzing the random permutation model is the leave-one-out loss. In the leave-one-out model, there is no sequential structure. The adversary picks a data set D = {(x1 , y1), . . . , (xt , yt)} with x1 ¡ . . . ¡ xt . An index i is sampled uniformly at random, the learner is given D?i , the entire data set except (xi , yi), and predicts ybi (as a function of D?i) on instance xi . We call the difference between the expected loss of the learner and L? (D) the expected excess leave-one-out loss: ! X

t 2 1 ? ‘oot (D) := yi ? ybi (D?i) ? L (D) . (2) t i=1 The random permutation model has the important property that the bound on the excess leave-one-out loss of a prediction algorithm translates into a regret bound. A similar result has been shown by [9] for expected loss in the i.i.d. setting. Lemma 2.1. rt (D) ? ‘oot (D) for any t and any data set D = {(x1 , y1), . . . , (xt , yt)}. Pt 2 Proof. As x1 ¡ . . . ¡ xt , let (y1? , . . . , yt?) = argminf1 ?...?ft i=1 (yi ? fi) be the isotonic Pt ? ? regression function on D. From the definition of L , we can see that L (D) = i=1 (yi? ? yi)2 ? L? (D?i) + (yi ? yi?)2 . Thus, the regret increase rt (D) is bounded by rt (D) =

t X (yi ? ybi)2 + L? (D?i) i=1

t

? L? (D) ?

t X (yi ? ybi)2 ? (yi ? y ?)2 i

i=1

1

t

= ‘oot (D).

We assume all points xt are distinct as it will significantly simplify the presentation. All results in this paper are also valid for the case x1 ? . . . ? xT .

3

However, we note that lower bounds for ‘oot (D) do not imply lower bounds on regret. In what follows, our strategy will be to derive bounds ‘oot (D) ? Bt for various algorithms, from PT which the regret bound RT ? t=1 Bt can be immediately obtained. From now on, we abbreviate ‘oot (D) to ‘oot , (as D is clear from the context); we will also consistently assume x1 ¡ . . . ¡ xt . 2.2

Noise free case

As a warm-up, we analyze the noise-free case (when the labels themselves are isotonic) and demonstrate that analyzing ‘oot easily results in an optimal bound for this setting. Proposition 2.2. Assume that the labels satisfy y1 ? y2 ? . . . ? yt . The prediction ybi that is the linear interpolation between adjacent labels ybi = 12 (yi?1 + yi+1), has ‘oot ?

1 1 , and thus RT ? log(T + 1). 2t 2

Pt 1 2 ? Proof. For ?i := yi ? yi?1 , it is easy to check that ‘oot = 4t i=1 (?i+1 ? ?i) because the L (D) term is zero. This expression is a convex function of ?1 , . . . , ?t+1 . Note that ?i ? 0 for each Pt+1 i = 1, . . . , t + 1, and i=1 ?i = 1. Since the maximum of a convex function is at the boundary of the feasible region, the maximizer is given by ?i = 1 for some i ? {1, . . . , t + 1}, and ?j = 0 for all j ? {1, . . . , t + 1}, j 6= i. This implies that ‘oot ? (2t)?1 . 2.3

General Lower Bound

In [14], a general lower bound was derived showing that the regret of any online isotonic regression 1 procedure is at least ?(T 3) for the adversarial setup (when labels and the index order were chosen adversarially). This lower bound applies regardless of the order of outcomes, and hence it is also a lower bound for the random permutation model. This bound translates into ‘oot = ?(t?2/3).

3

Online-to-batch for fixed design

Here, we describe an online-to-batch conversion that relates the adversarial fixed design model with the random permutation model considered in this paper. In the fixed design model with time horizon Tfd the learner is given the points x1 , . . . , xTfd in advance (which is not the case in the random permutation model), but the adversary chooses the order ? in which the labels are revealed (as opposed to ? being drawn at random). We can think of an algorithm for fixed design as a prediction function

