The SMART Approach to Instance-Optimal Online Learning

Siddhartha Banerjee
ORIE, Cornell
sbanerjee@cornell.edu

Alankrita Bhatt
CMS, Caltech
abhatt@caltech.edu

Christina Lee Yu
ORIE, Cornell
cleeyu@cornell.edu

Abstract

We devise an online learning algorithm – titled *Switching via Monotone Adapted Regret Traces* (SMART) – that adapts to the data and achieves regret that is *instance optimal*, i.e., simultaneously competitive on every input sequence compared to the performance of the follow-the-leader (FTL) policy and the worst case guarantee of any other input policy $\mathsf{ALG}_{\mathsf{WC}}$. We show that the regret of the SMART policy on *any* input sequence is within a multiplicative factor $e/(e-1) \approx 1.58$ of the smaller of: 1) the regret obtained by FTL on the sequence, and 2) the upper bound on regret guaranteed by the given worst-case policy. This implies a strictly stronger guarantee than typical 'best-of-both-worlds' bounds as the guarantee holds for every input sequence regardless of how it is generated. SMART is simple to implement as it begins by playing FTL and switches at most once during the time horizon to $\mathsf{ALG}_{\mathsf{WC}}$. Our approach and results follow from an operational reduction of instance optimal online learning to competitive anaylsis for the ski-rental problem. We complement our competitive ratio upper bounds with a fundamental lower bound showing that over all input sequences, no algorithm can get better than a 1.43-fraction of the minimum regret achieved by FTL and the minimax-optimal policy. We also present a modification of SMART that combines FTL with a "small-loss" algorithm to achieve instance optimality between the regret of FTL and the small loss regret bound.

1 Introduction

Our work aims to develop algorithms for online learning that are *instance optimal* [Fagin et al., 2001],[Roughgarden, 2021, Chapter 3] with respect to the stochastic and minimax optimal algorithms for a given setting. This is best motivated via a concrete example:

Example 1 (Binary Prediction)**.** *We are given bit stream $y^n := y_1, y_2, \ldots, y_n \in \{0,1\}^n$. At the start of day t, before seeing y_t, we choose (possibly randomized) prediction $\widehat{Y}_t \sim$ Bernoulli(a_t) (for $a_t \in [0,1]$) for the upcoming bit y_t, given the history y^{t-1}. Our resulting loss on day t is $\ell_t(a_t) = \mathbb{P}(\widehat{Y}_t \neq y_t) = |a_t - y_t|$, and our total loss is $L_n(\mathsf{ALG}, y^n) := \sum_{t=1}^n \ell_t(a_t)$. The objective is to achieve low* regret *(i.e., additive loss) compared to the loss $L_n(a, y^n) = \sum_{t=1}^n \ell_t(a)$ of the best fixed action $a^* \in [0,1]$ in hindsight. As a^* is the majority in y^n between 0 and 1, it follows that $L_n(a^*, y^n) = \min\left\{\sum_{t=1}^n y_t, n - \sum_{t=1}^n y_t\right\}$. Formally, for sequence $y^n \in \{0,1\}^n$, policy ALG incurs regret*

$$\mathrm{Reg}(\mathsf{ALG}, y^n) := L_n(\mathsf{ALG}, y^n) - L_n(a^*, y^n) = \sum_{t=1}^n |a_t - y_t| - \min_{a\in[0,1]} \sum_{t=1}^T |a - y_t|. \quad (1)$$

Binary prediction goes back to the seminal works of Blackwell [1956] and Hannan [1957]. The definition of regret is motivated by the case where y_t is *randomly* generated as i.i.d. Bernoulli(p). If

p is known, then the optimal policy is the 'Bayes predictor' $a^{\mathsf{Bayes}} = \lfloor 2p \rfloor$ (i.e., nearest integer to p), which coincides with hindsight optimal a^* with high probability when p is away from $1/2$. When p is unknown, the *stochastic optimal* policy is the *Follow The Leader* or FTL policy, which sets $a_t = \mathsf{Majority}(y^{t-1})$, i.e. the majority bit amongst the first $t-1$ bits ($a_t = 1/2$ if both are equal[1]).

A starting point for online learning is the observation that it is easy to construct a sequence y^n such that FTL has poor regret: For example, if $y^n = (1, 0, 1, 0, 1, 0, \ldots)$, i.e., alternate 1s and 0s, then the regret of FTL grows linearly with n. In contrast, *worst-case optimal* online learning policies such as those of Blackwell and Hannan, or more modern versions like Multiplicative Weights or Follow The Perturbed Leader (see Cesa-Bianchi and Lugosi [2006], Slivkins [2019]) guarantee regret of $\Theta(\sqrt{n})$ over all sequences. Indeed, for bit prediction, the *exact* minimax optimal policy was established by Cover [1966], and this policy (which we refer to as Cover) achieves[2] $\mathrm{Reg}(\mathsf{Cover}, y^n) = \sqrt{\frac{n}{2\pi}}(1 + o(1))$ under *any* $y^n \in \{0,1\}^n$, implying it is an *equalizer* (achieves same regret over all sequences).

While the above discussion seems a convincing endorsement of worst-case online learning algorithms, the situation is more complicated. One problem is that while FTL has bad regret on certain pathological sequences, on more 'realistic' sequences FTL performs orders of magnitude better than the minimax regret. As an example, with i.i.d. Bernoulli(p) input, $\mathrm{Reg}(\mathsf{FTL}, y^n)$ is actually *independent of* n (i.e., $O(1)$) as long as p is away from $1/2$ with high probability. We demonstrate this in Figure 1(a), where we see $\mathrm{Reg}(\mathsf{FTL})$ is much lower than $\mathrm{Reg}(\mathsf{Cover}) \approx 0.39\sqrt{n}$ unless p is very close to $1/2$. This phenomena is known in more general settings [Huang et al., 2016], suggesting that in practice one may be better off just using FTL. On the other hand, as Figure 1(b, c) indicates, we know how to generate sequences y^n [Feder et al., 1992] for which $\mathrm{Reg}(\mathsf{FTL}, y^n)$ grows linearly with n, and so the $\sqrt{n}$ regret of Cover becomes appealing.

Now suppose instead that a fictitious oracle is told beforehand which of FTL or Cover is better suited for the upcoming sequence y^n; the demand made by *instance optimality* is that we try to be competitive against such an oracle *on every sequence* y^n.

Definition 1 (Instance Optimality)**.** *A binary prediction policy* ALG *is instance optimal with respect to the regret of* FTL *and* Cover *if there exists some universal* $\gamma_n \geq 1$ *such that for all* $y^n \in \{0,1\}^n$*:*

$$\mathrm{Reg}(\mathsf{ALG}, y^n) \leq \gamma_n \min\{\mathrm{Reg}(\mathsf{FTL}, y^n), \mathrm{Reg}(\mathsf{Cover}, y^n)\}$$

We henceforth refer to γ_n as the competitive ratio achieved by ALG; ideally we want this ratio to be a constant, i.e., $\gamma_n = O(1)$. This necessitates that on sequences where FTL gets a constant regret, then ALG basically follows FTL throughout, while on sequences where FTL has high (in particular, $\omega(\sqrt{n})$) regret, then ALG follows Cover in most rounds.

The challenge in designing instance optimal algorithms is that the regret of any algorithm is a quantity that is not adapted to the natural filtration, i.e. it may not be possible to track $\mathrm{Reg}(\mathsf{ALG}, y^n)$ for any ALG from just the history $(y_1, y_2, \ldots, y_{t-1})$, since the hindsight optimal action a^* depends on the *entire* sequence y^n. One proxy is to track an algorithm's loss instead, leading to the idea of 'corralling' policies [Agarwal et al., 2017, Pacchiano et al., 2020, Dann et al., 2023], that run online learning over the reference algorithms to get within $O(\mathrm{poly}(n))$ of the smaller of the two losses. Such an approach can not ensure $\gamma_n = O(1)$: for example, consider an i.i.d. sequence of Bernoulli(0.1) bits, where FTL has lower regret than Cover. With high probability on any such sequence we have small $\mathrm{Reg}(\mathsf{FTL}, y^n) = O(1)$ and yet high loss $L_n(a^*, y^n) = \Theta(n)$; now any corralling algorithm (even a small loss one) must suffer $O(\mathrm{poly}(n))$ regret, and hence $\omega(1)$

[1] We choose this specific tie-breaking rule for convenience; however, we can take any $a_t \in [0,1]$.

[2] Here f_n, the so-called Rademacher complexity of the setting, is a *fixed* function of n that does not depend on sequence y^n. For binary prediction, $f_n = \frac{\mathbb{E}\,|\sum_{t=1}^n Z_t|}{2} \approx \sqrt{\frac{n}{2\pi}}$ where $Z^n \sim \mathrm{Unif}\{1, -1\}$ i.i.d.

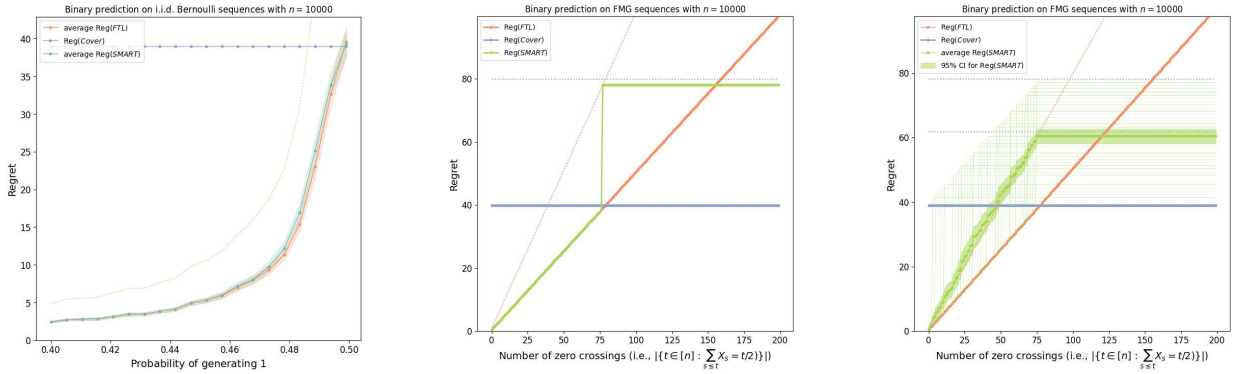

Figure 1: *Comparing regret of* FTL, Cover *and* SMART *on a collection of input sequences (for fixed n).*
• *In Fig. (a), we consider i.i.d. Bernoulli(p) inputs for varying p. The regret of* FTL *is much lower than* Cover *for $p < 1/2$; the regret of* SMART *tracks* FTL *closely (better than* $2\mathrm{Reg}(\mathsf{FTL})$*, indicated by dotted line).*
• *In Fig. (b) and (c), we consider 'worst-case' binary sequences (as per [Feder et al., 1992]) parameterized by the number of 'lead-changes': the sequence with parameter c comprises of c pairs '$0,1$' or '$1,0$', followed by $n-2c$ '1's. In Fig. (b), we consider* SMART *with a deterministic switching threshold (Theorem 1) and compare* $\mathrm{Reg}(\mathsf{SMART})$ *with* $2\mathrm{Reg}(\mathsf{FTL})$ *and* $2\mathrm{Reg}(\mathsf{Cover})$ *(dotted lines); in Fig. (c), we use a randomized threshold (Theorem 2), and show the average regret over the randomized threshold, as well as sample paths (plotted in green), and compare with* $\frac{e}{e-1}$ *times* $\mathrm{Reg}(\mathsf{FTL})$ *and* $\mathrm{Reg}(\mathsf{Cover})$ *(dotted lines).*

competitive ratio. This example also shows that achieving a constant factor guarantee with respect to the minimum of the two *losses* does not translate to a constant factor guarantee with respect to the minimum of the two *regrets*.