ybfd x?t y?1 , . . . , y?t?1 , {x1 , . . . , xTfd } , for any order ?, any set {x1 , . . . , xTfd } (and hence any time horizon Tfd), and any time t. This notation is quite heavy, but makes it explicit that the learner, while predicting at point x?t , knows the previously revealed labels and the whole set of instances. In the random permutation model, at trial t, the learner only knows the previously revealed t ? 1 labeled instances and predicts on the new instance. Without loss of generality, denote the past instances by D?i = {(x1 , y1), . . . , (xi?1 , yi?1), (xi+1 , yi+1), . . . (xt , yt)}, and the new instance by xi , for some i ? {1, . . . , t}. Given an algorithm for fixed design ybfd , we construct a prediction ybt = ybt (D?i , xi) of the algorithm in the random permutation model. The reduction goes through an online-to-batch conversion. Specifically, at trial t, given past labeled instances D?i , and a new point xi , the learner plays the expectation of the prediction of the fixed design algorithm with time horizon T fd = t and points {x1 , . . . , xt } under a uniformly random time from the past j ? {1, . . . , t} and a random permutation ? on {1, . . . , t}, with ?t = i, i.e.2 X

t

1 fd ybt = E{?:?t =i} yb (xi —y?1 , . . . , y?j?1 , {x1 , . . . , xt } . (3) t j=1

2 Choosing the prediction as an expectation is elegant but inefficient. However, the proof indicates that we might as well sample a single j and a single random permutation ? to form the prediction and the reduction would also work in expectation.

4

Note that this is a valid construction, as the right hand side only depends on D?i and xi , which are known to the learner in a random permutation model at round t. We prove (in Appendix A) that the excess leave-one-out loss of yb at trial t is upper bounded by the expected regret (over all permutations) of ybfd in trials 1, . . . , t divided by t: Theorem 3.1. Let D = {(x1 , y1), . . . , (xt , yt)} be a set of t labeled instances. Fix any algorithm ybfd for online adversarial isotonic regression with fixed design, and let Regt (b y fd — ?) denote its regret on D when the labels are revealed in order ?. The random permutation learner yb from (3) ensures y fd — ?)]. ‘oot (D) ? 1t E? [Regt (b 1

? 3) fixed design regret result from [14]. This constructions allows immediate transport of the O(T fd Theorem 3.2. There is an algorithm for the random-permutation model with excess leave-one-out ? ? 23) and hence expected regret RT ? P O(t ? 13). ? ? 32) = O(T loss ‘oot = O(t t

4

Forward Algorithms

For clarity of presentation, we use vector notation in this section: y = (y1 , . . . , yt) is the label vector, y ? = (y1? , . . . , yt?) is the isotonic regression function, and y?i = (y1 , . . . , yi?1 , yi+1 , . . . , yt) is y with i-th label removed. Moreover, keeping in mind that x1 ¡ . . . ¡ xt , we can drop xi ?s entirely from the notation and refer to an instance xi simply by its index i. Given labels y?i and some index i to predict on, we want a good prediction for yi . Follow the Leader (FL) algorithms, which predict using the best isotonic function on the data seen so far, are not directly applicable to online isotonic regression: the best isotonic function is only defined at the observed data instances and can be arbitrary (up to monotonicity constraint) otherwise. Instead, we analyze a simple and natural class of algorithms which we dub forward algorithms3 . We define a forward algorithm, or FA, to be any algorithm that estimates a label yi0 ? [0, 1] (possibly dependent on i and y?i), and plays with the FL strategy on the sequence of past data including the new instance with the estimated label, i.e. performs offline isotonic regression on y

0 , X

t yb = argmin (yj0 ? fj)2 , where y 0 = (y1 , . . . , yi?1 , yi0 , yi+1 , . . . , yt). f1 ?...?ft

j=1

Then, FA predicts with ybi , the value at index i of the offline function of the augmented data. Note that if the estimate turned out to be correct (yi0 = yi), the forward algorithm would suffer no additional loss for that round. Forward algorithms are practically important: we will show that many popular algorithms can be cast as FA with a particular estimate. FA automatically inherit any computational advances for offline isotonic regression; in particular, they scale efficiently to partially ordered data [16]. To our best knowledge, we are first to give bounds on the performance of these algorithms in the online setting. Alternative formulation We can describe a FA using a minimax representation of the isotonic regression [see, e.g., 24]: the optimal isotonic regression y ? satisfies yi? = min max y ‘,r = max min y ‘,r , r?i ‘?i

Pr

‘?i r?i

(4)

yj

j=‘ where y ‘,r = r¿+1 . The ?saddle point? (‘i , ri) for which yi? = y ‘i ,ri , specifies the boundaries of ? the level set {j : yj = yi? } of the isotonic regression function that contains i.

It follows from (4) that isotonic regression is monotonic with respect to labels: for any two label sequences y and z such that yi ? zi for all i, we have yi? ? zi? for all i. Thus, if we denote the predictions for label estimates yi0 = 0 and yi0 = 1 by ybi0 and ybi1 , respectively, the monotonicity implies that any FA has ybi0 ? ybi ? ybi1 . Conversely, using the continuity of isotonic regression y ? as a function of y, (which follows from (4)), we can show that for any prediction ybi with ybi0 ? ybi ? ybi1 , there exists an estimate yt0 ? [0, 1] that could generate this prediction. Hence, we can equivalently interpret FA as an algorithm which in each trial predicts with some ybi in the range [b yi0 , ybi1]. 3

The name highlights resemblance to the Forward algorithm introduced by [2] for exponential family models.

5

4.1

Instances

With the above equivalence between forward algorithms and algorithms that predict in [b yi0 , ybi1], we can show that many of the well know isotonic regression algorithms are forward algorithms and thereby add weight to our next section where we prove regret bounds for the entire class. Isotonic regression with interpolation (IR-Int)[28] Given y?i and index i, the algorithm first ? ? computes f ? , the isotonic regression of y?i , and then predicts with ybiint = 21 fi?1 + fi+1 , where ? we used f0? = 0 and ft+1 = 1. To see that this is a FA, note that if we use estimate yi0 = ybiint , the ? ? , yi0 , fi+1 , . . . , ft?). isotonic regression of y 0 = (y1 , . . . , yi?1 , yi0 , yi+1 , . . . , yt) is yb

= (f1? , . . . , fi?1 ? This is because: i) yb is isotonic by construction; ii) f has the smallest squared error loss for y?t among isotonic functions; and iii) the loss of yb on point yi0 is zero, and the loss of yb on all other points is equal to the loss of f ? . Direct combination of ybi0 and ybi1 . It is clear from Section 4, that any algorithm that predicts ybi = ?i ybi0 + (1 ? ?i)b yi1 for some ?i ? [0, 1] is a FA. The weight ?i can be set to a constant (e.g., ?i = 1/2), or can be chosen depending on ybi1 and ybi0 . Such algorithms were considered by [27]: log-IVAP : ybilog =

ybi1

ybi1 , + 1 ? ybi0

Brier-IVAP : ybiBrier =

1 + (b yi0)2 ? (1 ? ybi1)2 . 2

It is straightforward to show that both algorithms satisfy ybi0 ? ybi ? ybi1 and are thus instances of FA. Last-step minimax (LSM). LSM plays the minimax strategy with one round remaining, n o ybi = argmin max (b y ? yi)2 ? L? (y) , y b?[0,1] yi ?[0,1]

where L? (y) is the isotonic regression loss on y. Define L?b = L? (y1 , . . . , yi?1 , b, yi+1 , . . . , yt) for b ? {0, 1}, i.e. L?b is the loss of isotonic regression function with label estimate yi0 = b. In ? 1+L? 0 ?L1 Appendix B we show that ybi = and it is also an instance of FA. 2 4.2

Bounding the leave-one-out loss q We now give a O(logt t) bound on the leave-one-out loss for forward algorithms. Interestingly, the bound holds no matter what label estimate the algorithm uses. The proof relies on the stability of isotonic regression with respect to a change of a single label. While the bound looks suboptimal in light of Section 2.3, we will argue in Section 4.5 that the bound is actually tight (up to a logarithmic factor) for one FA and experimentally verify that all other mentioned forward algorithms also have a tight lower bound of that form for the same sequence of outcomes. We will bound ‘oot by defining ?i = ybi ? yi? and using the following simple inequality: t

t

t

1X 1 X 2X (b yi ? yi)2 ? (yi? ? yi)2 = (b yi ? yi?)(b yi + yi? ? 2yi) ? —?i —. t i=1 t i=1 t i=1 q

log t Theorem 4.1. Any forward algorithm has ‘oot = O . t ‘oot =

Proof. Fix some forward algorithm. For any i, let {j : yj? = yi? } = {‘i , . . . , ri }, for some ‘i ? i ? ri , be the level set of isotonic regression at level yi? . We need the stronger version of the minimax representation, shown in Appendix C: yi? = min y ‘i ,r = max y ‘,ri . r?i

‘?i

(5)

n h o k We partition the points {1, . . . , t} into K consecutive segments: Sk = i : yi? ? k?1 , for K K