The instance optimal guarantee is closely related to *best-of-both-worlds* guarantees, which aim for algorithms that simultaneously achieve (up to constant factors) both the low *pseudoregret* guarantee of policies designed for stochastic inputs (as with FTL in our setting, or the Upper Confidence Bound (UCB) algorithm in bandits), as well as a per-sequence regret guarantee comparable to a worst-case optimal algorithm $\mathsf{ALG}_{\mathsf{WC}}$ (Eg. Cover or Hedge in online learning; EXP3 in bandits Auer et al. [2002a]). Such guarantees have been shown in a variety of settings, including online learning [De Rooij et al., 2014, Orabona and Pál, 2015, Mourtada and Gaïffas, 2019, Bilodeau et al., 2023] and bandit settings [Bubeck and Slivkins, 2012, Zimmert and Seldin, 2019, Lykouris et al., 2018, Dann et al., 2023]. One problem though is that since pseudoregret and worst-case regret are very different quantities, the above results tend to be hard to interpret, and less predictive of good performance[3]. Note though that given a pair of stochastic/worst-case optimal algorithms, a policy that is γ-*instance-optimal* w.r.t. these immediately satisfies a best-of-both-worlds guarantee with constant factor γ. In this regard, instance optimality provides a stronger guarantee as it holds on *every* sequence y^n regardless of how it is generated. Moreover, the parameter γ can also provide sharper comparisons between algorithms, as well as admit hardness results on the limits of such guarantees.

1.1 Our Contributions

We consider a general online learning setting where at the beginning of each round $t \in [n]$, a policy ALG first plays an action $a_t \in \mathcal{A}$, following which, a loss function $\ell_t : \mathcal{A} \to [0,1]$ is revealed, resulting in a loss of $\ell_t(a_t)$. The *regret* is defined according to:

$$\mathrm{Reg}(\mathsf{ALG}, \ell^n) = \textstyle\sum_{t=1}^n \ell_t(a_t) - \inf_{a \in \mathcal{A}} \sum_{t=1}^n \ell_t(a). \tag{2}$$

[3]As an example, Hedge has optimal pseudoregret in certain stochastic settings [Mourtada and Gaïffas, 2019], but this is known to be sensitive to perturbations in the distributions [Bilodeau et al., 2023].

More generally, as in with bit prediction, we allow ALG to play in round t a measure $w_t \in \Delta_{\mathcal{A}}$ (i.e., play $\{w_t : \mathcal{A} \mapsto [0,1] | \sum_{a\in\mathcal{A}} w_t(a) = 1\}$), resulting in an expected loss of $\sum_{a\in\mathcal{A}} w_t(a)\ell_t(a)$. For notational convenience, we henceforth use $(a_t, \ell_t(a_t))$ for the action/loss, and reserve use of expectations for randomness in the algorithm and/or sequence.

We want to understand when is it possible to attain instance optimality as in Eq. (1) with respect to a given pair of algorithms. Ideally, we want the first to be optimal for stochastic instances, and the second to be minimax optimal; unfortunately however exact optimal policies are unknown except in simple settings. To this end, we make two amendments to our goal: First, for the stochastic optimal policy, we use FTL; this is well defined in any online learning setting, and moreover, known to be optimal or near-optimal for a wide range of settings under minimal assumptions [Kotłowski, 2018]. Second, instead of the minimax policy, we use as reference *any* policy $\mathsf{ALG}_{\mathsf{WC}}$ which has a known worst case regret bound $g(n)$. With these modifications in place, we have the following objective.

Definition 2. *Given* FTL *and any algorithm* $\mathsf{ALG}_{\mathsf{WC}}$ *with* $\sup_{\ell^n} \mathrm{Reg}(\mathsf{ALG}_{\mathsf{WC}}, \ell^n) \le g(n)$*, we say a policy* ALG *is instance optimal with respect to the pair if there exists some universal* $\gamma_n \ge 1$ *(i.e. not depending on* y^n*) such that for every sequence of losses* ℓ^n*:*

$$\mathrm{Reg}(\mathsf{ALG}, \ell^n) \le \gamma_n \min\{\mathrm{Reg}(\mathsf{FTL}, \ell^n), g(n)\}$$

While the above guarantee is not truly instance-optimal in that we are comparing against a worst-case regret bound $g(n)$ for $\mathsf{ALG}_{\mathsf{WC}}$ rather than its performance on the instance ℓ^n, the two are the same if $\mathsf{ALG}_{\mathsf{WC}}$ is minimax optimal and hence attaining equal regret on all loss sequences; recall this is true of Cover for binary prediction.

To realize the above goal, we propose the *Switch via Monotone Adapted Regret Traces* (SMART) approach, which at a high level is a black-box way to convert design of instance-optimal policies into a simple *optimal stopping* problem. Our approach depends on just two ingredients: first, owing to the additive structure of online learning problems, we have that the minimax guarantee $g(k)$ above holds over *any* $k \in \mathbb{Z}$ and any (sub)sequence of n loss functions; second, we show that FTL admits simple anytime regret estimator $\Sigma_\tau^{\mathsf{FTL}}$ (see Lemma 1) which is *monotone* and *adapted* (i.e., a function only of historical data). Using these two observations, we can reduce the task of minimizing regret to a version of the 'ski-rental' problem [Karlin et al., 1994, Borodin and El-Yaniv, 2005], as follows: we play FTL up to some stopping time τ, and then switch to $\mathsf{ALG}_{\mathsf{WC}}$ for the remaining $n-\tau$ periods, resetting all losses to zero. This algorithm incurs a *total* regret bounded by $\Sigma_\tau^{\mathsf{FTL}} + g(n-\tau)$, and using ideas from competitive analysis, we get that there is a simple way to choose the stopping time τ to achieve an $e/(e-1) \approx 1.58$-competitive ratio guarantee with respect to the minimum between the regret of FTL and the worst case guarantee $g(n)$.

Theorem. (See Theorem 2) *Let* $\mathsf{ALG}_{\mathsf{WC}}$ *have worst-case regret* $\sup_{\ell^n} \mathrm{Reg}(\mathsf{ALG}_{\mathsf{WC}}, \ell^n) \le g(n)$ *where* $g(n)$ *is some monotonic function of* n*. An instantiation of* SMART *achieves*

$$\mathrm{Reg}(\mathsf{SMART}, \ell^n) \le \frac{e}{e-1} \min\{\mathrm{Reg}(\mathsf{FTL}, \ell^n), g(n)\} + 1. \tag{3}$$

A highlight of our approach is the surprising simplicity of the algorithm and analysis, despite the strength of the instance optimality guarantee. In particular, our approach is modular, allowing one to plug in any $\mathsf{ALG}_{\mathsf{WC}}$ and corresponding worst case bound $g(n)$, thus letting us handle any online learning setting with known minimax bounds. This results in an entire family of instance optimal policies for settings such as predictions with experts and online convex optimization. Moreover, the approach is easy to extend to get more complex guarantees; as an example, if $\mathsf{ALG}_{\mathsf{WC}}$ is designed

to get low regret for benign (i.e., 'small-loss') sequences ℓ^n, then we show how to use SMART as a subroutine and achieve an instance optimal guarantee with respect to the regret of FTL and a small loss regret bound.

Corollary 1 (Following Theorem 5)**.** *Consider the prediction with expert advice setting [Cesa-Bianchi et al., 1997], where $\mathcal{A} = \Delta^{m-1}$, the $m-$simplex for $m \geq 2$, and $\ell_t(a) = \langle a, \ell_t \rangle$ for $\ell_t \in [0,1]^m$. Let $L^* := \min_j \sum_{t=1}^n \ell_{tj}$. An instantiation of* SMART *achieves*

$$\mathrm{Reg}(\mathsf{SMART}, \ell^n) \leq 2 \min \left\{ \mathrm{Reg}(\mathsf{FTL}, \ell^n), 10\sqrt{2L^* \log m} \right\} + O(\log L^* \log m).$$

Finally, studying instance optimality also lets us understand the fundamental limits of best-of-both worlds algorithms. To this end, we provide a lower bound that shows our algorithm is nearly optimal in the competitive ratio. To the best of our knowledge, this is the first hardness result for best-of-both-worlds guarantees in online learning.

Theorem. (See Theorem 3) *In the binary prediction setting, given* any *online algorithm* ALG*, there exist sequences $y^n \in \{0,1\}^n$ such that:*

$$\mathrm{Reg}(\mathsf{ALG}, y^n) \geq 1.43 \min \left\{ \mathrm{Reg}(\mathsf{FTL}, y^n), \mathrm{Reg}(\mathsf{Cover}, y^n) \right\}$$

Note again that in binary prediction, FTL achieves the optimal pseudoregret under i.i.d. inputs, while Cover is the true minimax policy; thus, this is a fundamental lower bound on best-of-both-worlds guarantee in any online learning setting.

1.2 Related work

There have been many approaches towards combining stochastic and worst-case guarantees. As we discussed before, there is a large literature on best-of-both-worlds algorithms for both full and partial information settings [Wei and Luo, 2018, Bubeck et al., 2019, Zimmert and Seldin, 2019, Dann et al., 2023], and also more complex settings such as metrical task systems [Bhuyan et al., 2023] and control [Sabag et al., 2021, Goel et al., 2023]. Another line of work [Rakhlin et al., 2011, Haghtalab et al., 2022, Block et al., 2022, Bhatt et al., 2023] considers *smoothed analysis*, where the worst-case actions of the adversary are perturbed by nature. A third approach [Bubeck and Slivkins, 2012, Lykouris et al., 2018, Amir et al., 2020, Zimmert and Seldin, 2019] interpolates between the stochastic and adversarial settings by considering most ℓ_t to be i.i.d., interspersed with a few adversarially chosen instances (corruptions). Finally, another line considers the data-generating distribution to come from a ball of specified radius around i.i.d. distributions [Mourtada and Gaïffas, 2019, Bilodeau et al., 2023]. While all these approaches provide useful insights into the gap between average and worst-case guarantees, one can argue they are all imprecisely specified – given an instance $\{\ell_t\}_{t\in[n]}$ in hindsight, there is no clear sense as to which model best 'explains' the instance.

Our focus on instance optimality instead follows the approach of better understanding and shaping the per-sequence regret landscape. The origins of this approach arguably come from the seminal work of Cover [1966] for binary prediction (we discuss this in more detail in Section 3), with a later focus on better bounds for benign instances in general online learning [Auer et al., 2002b, Cesa-Bianchi et al., 2005, Hazan and Kale, 2010]. More recently, a line of work [Koolen et al., 2014, Van Erven et al., 2015, Van Erven and Koolen, 2016, Gaillard et al., 2014] have studied designing policies that can adapt to different types of data sequences and achieve multiple performance guarantees simultaneously. The main idea is to use multiple learning rates that are weighted according to their empirical performance on the data. While the focus is still primarily on classifying instances based on when they are easier/harder to learn, some of the resulting guarantees have an

instance-optimality flavor; for example, Van Erven and Koolen [2016] show how to simultaneously match the performance guarantee (in terms of certain variance bounds) attained by different learning rates in Hedge. Such approaches however need to understand their baseline algorithms in great detail, and use them in a 'white-box' way to get their guarantees.

In contrast, our approach fundamentally focuses on combining policies in a black-box way to get instance optimal outcomes. As we mention, this is similar in spirit to corralling bandit algorithms Agarwal et al. [2017], Pacchiano et al. [2020], Dann et al. [2023], as well as more recent work on online algorithms with predictions Bamas et al. [2020], Dinitz et al. [2022], Anand et al. [2022]; however, as we mention above, these all get guarantees with respect to the loss of the reference algorithms, which is much weaker than our regret guarantees (though they do so in much more complex settings with partial information and/or state). To our knowledge, the only previous result which attains a comparable instance-optimality guarantee to ours is that of De Rooij et al. [2014] for the experts problem, where the authors propose the FlipFlop policy which interleaves Hedge (with varying learning rates) and FTL to obtain a regret guarantee similar to that of Corollary 1. In fact, their guarantee is stronger as FlipFlop is shown to be 5.64-competitive with respect to $\min\{\mathrm{Reg}(\mathsf{FTL}, \ell^n), g(L^*)\}$ where $g(L^*) \leq \sqrt{L^* \log m}$ [De Rooij et al., 2014, Corollary 16]. However, while FlipFlop depends on a clever choice of learning rates in Hedge, SMART can black-box interleave FTL with *any* worst-case/small-loss algorithm without knowing the inner workings of said algorithm, which we see as a significant engineering strength. More importantly, our approach to this problem is distinct as we focus on the fundamental limits (upper and lower bounds) on the *competitive ratio* that must be incurred when combining FTL with a worst-case policy; to the best of our knowledge this viewpoint, and the corresponding reduction to an optimal stopping problem, has not been previously explored.

2 Instance Optimal Online Learning: Achievability via SMART

Given the setting and problem statement above, we can now present the SMART policy. We do this for a general online learning problem, wherein we want to combine FTL with any given algorithm $\mathsf{ALG}_{\mathsf{WC}}$ with a worst-case regret guarantee $\sup_{\ell^n} \mathrm{Reg}(\mathsf{ALG}_{\mathsf{WC}}, \ell^n) \leq g(n)$. Before presenting the policy, we first need the following regret decomposition for FTL.

Lemma 1 (Regret of FTL)**.** *If $L_t(\cdot) := \sum_{i=1}^t \ell_t(\cdot)$, i.e. the cumulative loss function, then*

$$\mathrm{Reg}(\mathsf{FTL}, \ell^n) = \textstyle\sum_{t=1}^n (L_t(a^*_{t-1}) - L_t(a^*_t)).$$

This is reminiscent of the 'be-the-leader' lemma [Kalai and Vempala, 2005, Slivkins, 2019], although never stated explicitly as an exact decomposition.

Proof. Recall we define $a_t^{\mathsf{FTL}} = a^*_{t-1}$, and hence $\inf_{a \in \mathcal{A}} \sum_{t=1}^n \ell_t(a) = L_n(a^*_n)$. Now we have

$$\begin{aligned}
\mathrm{Reg}(\mathsf{FTL}, \ell^n) &= \textstyle\sum_{t=1}^n \ell_t(a^*_{t-1}) - L_n(a^*_n) \\
&= \textstyle\sum_{t=1}^n (L_t(a^*_{t-1}) - L_{t-1}(a^*_{t-1})) - L_n(a^*_n) \\
&= \textstyle\sum_{t=1}^n (L_t(a^*_{t-1}) - L_t(a^*_t)) + \sum_{t=1}^n (L_t(a^*_t) - L_{t-1}(a^*_{t-1})) - L_n(a^*_n) \\
&\overset{(a)}{=} \textstyle\sum_{t=1}^n (L_t(a^*_{t-1}) - L_t(a^*_t))
\end{aligned}$$

where (a) follows since $\sum_{t=1}^n (L_t(a^*_t) - L_{t-1}(a^*_{t-1})) = L_n(a^*_n)$ by telescoping. □

Next, let Σ_t^{FTL} denote the *anytime* regret that FTL incurs up to round t (i.e., assuming the game ends after round t). From Lemma 1 we have

$$\Sigma_t^{\mathsf{FTL}} := \mathrm{Reg}(\mathsf{FTL}, \ell^t) = \textstyle\sum_{i=1}^{t}(L_i(a_{i-1}^*) - L_i(a_i^*)) \tag{4}$$

Now we make three critical observations:

- Σ_t^{FTL} is **adapted**: it can be computed at the end of the t^{th} round
- Σ_t^{FTL} is **monotone** non-decreasing in t (since by definition $L_i(a_{i-1}^*) - L_i(a_i^*) \geq 0$)
- Σ_t^{FTL} is an **anytime lower bound** for $\mathrm{Reg}(\mathsf{FTL}, \ell^n)$, with $\Sigma_n^{\mathsf{FTL}} = \mathrm{Reg}(\mathsf{FTL}, \ell^n)$

For an input threshold $\theta \geq 0$ and an algorithm $\mathsf{ALG_{WC}}$, we get the following (meta)algorithm.

Algorithm 1: Switching via Monotone Adapted Regret Traces (SMART)

Input: Policies $\mathsf{FTL}, \mathsf{ALG_{WC}}$, threshold θ
Initialize $\Sigma_0^{\mathsf{FTL}} = 0$, $t = 1$;
while $\Sigma_{t-1}^{\mathsf{FTL}} \leq \theta$ **do**
  Set $a_t = a_{t-1}^*$; `// Play FTL`
  Observe $\ell_t(\cdot)$;
  Update $L_t(\cdot) = L_{t-1}(\cdot) + \ell_t(\cdot)$ and $\Sigma_t^{\mathsf{FTL}} = \Sigma_{t-1}^{\mathsf{FTL}} + (L_t(a_{t-1}^*) - L_t(a_t^*))$ and $t = t + 1$;
end
Reset losses to 0 and play $\mathsf{ALG_{WC}}$ for remaining rounds (See Figure 2(b)) ;

We now have the following performance guarantee for Algorithm 1 for $\theta = g(n)$.

Theorem 1 (Regret of SMART with deterministic threshold)**.** *We are given* FTL*, and any other policy* $\mathsf{ALG_{WC}}$ *with worst-case regret* $\sup_{\ell^n} \mathrm{Reg}(\mathsf{ALG_{WC}}, \ell^n) \leq g(n)$ *for some monotone function* g*. Then, playing* SMART *with threshold* $\theta = g(n)$ *ensures*

$$\mathrm{Reg}(\mathsf{SMART}, \ell^n) \leq 2\min\{\mathrm{Reg}(\mathsf{FTL}, \ell^n), g(n)\} + 1. \tag{5}$$

As we mention before, the structure of the SMART algorithm (and the resulting guarantee) parallels the standard 2-competitive guarantee for the *ski-rental* problem [Karlin et al., 1994]. This is a classical optimal stopping problem, where a principal faces an unknown horizon, and in each period must decide whether to rent a pair of skis for the period (for fixed cost \$1) or buy the skis for the remaining horizon (for fixed cost \$$B$). The aim is to design a policy which is minimax optimal (over the unknown horizon) with regards to the ratio of the cost paid by the principal, and the optimal cost in hindsight. We further expand on this connection in Section 2.1 for the case of binary prediction. However, the connection suggests a natural follow-up question of whether randomized switching can help (as is the case for ski-rental); the following result answers this in the affirmative.

Theorem 2 (Regret of SMART with Randomized Thresholds)**.** *We are given* FTL *and any other policy* $\mathsf{ALG_{WC}}$ *with worst-case regret* $\sup_{\ell^n} \mathrm{Reg}(\mathsf{ALG_{WC}}, \ell^n) \leq g(n)$ *for some monotone function* g*. Moreover, given random sample* $U \sim \mathit{Unif}[0, 1]$*, suppose we set*

$$\theta = g(n)\ln(1 + (e-1)U)$$

Then playing the SMART *policy (Algorithm 1) with random threshold* θ *ensures*

$$\mathbb{E}_\theta\left[\mathrm{Reg}(\mathsf{SMART}, \ell^n)\right] \leq \frac{e}{e-1}\min\{\mathrm{Reg}(\mathsf{FTL}, \ell^n), g(n)\} + 1.$$

While we state the above as a randomized switching policy, this is more for interpretability – it is easier to view our policy as switching between two black-box algorithms rather than playing a convex combination of the two. However, since we define the loss incurred by any ALG in round t to be the expected loss when ALG plays a distribution w_t over actions $\mathcal{A}$ (see Section 1.1), therefore we can alternately implement the above by mixing between the actions of FTL and $\mathsf{ALG}_{\mathsf{WC}}$. More specifically, the above policy induces a *monotone* mixing rule, where over t, the weight on the action suggested by $\mathsf{ALG}_{\mathsf{WC}}$ is non-decreasing.

Remark 1 (Optimality over Monotone Mixing Policies)**.** *The competitive ratio of $\frac{e}{e-1}$ is known to be optimal for the ski-rental problem via Yao's minmax theorem [Borodin and El-Yaniv, 2005]. A direct corollary of this is the optimality of* SMART *over algorithms that are single switch (i.e., where the weight on $\mathsf{ALG}_{\mathsf{WC}}$ is non-decreasing in t). One difference between our setting and ski-rental is that switching back-and-forth between* FTL *and $\mathsf{ALG}_{\mathsf{WC}}$ is possible; see for example the* FlipFlop *policy of De Rooij et al. [2014]. In Section 3 we provide a fundamental lower bound of* 1.43 *on the competitive ratio over* all *algorithms; this suggests that multiple switching can help get a better competitive ratio (since $e/(e-1) \approx 1.58$), but also, that single-switch policies are surprisingly close to optimal.*

2.1 Illustrating the Reduction to Ski Rental in Binary Prediction

Before proving Theorems 1 and 2, we first illustrate the basic idea of our approach and reduction for the binary prediction setting. This is aided by the observation that the regret of FTL in this setting admits a simple geometric interpretation: for any sequence, and any time t, we have that Σ^{FTL} is equal to $1/2$ times the number of 'lead changes' up to time t; where a lead change is a time step i where the count of 1s and 0s in the (sub)sequence y^{i-1} is equal (see Figure 2(a)).

Corollary 2 (FTL for binary prediction)**.** *In binary prediction, for any sequence $y \in \{0,1\}^n$ and any time $t \le n$, define the lead-change counter*

$$c(y^t) := |\{0 \le j \le t \text{ s.t. } \sum_{i=1}^{j} y_i = \sum_{i=1}^{j} (1-y_i)\}|.$$

Then we have $\Sigma_t^{\mathsf{FTL}} = \frac{1}{2} c(y^{t-1})$ and thus $\mathrm{Reg}(\mathsf{FTL}, y^n) = \frac{1}{2} c(y^{n-1})$.

Corollary 2 follows from the regret decomposition in Lemma 1. Furthermore, since the losses of 0s and 1s are equal at a lead change, it also follows that at a lead change t, the anytime regret Σ_t^{FTL} is also equal to the hindsight regret incurred by FTL up to time t. Since Σ_t^{FTL} only increases in value at lead changes, if the SMART algorithm switches to $\mathsf{ALG}_{\mathsf{WC}}$, it will only happen after a lead change, and thus SMART behaves as if it had oracle knowledge of the regret of FTL from just the history up to the current time.

Consider the instantiation of SMART where $\mathsf{ALG}_{\mathsf{WC}} = \mathsf{Cover}$ and $g(n) = \sqrt{\frac{n}{2\pi}}$. As mentioned in Section 1, a remarkable property of Cover is that it is the true minimax optimal algorithm, where $\mathrm{Reg}(\mathsf{Cover}, y^n) = \sqrt{\frac{n}{2\pi}}(1+o(1))$ for all sequences y^n, such that $g(n)$ is nearly equal to $\mathrm{Reg}(\mathsf{Cover}, y^n)$ regardless of the sequence [Cover, 1966].