K?1 ? ? k = 1, . . . , K ? 1 and SK = i : yi ? K . Due to monotonicity of y , Sk are subsets of the form {‘k , . . . , rk } (where we use rk = ‘k ? 1 if Sk is empty). From the definition, every level set of y ? is contained in Sk for some k, and each ‘k (rk) is a left-end (right-end) of some level set. 6

Now, choose some index i, and let Sk be such that i ? Sk . Let yi0 be the estimate of the FA, and let y 0 = (y1 , . . . , yi?1 , yi0 , yi+1 , . . . , yt). The minimax representation (4) and definition of FA imply yi0 ? yi o r?i r?i ‘?i r?i r ? ‘k + 1 yi0 ? yi yi0 ? yi ? min y ‘k ,r + min ? min y ‘k ,r + min r?i r?i r ? ‘k + 1 r?i r ? ‘k + 1 r¿k by (5) 1 ?1 1 1 ? y‘?k + min ? y‘?k ? ? yi? ? ? . r?i r ? ‘k + 1 i ? ‘k + 1 K i ? ‘k + 1 1 1 + rk ?i+1 . Hence, we can bound —?i — = —b A symmetric argument gives ybi ? yi? + K yi ? yi? — ?

P —Sk — 1 1 1 i?Sk —?i — ? K + 2 1 + log —Sk — , K + max i¿k +1 , rk ?i+1 . Summing over i ? Sk yields which allows the bound K 2X 2 ‘oot ? —?i — ? + 4 (1 + log t). t i K t p The theorem follows from setting K = ?(t/ log t). n ybi = max min y 0‘,r ? min y 0‘k ,r = min y ‘k ,r +

4.3

Forward algorithms for the well-specified case

p P While the ‘oot upper bound of the previous section yields a regret bound RT ? t O(log t/t) = ? 12) that is a factor O(T 16) gap from the lower bound in Section 2.3, there are two pieces of good O(T news. First, forward algorithms do get the optimal rate in the well-specified setting, popular in the classical statistics literature, where the labels are generated i.i.d. such that E[yi] = ?i with isotonic 1 ?1 ? . . . ? ?t .4 Second, there is a ?(t? 2) lower bound for forward algorithms as proven in the next section. Together, these results imply that the random permutation model in indeed harder than the well-specified case: forward algorithms are sufficient for the latter but not the former. Theorem 4.2. For data generated from the well-specified setting (monotonic means with i.i.d. noise), ? 13) bound on the regret. ? ? 23), which translates to a O(T any FA has ‘oot = O(t The proof is given in Appendix D. Curiously, the proof makes use of the existence of the seemingly ? ? 32) excess leave-one-out loss from Theorem 3.2. unrelated optimal algorithm with O(t 4.4

Entropic loss

We now abandon the squared loss for a moment and analyze how a FA performs when the loss function is the entropic loss, defined as ?y log yb ? (1 ? y) log(1 ? yb) for y ? [0, 1]. Entropic loss (precisely: its binary-label version known as log-loss) is extensively used in the isotonic regression context for maximum likelihood estimation [14] or for probability calibration [28, 21, 27]. A surprising fact in isotonic regression is that minimizing entropic loss5 leads to exactly the same optimal solution as in the squared loss case, the isotonic regression function y ? [24]. Not every FA is appropriate for entropic loss, as recklessly choosing the label estimate might result in an infinite loss in just a single trial (as noted by [27]). Indeed, consider a sequence of outcomes with y1 = 0 and yi = 1 for i ¿ 1. While predicting on index i = 1, choosing y10 = 1 results in yb1 = 1, for which the entropic loss is infinite (as y1 = 0). Does there exists a FA which achieves a meaningful bound on ‘oot in the entropic loss setup? We answer this question in the affirmative, showing that the log-IVAP predictor FA gets the same excess-leave-one-out loss bound as given in Theorem

4.1. As the reduction from the regret to leaveone-out loss (Lemma 2.1) does not use any properties of the loss function, this immediately implies a bound on the expected regret. Interestingly, the proof (given in Appendix G) uses as an intermediate step the bound on —?i — for the worst possible forward algorithm which always produces the estimate being the opposite of the actual label. q

log t Theorem 4.3. The log-IVAP algorithm has ‘oot = O for the entropic loss. t 4 5

The ?(T 1/3) regret lower bound in [14] uses a mixture of well-specified distributions and still applies. In fact, this statement applies to any Bregman divergence [24].