It follows that SMART is equivalent to an algorithm which starts with FTL, plays it until the regret of FTL matches the minimax regret guaranteed by Cover for the remaining sequence, and then switches to Cover until the end. Let t_{sw} denote the last round SMART plays FTL before switching to $\mathsf{ALG}_{\mathsf{WC}}$ (with $t_{\mathrm{sw}} = n$ if it never switches). For a single switch algorithm, the sequence which maximizes the regret is one that maximizes the FTL regret before the switch at t_{sw}, and minimizes

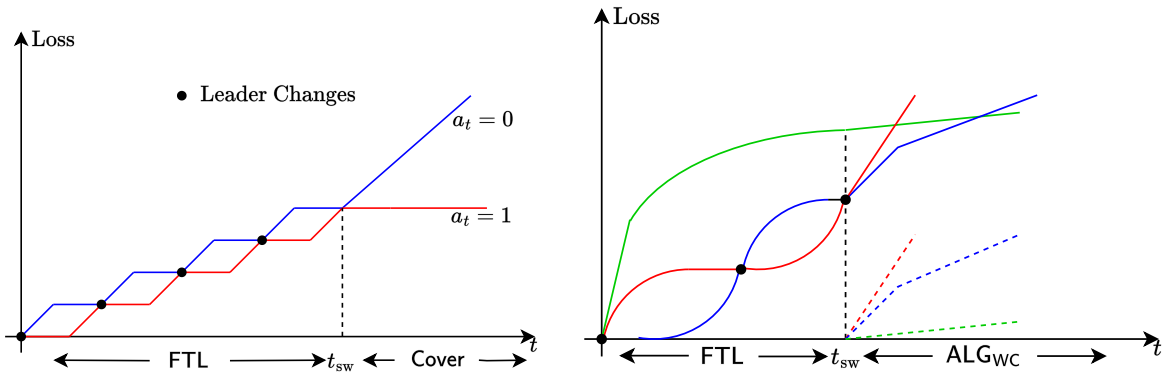

Figure 2: *Figure 2(a) on the left shows the worst case instance in binary prediction for an algorithm which starts with* FTL *and switches at most once during the time horizon to* Cover*. Figure2(b) on the right depicts in a prediction with experts setting how* SMART *resets the losses after the switch from* FTL *to* ALG$_{WC}$*.*

the FTL after the switch, as depicted in Figure 2(a). For such a sequence, $\Sigma_t^{\mathsf{FTL}} = t/4$ at lead changes t, such that the regret incurred by the algorithm is linear before the switch, matching the linear cost of renting skis in the ski rental problem. Note that t_{sw} will necessarily be $o(n)$ in such sequences as the time it takes until $\Sigma_t^{\mathsf{FTL}} \geq g(n)$ is linear in $g(n) = o(n)$. After the switch, Cover will incur regret $\sqrt{\frac{n-t_{\text{sw}}}{2\pi}}(1+o(1)) = g(n)(1+o(1))$, matching the fixed cost of buying skis at the switching point; Corollary 3 follows as a result of this analysis. Furthermore, in the binary prediction setting, SMART achieves the stronger notion of instance optimality stated in Definition 1.

Corollary 3. *For all $y^n \in \{0,1\}^n$,* SMART *with* ALG$_{WC}$ = Cover *and $\theta = \sqrt{\frac{n}{2\pi}}$ satisfies*

$$\text{Reg}(\mathsf{SMART}, y^n) \leq 2\min\{\text{Reg}(\mathsf{FTL}, y^n), \text{Reg}(\mathsf{Cover}, y^n)\} + 1.$$

SMART *with* ALG$_{WC}$ = Cover *and $\theta = \sqrt{\frac{n}{2\pi}}\ln(1+(e-1)U)$ for $U \sim$ Uniform$[0,1]$ satisfies*

$$\text{Reg}(\mathsf{SMART}, y^n) \leq 1.58\min\{\text{Reg}(\mathsf{FTL}, y^n), \text{Reg}(\mathsf{Cover}, y^n)\} + 1.$$

2.2 Proofs for General Online Learning

In the general online learning setting, the proof is nearly the same, with the reduction to the ski-rental problem captured by Lemma 2.

Lemma 2. *Let $t_{\text{sw}} := \min_{1\leq t\leq n-1} \Sigma_t^{\mathsf{FTL}} > \theta$ denote the last round* SMART *plays* FTL *before switching to* ALG$_{WC}$ *(with $t_{\text{sw}} = n$ if it never switches).* SMART *incurs regret bounded by*

$$\text{Reg}(\mathsf{SMART}, \ell^n) \leq \text{Reg}(\mathsf{FTL}, \ell^{t_{\text{sw}}}) + \text{Reg}(\mathsf{ALG}_{\mathsf{WC}}, \ell^n_{t_{\text{sw}}+1}) \leq \theta + \text{Reg}(\mathsf{ALG}_{\mathsf{WC}}, \ell^n_{t_{\text{sw}}+1}) + 1.$$

Proof. We separately bound the regret of SMART before the switch and after the switch,

$$\begin{aligned}
\text{Reg}(\mathsf{SMART}, \ell^n) &= \left(\textstyle\sum_{t=1}^{t_{\text{sw}}} \ell_t(a_t) - \sum_{t=1}^{t_{\text{sw}}} \ell_t(a_n^*)\right) + \left(\sum_{t=t_{\text{sw}}+1}^{n} \ell_t(a_t) - \sum_{t=t_{\text{sw}}+1}^{n} \ell_t(a_n^*)\right) \\
&\overset{(a)}{\leq} \left(\textstyle\sum_{t=1}^{t_{\text{sw}}} \ell_t(a_t) - \sum_{t=1}^{t_{\text{sw}}} \ell_t(a_{t_{\text{sw}}}^*)\right) + \left(\sum_{t=t_{\text{sw}}+1}^{n} \ell_t(a_t) - \sum_{t=t_{\text{sw}}+1}^{n} \ell_t(a_{t_{\text{sw}}+1:n}^*)\right) \\
&= \text{Reg}(\mathsf{FTL}, \ell^{t_{\text{sw}}}) + \text{Reg}(\mathsf{ALG}_{\mathsf{WC}}, \ell^n_{t_{\text{sw}}+1}).
\end{aligned}$$

The first term is upper bounded by $\mathrm{Reg}(\mathsf{FTL}, \ell^{t_{\mathrm{sw}}})$ since $L_{t_{\mathrm{sw}}}(a^*_n) \geq L_{t_{\mathrm{sw}}}(a^*_{t_{\mathrm{sw}}})$ by definition, as $a^*_{t_{\mathrm{sw}}}$ is the minimizer of $L_{t_{\mathrm{sw}}}$, and furthermore SMART always plays according to FTL in rounds up to t_{sw}. The second term is upper bounded by $\mathrm{Reg}(\mathsf{ALG_{WC}}, \ell^n_{t_{\mathrm{sw}}+1})$ because $\sum_{t=t_{\mathrm{sw}}+1}^{n} \ell_t(a^*_{t_{\mathrm{sw}}+1:n}) \leq \sum_{t=t_{\mathrm{sw}}+1}^{n} \ell_t(a^*_n)$ as $a^*_{t_{\mathrm{sw}}+1:n}$ is the minimizer of the losses after t_{sw}, and at time $t > t_{\mathrm{sw}}$, SMART plays according to $\mathsf{ALG_{WC}}$ on the sequence of losses limited to ℓ^n_{t+1}. This illustrates the important role of resetting the losses after the switch as depicted in Figure 2(b). Using the fact that $\Sigma^{\mathsf{FTL}}_{t_{\mathrm{sw}}-1} \leq \theta$ and $L_{t_{\mathrm{sw}}-1}(x^*_{t_{\mathrm{sw}}-1}) \leq L_{t_{\mathrm{sw}}-1}(x^*_{t_{\mathrm{sw}}})$, it follows that

$$\mathrm{Reg}(\mathsf{FTL}, \ell^{t_{\mathrm{sw}}}) = \Sigma^{\mathsf{FTL}}_{t_{\mathrm{sw}}-1} + L_{t_{\mathrm{sw}}-1}(a^*_{t_{\mathrm{sw}}-1}) - L_{t_{\mathrm{sw}}-1}(x^*_{t_{\mathrm{sw}}}) + \ell_{t_{\mathrm{sw}}}(a^*_{t_{\mathrm{sw}}-1}) - \ell_{t_{\mathrm{sw}}}(x^*_{t_{\mathrm{sw}}}) \leq \theta + 1. \quad \square$$

The reduction to ski-rental is again immediate due to the properties that $\Sigma^{\mathsf{FTL}}_t := \mathrm{Reg}(\mathsf{FTL}, \ell^{t_{\mathrm{sw}}})$ is adapted, monotone, and is an anytime lower bound for $\mathrm{Reg}(\mathsf{FTL}, \ell^n)$, while remaining an upper bound on the true regret incurred by FTL up to time t. As a result, the algorithm can pretend that it truly observes the regret it incurs at each time up to the switching time t_{sw}. After the switch, SMART incurs regret $\mathrm{Reg}(\mathsf{ALG_{WC}}, \ell^n_{t_{\mathrm{sw}}+1})$ which is upper bounded by $g(n - t_{\mathrm{sw}}) \leq g(n)$ by assumption.

Proof of Theorem 1. This follows immediately from Lemma 2. For ℓ^n such that $\mathrm{Reg}(\mathsf{FTL}, \ell^n) \leq g(n)$, SMART will never switch to Cover as $\Sigma^{\mathsf{FTL}}_t \leq \mathrm{Reg}(\mathsf{FTL}, \ell^n) \leq g(n)$ such that $\mathrm{Reg}(\mathsf{SMART}, \ell^n) = \min\{\mathrm{Reg}(\mathsf{FTL}, \ell^n), g(n)\}$. For ℓ^n such that $\mathrm{Reg}(\mathsf{FTL}, \ell^n) > g(n)$, by Lemma 2, $\mathrm{Reg}(\mathsf{SMART}, \ell^n) \leq g(n) + g(n - t_{\mathrm{sw}}) + 1 \leq 2g(n) + 1$.

$\square$

Proof of Theorem 2. The proof uses a primal-dual approach, similar to that of Karlin et al. [1994] for ski-rental. For a given sequence of loss functions ℓ^n, we use the shorthand $r = \mathrm{Reg}(\mathsf{FTL}, \ell^n)$ and $g = g(n)$. Also, for our given choice of cumulative distribution function F_n, the corresponding probability density function is given by $f(z) = \frac{e^{z/g}}{g(e-1)}$ for $z \in [0, g]$. As before, let $t_{\mathrm{sw}} := \min_{1 \leq t \leq n-1} \Sigma^{\mathsf{FTL}}_t > \theta$ be the (random) round where SMART switches from FTL to $\mathsf{ALG_{WC}}$ (with $t_{\mathrm{sw}} = n$ if it never switches). Then by Lemma 2 we have

$$\frac{\mathrm{Reg}(\mathsf{SMART}, \ell^n) - 1}{\min\{\mathrm{Reg}(\mathsf{FTL}, \ell^n), g(n)\}} \leq \begin{cases} \frac{\theta + g}{\min\{r, g\}} & \text{if } t_{\mathrm{sw}} < n \\ 1 & \text{if } t_{\mathrm{sw}} = n \end{cases}$$

where the second case follows from the fact that $\theta \in [0, g]$, and hence if we never switch, then $r \leq g(n)$. Taking expectation over θ, we have

$$\frac{\mathbb{E}_\theta\left[\mathrm{Reg}(\mathsf{SMART}, \ell^n)\right] - 1}{\min\{\mathrm{Reg}(\mathsf{FTL}, \ell^n), g\}} \leq \begin{cases} \int_0^r \frac{x+g}{r} f(x)dx + 1 - F(r) & \text{if } r \leq g \\ \int_0^g \frac{x+g}{g} f(x)dx & \text{if } r > g \end{cases}$$

Let $\phi(z) := \int_0^z \frac{(x+g)}{z} f(x)dx + 1 - F(z)$ for $z \in [0, g]$; then $\frac{\mathbb{E}_\theta\left[\mathrm{Reg}(\mathsf{SMART}, \ell^n)\right] - 1}{\min\{\mathrm{Reg}(\mathsf{FTL}, \ell^n), g\}} \leq \max_{z \in [0,g]} \phi(z)$. Moreover, we can differentiate to get $z^2\phi'(z) = gzf(z) - \int_0^z (x+g)f(x)dx$. Substituting our choice of f in this expression, we get

$$\frac{z^2 d\phi(z)}{dz} = \frac{ze^{z/g}}{(e-1)} - \int_0^z (x+g)\frac{e^{x/g}}{g(e-1)}dx = \frac{ze^{z/g} - g\int_0^{z/g}(w+1)e^w dw}{(e-1)} = 0$$

Thus, $\phi(z)$ is constant for all $z \in [0, g]$ and $\phi(g) = \frac{1}{g(e-1)} \int_0^g (1 + x/g)e^{x/g}dx = \frac{e}{e-1}$. $\square$

3 Instance Optimal Online Learning: Converse

In this section, we investigate fundamental limits on the instance-optimal regret guarantees achievable by *any* algorithm. More precisely, in the setting of binary prediction, we ask what is the smallest value of γ_n satisfying

$$\mathrm{Reg}(\mathsf{ALG}, y^n) \le \gamma_n \min\{\mathrm{Reg}(\mathsf{FTL}, y^n), \mathrm{Reg}(\mathsf{Cover}, y^n)\} = \gamma_n \min\left\{\tfrac{1}{2}c(y^{n-1}), f_n\right\} \tag{6}$$

for all y^n, where $f_n := \mathrm{Reg}(\mathsf{Cover}, y^n) = \sqrt{\frac{n}{2\pi}}(1+o(1))$. We show the following lower bound.

Theorem 3 (Lower bound on the competitive ratio)**.**

$$\lim_{n\to\infty} \gamma_n \ge \left(1 - e^{-1/\pi} + 2Q\big(\sqrt{2/\pi}\big)\right)^{-1} \approx 1.4335$$

where the $Q(\cdot)$ function is $Q(x) := \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2}$.

Since binary prediction is a specific online learning problem, this also yields a fundamental lower bound for instance-optimality for general online learning. Note particularly that $\gamma_n > 1$, implying that $(1+o(1))\min\{\mathrm{Reg}(\mathsf{FTL}), \mathrm{Reg}(\mathsf{ALG}_{\mathsf{WC}})\}$ regret is not possible to achieve. Thus, there is an inevitable multiplicative factor that must be paid in order to achieve an instance-optimal regret guarantee.

An equivalent way to state (6) is to find the smallest γ_n for which a predictor $\{a_t(y^{t-1})\}_{t=1}^n$ satisfies for all $y \in \{0,1\}^n$

$$\textstyle\sum_{t=1}^n |a_t(y^{t-1}) - y_t| \le \gamma_n \min\left\{\frac{1}{2}c(y^{n-1}), f_n\right\} + \min\left\{\sum_{t=1}^n y_i, n - \sum_{t=1}^n y_i\right\}. \tag{7}$$

In order to establish the values of γ_n for which the loss function in the right hand side of (7) are achievable, we utilize the following result of Cover [1966], which provides an exact characterization of the set of *all* loss functions achievable in binary prediction. Formally, we say a function $\phi : \{0,1\}^n \to \mathbb{R}^+$ is *achievable* in binary prediction if there exists a predictor/strategy $a_t : y^{t-1} \mapsto [0,1]$ that ensures $\sum_{t=1}^n |a_t(y^{t-1}) - y_t| = \phi(y^n) \quad , \forall\, y^n \in \{0,1\}^n$. Then, we have the following characterization.

Theorem 4 (Cover [1966])**.** *Let $\epsilon^n \sim \mathrm{Bern}\left(\frac{1}{2}\right)$ i.i.d. For ϕ to be achievable, it must satisfy the following:*

- *Balance:* $\mathbb{E}[\phi(\epsilon^n)] = \frac{n}{2}$.
- *Stability: Let $\phi_t(y^t) := \mathbb{E}[\phi(y^t \epsilon_{t+1}^n)]$; then $|\phi_t(y^{t-1}0) - \phi_t(y^{t-1}1)| \le 1\, \forall t \in [n]$, $y^t \in \{0,1\}^t$.*

Further any ϕ satisfying the above is realized by predictor $a_t(y^{t-1}) = \frac{1+\phi_t(y^{t-1}0)-\phi_t(y^{t-1}1)}{2}$.

As an immediate corollary, Theorem 4 equips us with the *exact* minimax optimal algorithm for binary prediction alluded to in Section 1. Returning to our setting, from the balance condition in Theorem 4, for $\epsilon^n \sim \mathrm{Bern}(1/2)$ i.i.d.

$$\gamma_n\, \mathbb{E}\left[\min\left\{\tfrac{1}{2}c(\epsilon^{n-1}), f_n\right\}\right] + \mathbb{E}\left[\min\left\{\textstyle\sum_{t=1}^n \epsilon_t, n - \sum_{t=1}^n \epsilon_t\right\}\right] \ge \tfrac{n}{2}.$$

for the function in (7) to be achievable. Using the definition of f_n,

$$\gamma_n \ge \frac{f_n}{\mathbb{E}\left[\min\left\{\frac{1}{2}c(\epsilon^{n-1}), f_n\right\}\right]} = \frac{2f_n}{\mathbb{E}\left[\min\left\{c(\epsilon^{n-1}), 2f_n\right\}\right]}.$$

The above bound immediately yields that $\gamma_n \ge 1$ as expected. We can further sharply characterize the asymptotics of γ_n, resulting in the stated lower bound. The full proof of Theorem 3 is provided in Appendix B.

4 Instance-Optimal Algorithms in Small-Loss Settings

So far, we have presented specializations of SMART that achieve instance-optimality between FTL and the *worst-case* regret $g(n)$. However, many worst-case algorithms can still adapt to the instance ℓ^n and achieve regret guarantees that are a function of the 'difficulty' of the instance ℓ^n. A common way to quantify this is difficulty is via *small-loss bounds*, where the regret is upper bounded by $g(L^*)$ where $g(\cdot)$ as earlier is a monotonic increasing function and $L^* := \min_{a\in\mathcal{A}} \sum_{t=1}^{n} \ell_t(a)$ is the loss achieved by the best action. Such guarantees imply that for sequences where there exists an action achieving low loss, the corresponding regret achieved is also low. Thus, a natural question is whether SMART can be specialized to yield an algorithm that is constant competitive with respect to $\min\{\mathrm{Reg}(\mathsf{FTL}, \ell^n), g(L^*)\}$.

As a starting point, if L^* is known apriori, it is easy to achieve a $\frac{e}{e-1}$ approximation by simply using SMART with (random) threshold $\theta = g(L^*)\ln(1 + (e-1)U), U \sim \mathrm{Unif}[0,1]$; this is an immediate corollary of Theorem 2. When L^* is not known, we use a guess-and-double argument to devise an algorithm that achieves the following instance-optimality guarantee.

Theorem 5 (Regret of SMART for unknown small loss)**.** *Let* $\mathsf{ALG}_{\mathsf{WC}}$ *have small loss regret guarantees satisfying* $\mathrm{Reg}(\mathsf{ALG}_{\mathsf{WC}}, \ell^n) \le g(L^*)$ *for any* ℓ^n *where* $L^* = \min_{j\in[m]} L_{tj}$*, i.e. the loss achieved by the best expert in hindsight. Then, if we play* SMART *for Small-Loss as stated in Algorithm* 2*, we have*

$$\mathrm{Reg}(\mathsf{SMART}, \ell^n) \le 2\min\left(\mathrm{Reg}(\mathsf{FTL}, \ell^n), \textstyle\sum_{z=1}^{\log(1+L^*/\log m)+1} g(2^z \log m)\right) + O(\log L^*/\log m)$$

In particular, in the prediction with expert advice setting, we know that Hedge with a time-varying learning rate achieves $g(L^*) \equiv 2\sqrt{2L^*\log m} + \kappa \log m$ (where $\kappa > 0$ is an absolute constant) [Auer et al., 2002b, Cesa-Bianchi et al., 2007]; this gives Corollary 1 in Section 1.

The intuitive idea behind the algorithm is to guess the value of L^*, and play SMART with this guessed value while simultaneously keeping track of the regret incurred. Whenever the regret incurred exceeds the guarantee established by SMART with known L^* double the guessed value and start again. We use the notation $\mathsf{ALG}_{\mathsf{WC}}(\ell_{t_1}^{t_2})$ to refer to the worst-case algorithm when the previously observed sequence is $\ell_{t_1}^{t_2}$; in particular this would be equivalent to the action recommended at time t_2+1 after throwing away all the observed losses before t_1. We let $\Sigma_{t_1:t_2}^{\mathsf{FTL}} = \sum_{i=t_1}^{t_2}(L_i(a_{i-1}^*) - L_i(a_i^*))$, which grows as the number of leader changes within $i \in [t_1, t_2]$. The algorithm's pseudocode is given in Algorithm 2 below, and a proof of Theorem 5 is provided in Appendix A.

5 Conclusion

In this paper, we present SMART, a simple and black-box online learning algorithm that adapts to the data and achieves instance optimal regret with respect to FTL and any given worst-case algorithm. We show that SMART only switches once from FTL to the worst-case algorithm, and attains a regret that is within a factor of $e/(e-1) \approx 1.58$ of the minimum of the regret of FTL and the minimax regret over all input sequences; we also show that any algorithm must incur an extra factor of at least 1.43 establishing that our simple approach is surprisingly close to optimal. Furthermore, we extend SMART to incorporate a small-loss algorithm and obtain instance optimality with respect to the small-loss regret bound. Our approach relies on a novel reduction of instance optimal online learning to the ski-rental problem, and leverages tools from information theory and competitive analysis. Our work suggests several open problems for future research, such as finding instance optimal algorithms for bandit settings, or designing algorithms that can adapt to multiple reference algorithms besides FTL and minimax algorithms.

Algorithm 2: SMART for Small-Loss

Input: Policies $\mathsf{FTL}, \mathsf{ALG}_{\mathsf{WC}}$; Small-loss bound $g(\cdot)$

for $z = 0, 1, \ldots$ *(epochs)* **do**

Let $t = t_z :=$start time of z^{th} epoch, $L_z^* := 2^z \log m$ (current guess for L^*), $\Sigma^{\mathsf{FTL}}_{t_z:t_z-1} = 0$;

while $\Sigma^{\mathsf{FTL}}_{t_z:t-1} \le g(L_z^*)$ **do**

Set $a_t = a^*_{t-1}$; `// Play FTL`

Observe $\ell_t(\cdot)$;

Update $L_t(\cdot) = L_{t-1}(\cdot) + \ell_t(\cdot)$ and $\Sigma^{\mathsf{FTL}}_{t_z:t} = \Sigma^{\mathsf{FTL}}_{t_z:t-1} + (L_t(a^*_{t-1}) - L_t(a^*_t))$ and $t = t+1$;

end

Let $\tau_z := \min_{t \ge t_z} \Sigma^{\mathsf{FTL}}_{t_z:t} > g(L_z^*)$ and $t = \tau_z + 1$;

`// Check if loss incurred by` $\mathsf{ALG}_{\mathsf{WC}}$ `in this epoch violates the upper bound from` L_z^* `is correct`

while $\sum_{t=t_z}^{t} \langle a_t, \ell_t \rangle \le L_z^* + 2\min\{\Sigma^{\mathsf{FTL}}_{t_z:t}, g(L_z^*)\} + 1$ **do**

Set $a_t = \mathsf{ALG}_{\mathsf{WC}}(\ell^{t-1}_{\tau_z+1})$; `// Play` $\mathsf{ALG}_{\mathsf{WC}}$ `forgetting losses before` $\tau_z + 1$

Observe $\ell_t(\cdot)$;

Update $L_t(\cdot) = L_{t-1}(\cdot) + \ell_t(\cdot)$ and $\Sigma^{\mathsf{FTL}}_{t_z:t} = \Sigma^{\mathsf{FTL}}_{t_z:t-1} + (L_t(a^*_{t-1}) - L_t(a^*_t))$ and $t = t+1$;

end

end

Acknowledgements

This work is supported by NSF grants CNS-1955997, CCF-2337796 and ECCS-1847393, and AFOSR grant FA9550-23-1-0301. This work was partially done when the authors were visitors at the Simons Institute for the Theory of Computing, UC Berkeley.