7

4.5

Lower bound 1

The last result of this section is that FA can be made to have ‘oot = ?(t? 2). We show this by means of a counterexample. Assume t = n2 for some integer n ¿ 0 and let the labels ? be binary, yi ? {0, 1}. We split the set {1, . . . , t} into n consecutive segments, each of size n = t. The proportion of ones (yi = 1) in the k-th segment is equal to nk , but within each segment all ones precede all zeros. For instance, when t = 25, the label sequence is: 10000 — {z } 11100 — {z } 11110 — {z } 11111 — {z }, — {z } 11000 1/5

2/5

3/5

4/5

5/5

One can use the minimax formulation (4) to verify that the segments will correspond to the level sets of the isotonic regression and that yi? = nk for any i in the k-th segment. This sequence is hard: 1

Lemma 4.4. The IR-Int algorithm run on the sequence described above has ‘oot = ?(t? 2). We prove the lower bound for IR-Int, since the presentation (in Appendix E) is clearest. Empirical simulations showing that the other forward algorithms also suffer this regret are in Appendix F. 4.6

Towards optimal forward algorithms

An attractive feature of forward algorithms is that they generalize to partial orders, for which efficient ? ? 12) offline optimization algorithms exist. However, in Section 4 we saw that FAs only give a O(t ? ? 23) is possible (with an algorithm that is not known to scale rate, while in Section 3 we saw that O(t to partial orders). Is there any hope of an algorithm that both generalizes and has the optimal rate? In this section, we propose the Heavy-? algorithm, a slight modification of the forward algorithm that plugs in label estimate yi0 = ? ? [0, 1] with weight c (with unit weight on all other points), then plays the value of the isotonic regression function. Implementation is straightforward for offline isotonic regression algorithms that permit the specification of weights (such as [16]). Otherwise, we might simulate such weighting by plugging in c copies of the estimated label ? at location xi . What label estimate ? and weight c should we use? We show that the choice of ? is not very sensitive, 1 but it is crucial to tune the weight to c = ?(t 3). Lemmas H.1 and H.2 show that higher and lower c are necessarily sub-optimal for ‘oot . This leaves only one choice for c, for which we believe 1 ? ? 23). Conjecture 4.5. Heavy-? with weight c = ?(t 3) has ‘oot = O(t We cannot yet prove this conjecture, although numerical experiments strongly suggest it. We do not believe that picking a constant label ? is special. For example, we might alternatively predict with the average of the predictions of Heavy-1 and Heavy-0. Yet not any label estimate works. In particular, if we estimate the label that would be predicted by IR-Int (see 4.1) and the discussion below it), and we plug that in with any weight c ? 0, then the isotonic regression function will still have that same 1 label estimate as its value. This means that the ?(t? 2) lower bound of Section 4.5 applies.

5

Conclusion

We revisit the problem of online isotonic regression and argue that we need a new perspective to design practical algorithms. We study the random permutation model as a novel way to bypass the stringent fixed design requirement of previous work. Our main tool in the design and analysis of algorithms is the leave-one-out loss, which bounds the expected regret from above. We start by observing that the adversary from the adversarial fixed design setting also provides a lower bound here. We then show that this lower bound can be matched by applying online-to-batch conversion to the optimal algorithm for fixed design. Next we provide an online analysis of the natural, popular and practical class of Forward Algorithms, which are defined in terms of an offline optimization oracle. We show that Forward algorithms achieve a decent regret rate in all cases, and match the optimal rate in special cases. We conclude by sketching the class of practical Heavy algorithms and conjecture that a specific parameter setting might guarantee the correct regret rate. Open problem The next major challenge is the design and analysis of efficient algorithms for online isotonic regression on arbitrary partial orders. Heavy-? is our current best candidate. We pose ? 13) regret on linear orders as an open problem. deciding if it in fact even guarantees O(T 8

Acknowledgments Wojciech Kot?owski acknowledges support from the Polish National Science Centre (grant no. 2016/22/E/ST6/00299). Wouter Koolen acknowledges support from the Netherlands Organization for Scientific Research (NWO) under Veni grant 639.021.439. This work was done in part while Koolen was visiting the Simons Institute for the Theory of Computing.