A Omitted proofs from Section 4

In this Section, we will establish the proofs of Theorem 5 and Corollary 1.

Recall Algorithm 2, where $\mathsf{ALG}_{\mathsf{WC}}(\ell^{t_2}_{t_1})$ refers to the worst-case algorithm when the previously observed sequence is $\ell^{t_2}_{t_1}$; in particular this would be equivalent to the action recommended at time t_2+1 after throwing away all the observed losses before t_1. We let $\Sigma^{\mathsf{FTL}}_{t_1:t_2} = \sum_{i=t_1}^{t_2}(L_i(a^*_{i-1}) - L_i(a^*_i))$, which grows as the number of leader changes within $i \in [t_1, t_2]$.

We first have the following decomposition of the regret for any algorithm ALG that plays the sequence of actions a_t^{ALG} at time t.

Lemma 3. *The regret incurred by any sequence of actions $(a_t^{\mathsf{ALG}})_{t\in[n+1]}$ can be written as*

$$\mathrm{Reg}(\mathsf{ALG}, \ell^n) = \sum_{t=1}^{n} \left(L_t(a_t^{\mathsf{ALG}}) - L_t(a_{t+1}^{\mathsf{ALG}}) \right), \tag{8}$$

where we let $a_{n+1}^{\mathsf{ALG}} := a_n^*$.

Proof.

$$L_n(\mathsf{ALG}) = \sum_{t=1}^{n} \ell_t(a_t^{\mathsf{ALG}}) \tag{9}$$

$$= \sum_{t=1}^{n} \left(L_t(a_t^{\mathsf{ALG}}) - L_{t-1}(a_t^{\mathsf{ALG}}) \right) \tag{10}$$

$$= L_n(a_n^{\mathsf{ALG}}) + \sum_{t=1}^{n-1} \left(L_t(a_t^{\mathsf{ALG}}) - L_t(a_{t+1}^{\mathsf{ALG}}) \right) - L_0(a_1^{\mathsf{ALG}}). \tag{11}$$

This implies a regret decomposition of

$$\mathrm{Reg}(\mathsf{ALG}, \ell^n) = L_n(\mathsf{ALG}) - L_n(a_n^*) \tag{12}$$

$$= L_n(a_n^{\mathsf{ALG}}) - L_n(a_n^*) + \sum_{t=1}^{n-1} \left(L_t(a_t^{\mathsf{ALG}}) - L_t(a_{t+1}^{\mathsf{ALG}}) \right) - L_0(a_1^{\mathsf{ALG}}) \tag{13}$$

As $L_0(a_1^{\mathsf{ALG}}) = 0$, and $a_{n+1}^{\mathsf{ALG}} := a_n^*$, it follows that

$$\mathrm{Reg}(\mathsf{ALG}, \ell^n) = \sum_{t=1}^{n} \left(L_t(a_t^{\mathsf{ALG}}) - L_t(a_{t+1}^{\mathsf{ALG}}) \right). \tag{14}$$

□

Next, we use this decomposition to establish the regret of any algorithm ALG that alternates between playing FTL and another algorithm $\mathsf{ALG}_{\mathsf{WC}}$.

Lemma 4. *Consider an algorithm* ALG *which alternates between playing* FTL *and* $\mathsf{ALG}_{\mathsf{WC}}$*, where* FTL *is played in the intervals* $\{[t_z, \tau_z]\}_{z \in [z_{\text{last}}]}$*, and* $\mathsf{ALG}_{\mathsf{WC}}$ *is played in intervals* $\{[\tau_z + 1, t_{z+1} - 1]\}_{z \in [z_{\text{last}}]}$*. The regret of* ALG *is bounded by*

$$\mathrm{Reg}(\mathsf{ALG}, \ell^n) \le \sum_z \left(\Sigma^{\mathsf{FTL}}_{t_z:\tau_z - 1} + \mathrm{Reg}(\mathsf{ALG}_{\mathsf{WC}}, \ell_{\tau_z+1}^{t_{z+1}-1}) + 1 \right). \tag{15}$$

Proof. We let $t_{z_{\text{last}}} = n + 1$, and $a_{n+1} = a_n^*$. We use Lemma 3 and rearrange the terms by grouping them by the FTL periods and the $\mathsf{ALG}_{\mathsf{WC}}$ periods.

$$\begin{aligned}
&\mathrm{Reg}(\mathsf{ALG}, \ell^n) \\
&= \sum_z \left(\sum_{t=t_z}^{t_{z+1}-1} (L_t(a_t) - L_t(a_{t+1})) \right) \\
&= \sum_z \left(\sum_{t=t_z}^{\tau_z - 1} (L_t(a_{t-1}^*) - L_t(a_t^*)) + L_{\tau_z}(a_{\tau_z - 1}^*) - L_{\tau_z}(a_{\tau_z+1}) + \sum_{t=\tau_z+1}^{t_{z+1}-1} (L_t(a_t) - L_t(a_{t+1})) \right) \\
&= \sum_z \left(\sum_{t=t_z}^{\tau_z - 1} (L_t(a_{t-1}^*) - L_t(a_t^*)) + L_{\tau_z}(a_{\tau_z - 1}^*) + \sum_{t=\tau_z+1}^{t_{z+1}-1} \ell_t(a_t) - L_{t_{z+1}-1}(a_{t_{z+1}}) \right) \\
&= \sum_z \sum_{t=t_z}^{\tau_z - 1} (L_t(a_{t-1}^*) - L_t(a_t^*)) + \sum_z \left(\sum_{t=\tau_z+1}^{t_{z+1}-1} \ell_t(a_t) + L_{\tau_z}(a_{\tau_z - 1}^*) - L_{t_{z+1}-1}(a_{t_{z+1}-1}^*) \right).
\end{aligned}$$

We bound the first term by the FTL regret. Recall the notation

$$\Sigma^{\mathsf{FTL}}_{t_z:\tau_z - 1} := \sum_{t=t_z}^{\tau_z - 1} (L_t(a_{t-1}^*) - L_t(a_t^*)). \tag{16}$$

Because we are playing FTL at both time τ_z and $\tau_z - 1$, it holds that

$$L_{\tau_z}(a^*_{\tau_z-1}) = L_{\tau_z-1}(a^*_{\tau_z-1}) + \ell_{\tau_z}(a^*_{\tau_z-1}) \le L_{\tau_z}(a^*_{\tau_z}) + 1.$$

To bound the second term, we will show that

$$\begin{aligned}
&\sum_{t=\tau_z+1}^{t_{z+1}-1} \ell_t(a_t) + L_{\tau_z}(a^*_{\tau_z-1}) - L_{t_{z+1}-1}(a^*_{t_{z+1}-1}) \\
&\qquad \le \sum_{t=\tau_z+1}^{t_{z+1}-1} \ell_t(a_t) + L_{\tau_z}(a^*_{\tau_z}) - L_{t_{z+1}-1}(a^*_{t_{z+1}-1}) + 1 && (17)\\
&\qquad = \sum_{t=\tau_z+1}^{t_{z+1}-1} \ell_t(a_t) + \min_a L_{\tau_z}(a) - \min_a L_{t_{z+1}-1}(a) + 1 && (18)\\
&\qquad \le \sum_{t=\tau_z+1}^{t_{z+1}-1} \ell_t(a_t) - \min_a \left(L_{t_{z+1}-1}(a) - L_{\tau_z}(a)\right) + 1 && (19)\\
&\qquad = \mathrm{Reg}(\mathsf{ALG}_{\mathsf{WC}}, \ell_{\tau_z+1}^{t_{z+1}-1}) + 1. && (20)
\end{aligned}$$

□

We now complete the proof of Theorem 3 by showing that the conditions that determine the switching time between FTL and $\mathsf{ALG}_{\mathsf{WC}}$ are appropriately chosen to upper bound $\Sigma^{\mathsf{FTL}}_{t_z:\tau_z-1}$ and $\mathrm{Reg}(\mathsf{ALG}_{\mathsf{WC}}, \ell_{\tau_z+1}^{t_{z+1}-1})$. Firstly, note that for any $z < z_{\text{last}}$, $\mathrm{Reg}(\mathsf{ALG}_{\mathsf{WC}}, \ell_{\tau_z+1}^{t_{z+1}-1}) > g(L^*_z)$, which implies that $L^* > L^*_z$. In particular, the epoch $z_{\text{last}} - 1$ was exited, which implies that[4]

$$2^{z_{\text{last}}-1} \log m \le L^* \implies z_{\text{last}} \le \log\left(\frac{L^*}{\log m}\right) + 1$$

By the stopping condition of the epoch, $\mathrm{Reg}(\mathsf{ALG}_{\mathsf{WC}}, \ell_{\tau_z+1}^{t_{z+1}-1}) \le \mathrm{Reg}(\mathsf{ALG}_{\mathsf{WC}}, \ell_{\tau_z+1}^{t_{z+1}-2}) + 1 \le g(L^*_z) + 1$, such that substituting into Lemma 4 implies

$$\mathrm{Reg}(\mathsf{SMART}, \ell^n) \le \sum_z \left(\Sigma^{\mathsf{FTL}}_{t_z:\tau_z-1} + g(L^*_z) + 2\right). \tag{21}$$

Also, it always holds that $\Sigma^{\mathsf{FTL}}_{t_z:\tau_z-1} \le g(L^*_z)$ and for $z < z_{\text{last}}$, $\Sigma^{\mathsf{FTL}}_{t_z:\tau_z} > g(L^*_z)$. Therefore

$$\sum_z^{z_{\text{last}}-1} g(L^*_z) < \sum_z \Sigma^{\mathsf{FTL}}_{t_z:\tau_z-1} \le \mathrm{Reg}(\mathsf{FTL}, \ell^n)$$

and $\sum_z \Sigma^{\mathsf{FTL}}_{t_z:\tau_z-1} \le \sum_z^{z_{\text{last}}} g(L^*_z)$.

To put it all together, if $\sum_z^{z_{\text{last}}} g(L^*_z) \le \mathrm{Reg}(\mathsf{FTL}, \ell^n)$, then

$$\mathrm{Reg}(\mathsf{SMART}, \ell^n) \le 2\sum_z g(L^*_z) + 2z_{\text{last}}. \tag{22}$$

[4]Here we have implicitly assumed that $L^* > \log m$—if not, then there is only one epoch, $z_{\text{last}} = 0$ and the result is readily implied by Corollary 2.