2 References

[1] 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. [2] 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. [3] Lucien Birg? and Pascal Massart. Rates of convergence for minimum contrast estimators. Probability Theory and Related Fields, 97:113?150, 1993.

[4] H. D. Brunk. Maximum likelihood estimates of monotone parameters. Annals of Mathematical Statistics, 26(4):607?616, 1955. [5] Nicol? Cesa-Bianchi and G?bor Lugosi. Worst-case bounds for the logarithmic loss of predictors. Machine Learning, 43(3):247?264, 2001. [6] Nicol? Cesa-Bianchi and G?bor Lugosi. Prediction, learning, and games. Cambridge University Press, 2006. [7] Jan de Leeuw, Kurt Hornik, and Patrick Mair. Isotone optimization in R: Pool-adjacent-violators algorithm (PAVA) and active set methods. Journal of Statistical Software, 32:1?24, 2009. [8] Tom Fawcett and Alexandru Niculescu-Mizil. PAV and the ROC convex hull. Machine Learning, 68(1):97?106, 2007. [9] J?rgen Forster and Manfred K Warmuth. Relative expected instantaneous loss bounds. Journal of Computer and System Sciences, 64(1):76?102, 2002. [10] Pierre Gaillard and S?bastien Gerchinovitz. A chaining algorithm for online nonparametric regression. In Conference on Learning Theory (COLT), pages 764?796, 2015. [11] Sham M Kakade, Varun Kanade, Ohad Shamir, and Adam Kalai. Efficient learning of generalized linear and single index models with isotonic regression. In Neural Information Processing Systems (NIPS), pages 927?935, 2011. [12] Adam Tauman Kalai and Ravi Sastry. The isotron algorithm: High-dimensional isotonic regression. In COLT, 2009. [13] Wojciech Kot?owski and Roman S?owi?nski. Rule learning with monotonicity constraints. In International Conference on Machine Learning (ICML), pages 537?544, 2009. [14] Wojciech Kot?owski, Wouter M. Koolen, and Alan Malek. Online isotonic regression. In Vitaly Feldman and Alexander Rakhlin, editors, Proceedings of the 29th Annual Conference on Learning Theory (COLT), pages 1165?1189, June 2016. [15] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika, 29(1):1?27, 1964. [16] Rasmus Kyng, Anup Rao, and Sushant Sachdeva. Fast, provable algorithms for isotonic regression in all ‘p -norms. In Neural Information Processing Systems (NIPS), 2015. [17] Ronny Luss, Saharon Rosset, and Moni Shahar. Efficient regularized isotonic regression with application to gene?gene interaction search. Annals of Applied Statistics, 6(1):253?283, 2012. 9

[18] Aditya Krishna Menon, Xiaoqian Jiang, Shankar Vembu, Charles Elkan, and Lucila OhnoMachado. Predicting accurate probabilities with a ranking loss. In Interantional Conference on Machine Learning (ICML), 2012. [19] T. Moon, A. Smola, Y. Chang, and Z. Zheng. Intervalrank: Isotonic regression with listwise and pairwise constraint. In WSDM, pages 151?160. ACM, 2010. [20] Harikrishna Narasimhan and Shivani Agarwal. On the relationship between binary classification, bipartite ranking, and binary class probability estimation. In Neural Information Processing Systems (NIPS), pages 2913?2921, 2013. [21] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In ICML, volume 119, pages 625?632. ACM, 2005. [22] 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. [23] Alexander Rakhlin and Karthik Sridharan. Online nonparametric regression. In Conference on Learning Theory (COLT), pages 1232?1264, 2014. [24] T. Robertson, F. T. Wright, and R. L. Dykstra. Order Restricted Statistical Inference. John Wiley & Sons, 1998. [25] Mario Stylianou

and Nancy Flournoy. Dose finding using the biased coin up-and-down design and isotonic regression. Biometrics, 58(1):171?177, 2002. [26] Sara Van de Geer. Estimating a regression function. Annals of Statistics, 18:907?924, 1990. [27] Vladimir Vovk, Ivan Petej, and Valentina Fedorova. Large-scale probabilistic predictors with and without guarantees of validity. In Neural Information Processing Systems (NIPS), pages 892?900, 2015. [28] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In International Conference on Knowledge Discovery and Data Mining (KDD), pages 694?699, 2002. [29] Cun-Hui Zhang. Risk bounds in isotonic regression. The Annals of Statistics, 30(2):528?555, 2002.

10