If $\mathrm{Reg}(\mathsf{FTL}, \ell^n) < \sum_z^{z_{\text{last}}} g(L_z^*)$, then it must be that in the last epoch the algorithm never switches to $\mathsf{ALG}_{\mathsf{WC}}$. If it switched to $\mathsf{ALG}_{\mathsf{WC}}$ it would imply that $\mathrm{Reg}(\mathsf{FTL}, \ell^n) \geq \sum_z \Sigma_{t_z:\tau_z}^{\mathsf{FTL}} > \sum_z^{z_{\text{last}}} g(L_z^*)$ which would violate the assumption that $\mathrm{Reg}(\mathsf{FTL}, \ell^n) < \sum_z^{z_{\text{last}}} g(L_z^*)$. Therefore it must be that

$$\mathrm{Reg}(\mathsf{SMART}, \ell^n) \leq 2\mathrm{Reg}(\mathsf{FTL}, \ell^n) + 2z_{\text{last}}. \tag{23}$$

As a result, it follows that (putting the $L^* \leq \log m$ and the $L^* > \log m$ cases together)

$$\begin{aligned}
&\mathrm{Reg}(\mathsf{SMART}, \ell^n) \\
&\leq 2\min\left(\mathrm{Reg}(\mathsf{FTL}, \ell^n), \sum_{z=0}^{\log\left(1+\frac{L^*}{\log m}\right)+1} g(2^z \log m)\right) + 2\log\left(1+\frac{L^*}{\log m}\right) + 2.
\end{aligned} \tag{24}$$

A.1 Proof of Corollary 1

This follows from Theorem 5 and by calculating

$$\begin{aligned}
&\sum_{z=0}^{\log\left(1+\frac{L^*}{\log m}\right)+1} g(2^z \log m) \\
&= 2\sqrt{2}\log m \sum_{z=0}^{\log\left(1+\frac{L^*}{\log m}\right)+1} 2^{z/2} + \kappa \log m \log\left(1+\frac{L^*}{\log m}\right) + \kappa \log m \\
&\leq \log m \frac{4\sqrt{2}}{\sqrt{2}-1}\sqrt{1+\frac{L^*}{\log m}} + \kappa \log m \log\left(1+\frac{L^*}{\log m}\right) + \kappa \log m \\
&\leq 10\sqrt{2\log^2 m + 2L^* \log m} + \kappa \log m \log\left(1+\frac{L^*}{\log m}\right) + \kappa \log m \\
&\overset{(a)}{\leq} 10\sqrt{2L^* \log m} + \kappa \log\left(1+\frac{L^*}{\log m}\right)\log m + 10\sqrt{2}\log m + \kappa \log m
\end{aligned}$$

where (a) follows since for nonnegative a, b $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$.

B Proof of Theorem 3

Consider a large even n. We then have for horizon size $n+1$, $\mathrm{Reg}(\mathsf{FTL}, y^{n+1}) = \frac{c(y^n)}{2}$. Moreover, $\mathrm{Reg}(\mathsf{Cover}, y^{n+1}) = f_{n+1}$. Let[5]

$$p_{n,k} := \mathbb{P}[c(\epsilon^n) = k+1]. \tag{25}$$

We then have,

$$\begin{aligned}
\mathbb{E}[\min\{c(\epsilon^n), 2f_{n+1}\}] &= \sum_{k=1}^{n} \min\{k, 2f_{n+1}\} \Pr[c(\epsilon^n) = k] \\
&= \sum_{k=1}^{n} \min\{k, 2f_{n+1}\} p_{n,k-1}
\end{aligned}$$

[5]Note that k in this definition does not counting the origin as as a line crossing.

$$
\begin{aligned}
&= \sum_{k=0}^{n} \min\{k+1, 2f_{n+1}\} p_{n,k} \\
&= \sum_{k=0}^{\lfloor 2f_{n+1} \rfloor - 1} (k+1) p_{n,k} + 2f_{n+1} \mathbb{P}[c(\epsilon^n) \geq 2f_{n+1}] \\
&= \sum_{k=0}^{\lfloor 2f_{n+1} \rfloor - 1} (k+1) p_{n,k} + 2f_{n+1} \mathbb{P}[c(\epsilon^n) \geq 2f_{n+1}] + \mathbb{P}[c(\epsilon^n) \leq \lfloor 2f_{n+1} \rfloor] \\
&= \sum_{k=0}^{\lfloor 2f_{n+1} \rfloor - 1} k p_{n,k} + 2f_{n+1} \left(1 - \sum_{k=0}^{\lfloor 2f_{n+1} \rfloor - 1} p_{n,k} \right) + \mathbb{P}[c(\epsilon^n) \leq \lfloor 2f_{n+1} \rfloor] \qquad (26)
\end{aligned}
$$

Upon dividing (26) by $2f_{n+1}$, we get

$$
\frac{\mathbb{E}[\min\{c(\epsilon^n), 2f_{n+1}\}]}{2f_{n+1}} = \frac{\sum_{k=0}^{\lfloor 2f_{n+1} \rfloor - 1} k p_{n,k}}{2f_{n+1}} + \left(1 - \sum_{k=0}^{\lfloor 2f_{n+1} \rfloor - 1} p_{n,k} \right) + \frac{\mathbb{P}[c(\epsilon^n) \leq \lfloor 2f_{n+1} \rfloor]}{2f_{n+1}}. \qquad (27)
$$

Note that the third term vanishes since $f_{n+1} \to \infty$ and $0 \leq \mathbb{P}[\cdot] \leq 1$. We will now separately evaluate the first two terms in (27). To do this, we require an auxiliary lemma, the proof of which is provided later.

Lemma 5. *If $k \leq C\sqrt{n}$ for an absolute constant C, then for large enough n ($n \geq 32C^2$ suffices) we have*

$$
e^{-\frac{16C^3}{\sqrt{n}}} \leq \frac{p_{n,k}}{\sqrt{\frac{2}{n\pi}} e^{-k^2/2n}} \leq \sqrt{\frac{1 - C/\sqrt{n}}{1 - 2C/\sqrt{n}}} e^{\frac{16C^3}{\sqrt{n}}}. \qquad (28)
$$

That is,

$$
p_{n,k} = \sqrt{\frac{2}{n\pi}} e^{-k^2/2n} (1 + o(1)).
$$

We now evaluate the first term in (27). Since $\lfloor 2f_{n+1} \rfloor - 1 \leq 2\sqrt{n}$, we invoke Lemma 5 to evaluate

$$
\begin{aligned}
\sum_{k=0}^{\lfloor 2f_{n+1} \rfloor - 1} k p_{n,k} &= (1 + o(1)) \sum_{k=0}^{\lfloor 2f_{n+1} \rfloor - 1} k \sqrt{\frac{2}{n\pi}} e^{-\frac{k^2}{2n}} \qquad (29) \\
&= (1 + o(1)) \sqrt{\frac{2}{n\pi}} \sum_{k=0}^{\lfloor 2f_{n+1} \rfloor - 1} k e^{-\frac{k^2}{2n}} \qquad (30)
\end{aligned}
$$

Now, we note that since $x \mapsto x e^{-\frac{x^2}{2n}}$ is increasing on $(0, \lfloor 2f_{n+1} \rfloor - 1)$, by a Riemann approximation, we have that

$$
\int_0^{2f_{n+1}-2} x e^{-\frac{x^2}{2n}} dx - 1 \leq \sum_{k=0}^{\lfloor 2f_{n+1} \rfloor - 1} k e^{-\frac{k^2}{2n}} \leq \int_0^{2f_{n+1}} x e^{-\frac{x^2}{2n}} dx \qquad (31)
$$

and evaluating

$$
\int_0^{2f_{n+1}} x e^{-\frac{x^2}{2n}} dx = \frac{1}{2} \int_0^{4f_{n+1}^2} e^{-\frac{t}{2n}} dt
$$

$$= n\left(1 - e^{-\frac{4f_{n+1}^2}{2n}}\right)$$
$$= n\left(1 - e^{-\frac{1}{\pi}(1+o(1))}\right). \tag{32}$$

Evaluating the lower bound in (31) analogously we have

$$\sum_{k=0}^{\lfloor 2f_{n+1}\rfloor - 1} k e^{-\frac{k^2}{2n}} = (1+o(1))n\left(1 - e^{-\frac{1}{\pi}(1+o(1))}\right) \tag{33}$$

and therefore from (30),

$$\frac{\sum_{k=0}^{\lfloor 2f_{n+1}\rfloor - 1} k p_{n,k}}{2f_{n+1}} = (1+o(1))\frac{\left(1 - e^{-\frac{1}{\pi}(1+o(1))}\right) n\sqrt{\frac{2}{n\pi}}}{\sqrt{\frac{2(n+1)}{\pi}}}$$
$$= (1+o(1))\left(1 - e^{-\frac{1}{\pi}(1+o(1))}\right) \tag{34}$$

and therefore, from (34), we have that

$$\lim_{n\to\infty} \frac{\sum_{k=0}^{\lfloor 2f_{n+1}\rfloor - 1} k p_{n,k}}{2f_{n+1}} = 1 - e^{-\frac{1}{\pi}}. \tag{35}$$

We now address the second term in (27) by invoking Lemma 5 and noting that

$$\sum_{k=0}^{\lfloor 2f_{n+1}\rfloor - 1} p_{n,k} = (1+o(1))\sqrt{\frac{2}{n\pi}} \sum_{k=0}^{\lfloor 2f_{n+1}\rfloor - 1} e^{-\frac{k^2}{2n}}$$
$$\overset{(a)}{=} (1+o(1)) \int_0^{2f_{n+1}} e^{-\frac{x^2}{2n}} dx$$
$$= (1+o(1))\sqrt{\frac{2}{\pi}} \int_0^{\sqrt{\frac{2}{\pi}}} e^{-\frac{t^2}{2}} dt$$
$$= \frac{1}{\sqrt{2\pi}} \int_{-\sqrt{\frac{2}{\pi}}}^{\sqrt{\frac{2}{\pi}}} e^{-\frac{t^2}{2}} dt$$
$$= \mathbb{P}\left(-\sqrt{\frac{2}{\pi}} \le X \le \sqrt{\frac{2}{\pi}}\right) \tag{36}$$

where in (36) $X \sim \mathcal{N}(0,1)$. Then,

$$1 - \sum_{k=0}^{\lfloor 2f_{n+1}\rfloor - 1} p_{n,k} \to 1 - \mathbb{P}\left(-\sqrt{\frac{2}{\pi}} \le X \le \sqrt{\frac{2}{\pi}}\right)$$
$$= 2Q\left(\sqrt{\frac{2}{\pi}}\right). \tag{37}$$

Substituting (35) and (37) in (27) yields that

$$\frac{1}{\gamma_n} = \frac{\mathbb{E}[\min\{c(\epsilon^n), 2f_{n+1}\}]}{2f_{n+1}} \to 1 - e^{-\frac{1}{\pi}} + 2Q\left(\sqrt{\frac{2}{\pi}}\right) \tag{38}$$

and therefore

$$\gamma_n \to \frac{1}{1 - e^{-\frac{1}{\pi}} + 2Q\left(\sqrt{\frac{2}{\pi}}\right)}. \tag{39}$$

B.1 Proof of Lemma 5

We first note the following.

Proposition 1 (Feller, Chapter 3, Exercise 11)**.**

$$p_{n,k} = \frac{1}{2^{n-k}}\binom{n-k}{n/2}.$$

Therefore,

$$\frac{p_{n,k}2^n}{2^k} = \binom{n-k}{n/2} = \frac{(n-k)!}{\frac{n}{2}!\left(\frac{n}{2}-k\right)!}.$$

We now use the Stirling approximation:

$$\sqrt{2\pi m}\left(\frac{m}{e}\right)^m e^{\frac{1}{12m+1}} \le m! \le \sqrt{2\pi m}\left(\frac{m}{e}\right)^m e^{\frac{1}{12m}}. \tag{40}$$

Using (40) we have

$$\begin{aligned}
\frac{p_{n,k}2^n}{2^k} &\le \frac{\sqrt{2\pi(n-k)}}{\sqrt{2\pi(n/2)}\sqrt{2\pi(n/2-k)}} \cdot \frac{(n-k)^{n-k}}{(n/2)^{n/2}(n/2-k)^{n/2-k}} \\
&\qquad \cdot \exp\left(\frac{1}{12(n-k)} - \frac{1}{6n+1} - \frac{1}{6n-12k+1}\right) \\
&= \sqrt{\frac{2}{n\pi}}\sqrt{\frac{n-k}{n-2k}}\frac{(n-k)^{n-k}2^{n-k}}{n^{n/2}(n-2k)^{n/2-k}} \cdot \exp\left(\frac{1}{12n-k}\right) \\
&\overset{(a)}{\le} \sqrt{\frac{2}{n\pi}} \cdot 2^{n-k} \cdot \frac{(n-k)^{n-k}}{n^{n/2}(n-2k)^{n/2-k}} \exp\left(\frac{1}{12n-k}\right) \cdot \sqrt{\frac{1-C/\sqrt{n}}{1-2C/\sqrt{n}}} \\
&= \sqrt{\frac{2}{n\pi}} \cdot 2^{n-k} \underbrace{\frac{(1-k/n)^{n-k}}{(1-2k/n)^{n/2-k}}}_{=:T} \exp\left(\frac{1}{12n-k}\right) \cdot \sqrt{\frac{1-C/\sqrt{n}}{1-2C/\sqrt{n}}}.
\end{aligned} \tag{41}$$

We now analyze the term T in (41) in more detail.

Proposition 2.

$$-\frac{k^2}{2n^2} - c\frac{k^3}{n^2} \le \ln T \le -\frac{k^2}{2n^2} + c\frac{k^3}{n^2} \tag{42}$$

where $c \le 15$.

Proof. By Taylor theorem, we have

$$\ln(1-x) = -x - \frac{x^2}{2} - \frac{x^3}{3(1-\mu)^2} \text{ for } \mu \in (0,x). \tag{43}$$

Therefore, for $k \leq C\sqrt{n}$ and $n \geq 16C^2$ we have

$$\ln\left(1-\frac{k}{n}\right) = -\frac{k}{n} - \frac{k^2}{2n^2} - \frac{\alpha_1 k^3}{n^3} \tag{44}$$

$$\ln\left(1-\frac{2k}{n}\right) = -\frac{2k}{n} - \frac{2k^2}{2n^2} - \frac{\alpha_2 k^3}{n^3} \tag{45}$$

for $\alpha_1, \alpha_2 \in \left(\frac{1}{3}, \frac{8}{3}\right]$. Evaluating $\ln T$ we have

$$\begin{aligned}
\ln T &= (n-k)\ln\left(1-\frac{k}{n}\right) - \left(n-\frac{k}{2}\right)\ln\left(1-\frac{2k}{n}\right) \\
&\overset{(a)}{=} (n-k)\left(-\frac{k}{n} - \frac{k^2}{2n^2} - \frac{\alpha_1 k^3}{n^3}\right) - \left(n-\frac{k}{2}\right)\left(-\frac{2k}{n} - \frac{2k^2}{2n^2} - \frac{\alpha_2 k^3}{n^3}\right) \\
&= \left(-\frac{k^2}{2n} + \frac{k^2}{n} + \frac{k^2}{n} - \frac{2k^2}{n}\right) + \left(-\alpha_1 + \frac{1}{2} + \frac{\alpha_2}{2} - 2\right)\frac{k^3}{n^2} + (\alpha_1 - \alpha_2)\frac{k^4}{n^3} \\
&= -\frac{k^2}{2n^2} + c_1\frac{k^3}{n^2} + c_2\frac{k^4}{n^3}
\end{aligned} \tag{46}$$

where (a) follows by substituting (44) and (45). Now, since $\frac{k^4}{n^3} \leq \frac{k^3}{n^2}$ we have

$$\ln T \leq -\frac{k^2}{2n^2} + (|c_1| + |c_2|)\frac{k^3}{n^2} \tag{47}$$

and on the other hand for the same reason

$$\ln T \geq -\frac{k^2}{2n} - (|c_1| + |c_2|)\frac{k^3}{n^2} \tag{48}$$

The proposition follows by noticing $|c_1| + |c_2| \leq 15$ by using that $\alpha_1, \alpha_2 \in \left(\frac{1}{3}, \frac{8}{3}\right]$. □

Using Proposition 2 in (41) we have the upper bound

$$\begin{aligned}
p_{n,k} &\leq \sqrt{\frac{2}{n\pi}} \cdot \exp\left(-\frac{k^2}{2n} + \frac{15k^3}{n^2}\right) \cdot \exp\left(\frac{1}{12n-k}\right) \cdot \sqrt{\frac{1-C/\sqrt{n}}{1-2C/\sqrt{n}}} \\
&\overset{(a)}{\leq} \sqrt{\frac{2}{n\pi}} \cdot \exp\left(-\frac{k^2}{2n}\right) \cdot \exp\left(\frac{16k^3}{n^2}\right) \cdot \sqrt{\frac{1-C/\sqrt{n}}{1-2C/\sqrt{n}}} \\
&\overset{(b)}{\leq} \sqrt{\frac{2}{n\pi}} \cdot \exp\left(-\frac{k^2}{2n}\right) \cdot \exp\left(\frac{16C^3}{\sqrt{n}}\right) \cdot \sqrt{\frac{1-C/\sqrt{n}}{1-2C/\sqrt{n}}}
\end{aligned} \tag{49}$$

where (a) uses $\frac{1}{12n-k} + \frac{15k^3}{n^2} \leq \frac{16k^3}{n^2}$, and (b) uses the fact that $k \leq C\sqrt{n}$, which yields the upper bound in (28). The lower bound follows analogously by the Stirling approximation (40) and Proposition 2.

References

Ronald Fagin, Amnon Lotem, and Moni Naor. Optimal aggregation algorithms for middleware. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 102–113, 2001.

Tim Roughgarden. *Beyond the worst-case analysis of algorithms*. Cambridge University Press, 2021.

D Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, 1956.

James Hannan. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Aleksandrs Slivkins. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

Thomas M. Cover. Behavior of sequential predictors of binary sequences. In *Transactions of the Fourth Prague Conference on Information Theory*, 1966.

Ruitong Huang, Tor Lattimore, András György, and Csaba Szepesvári. Following the leader and fast rates in linear prediction: Curved constraint sets and other regularities. *Advances in Neural Information Processing Systems*, 29, 2016.

Meir Feder, Neri Merhav, and Michael Gutman. Universal prediction of individual sequences. *IEEE transactions on Information Theory*, 38(4):1258–1270, 1992.

Alekh Agarwal, Haipeng Luo, Behnam Neyshabur, and Robert E Schapire. Corralling a band of bandit algorithms. In *Conference on Learning Theory*, pages 12–38. PMLR, 2017.

Aldo Pacchiano, My Phan, Yasin Abbasi Yadkori, Anup Rao, Julian Zimmert, Tor Lattimore, and Csaba Szepesvari. Model selection in contextual stochastic bandit problems. *Advances in Neural Information Processing Systems*, 33:10328–10337, 2020.

Christoph Dann, Chen-Yu Wei, and Julian Zimmert. Best of both worlds policy optimization. *arXiv preprint arXiv:2302.09408*, 2023.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002a.

Steven De Rooij, Tim Van Erven, Peter D Grünwald, and Wouter M Koolen. Follow the leader if you can, hedge if you must. *The Journal of Machine Learning Research*, 15(1):1281–1316, 2014.

Francesco Orabona and Dávid Pál. Scale-free algorithms for online linear optimization. In *International Conference on Algorithmic Learning Theory*, pages 287–301. Springer, 2015.

Jaouad Mourtada and Stéphane Gaïffas. On the optimality of the hedge algorithm in the stochastic regime. *Journal of Machine Learning Research*, 20:1–28, 2019.

Blair Bilodeau, Jeffrey Negrea, and Daniel M Roy. Relaxing the iid assumption: Adaptively minimax optimal regret via root-entropic regularization. *The Annals of Statistics*, 51(4):1850–1876, 2023.

Sébastien Bubeck and Aleksandrs Slivkins. The best of both worlds: Stochastic and adversarial bandits. In *Conference on Learning Theory*, pages 42–1. JMLR Workshop and Conference Proceedings, 2012.

Julian Zimmert and Yevgeny Seldin. An optimal algorithm for stochastic and adversarial bandits. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 467–475. PMLR, 2019.

Thodoris Lykouris, Vahab Mirrokni, and Renato Paes Leme. Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 114–122, 2018.

Wojciech Kotłowski. On minimaxity of follow the leader strategy in the stochastic setting. *Theoretical Computer Science*, 742:50–65, 2018.

Anna R. Karlin, Mark S. Manasse, Lyle A. McGeoch, and Susan Owicki. Competitive randomized algorithms for nonuniform problems. *Algorithmica*, 11(6):542–571, 1994.

Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis*. cambridge university press, 2005.

Nicolo Cesa-Bianchi, Yoav Freund, David Haussler, David P Helmbold, Robert E Schapire, and Manfred K Warmuth. How to use expert advice. *Journal of the ACM (JACM)*, 44(3):427–485, 1997.

Chen-Yu Wei and Haipeng Luo. More adaptive algorithms for adversarial bandits. In *Conference On Learning Theory*, pages 1263–1291. PMLR, 2018.

Sébastien Bubeck, Yuanzhi Li, Haipeng Luo, and Chen-Yu Wei. Improved path-length regret bounds for bandits. In *Conference On Learning Theory*, pages 508–528. PMLR, 2019.

Neelkamal Bhuyan, Debankur Mukherjee, and Adam Wierman. Best of both worlds: Stochastic and adversarial convex function chasing. *arXiv preprint arXiv:2311.00181*, 2023.

Oron Sabag, Gautam Goel, Sahin Lale, and Babak Hassibi. Regret-optimal controller for the full-information problem. In *2021 American Control Conference (ACC)*, pages 4777–4782. IEEE, 2021.

Gautam Goel, Naman Agarwal, Karan Singh, and Elad Hazan. Best of both worlds in online control: Competitive ratio and policy regret. In *Learning for Dynamics and Control Conference*, pages 1345–1356. PMLR, 2023.

Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning: Stochastic, constrained, and smoothed adversaries. *Advances in neural information processing systems*, 24, 2011.

Nika Haghtalab, Tim Roughgarden, and Abhishek Shetty. Smoothed analysis with adaptive adversaries. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 942–953. IEEE, 2022.

Adam Block, Yuval Dagan, Noah Golowich, and Alexander Rakhlin. Smoothed online learning is as easy as statistical learning. In *Conference on Learning Theory*, pages 1716–1786. PMLR, 2022.

Alankrita Bhatt, Nika Haghtalab, and Abhishek Shetty. Smoothed analysis of sequential probability assignment. *Neural Information Processing Systems*, 2023.

Idan Amir, Idan Attias, Tomer Koren, Yishay Mansour, and Roi Livni. Prediction with corrupted expert advice. *Advances in Neural Information Processing Systems*, 33:14315–14325, 2020.

Peter Auer, Nicolo Cesa-Bianchi, and Claudio Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1):48–75, 2002b.

Nicolo Cesa-Bianchi, Gábor Lugosi, and Gilles Stoltz. Minimizing regret with label efficient prediction. *IEEE Transactions on Information Theory*, 51(6):2152–2162, 2005.

Elad Hazan and Satyen Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. *Machine learning*, 80:165–188, 2010.

Wouter M Koolen, Tim Van Erven, and Peter Grünwald. Learning the learning rate for prediction with expert advice. *Advances in neural information processing systems*, 27, 2014.

Tim Van Erven, Peter Grunwald, Nishant A Mehta, Mark Reid, Robert Williamson, et al. Fast rates in statistical and online learning. 2015.

Tim Van Erven and Wouter M Koolen. Metagrad: Multiple learning rates in online learning. *Advances in Neural Information Processing Systems*, 29, 2016.

Pierre Gaillard, Gilles Stoltz, and Tim Van Erven. A second-order bound with excess losses. In *Conference on Learning Theory*, pages 176–196. PMLR, 2014.

Etienne Bamas, Andreas Maggiori, and Ola Svensson. The primal-dual method for learning augmented algorithms. *Advances in Neural Information Processing Systems*, 33:20083–20094, 2020.

Michael Dinitz, Sungjin Im, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. Algorithms with prediction portfolios. *Advances in neural information processing systems*, 35: 20273–20286, 2022.

Keerti Anand, Rong Ge, Amit Kumar, and Debmalya Panigrahi. Online algorithms with multiple predictions. In *International Conference on Machine Learning*, pages 582–598. PMLR, 2022.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Nicolo Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66:321–352, 2007.

William Feller. *An introduction to probability theory and its applications, Volume 1, Third Edition*. John Wiley & Sons, New York.