# Efficient First-Order Contextual Bandits:
## Prediction, Allocation, and Triangular Discrimination

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

A recurring theme in statistical learning, online learning, and beyond is that faster convergence rates are possible for problems with low noise, often quantified by the performance of the best hypothesis; such results are known as *first-order* or *small-loss* guarantees. While first-order guarantees are relatively well-understood in statistical and online learning, adapting to low noise in *contextual bandits* (and more broadly, decision making) presents major algorithmic challenges. In a COLT 2017 open problem, Agarwal et al. [5] asked whether first-order guarantees are even possible for contextual bandits and—if so—whether they can be attained by efficient algorithms. We resolve this question by providing an optimal and efficient reduction from contextual bandits to online regression with the logarithmic (or, cross-entropy) loss. Our algorithm is simple and practical, readily accommodates rich function classes, and requires no distributional assumptions beyond realizability. In a large-scale empirical evaluation, we find our approach typically outperforms comparable non-first-order methods.

On the technical side, we show that the logarithmic loss and an information-theoretic quantity called the *triangular discrimination* play a fundamental role in obtaining first-order guarantees, and we combine this observation with new refinements to the regression oracle reduction framework of Foster and Rakhlin [29]. The use of the triangular discrimination yields novel results even for the classical statistical learning model, and we anticipate that it will find broader use.

## 1 Introduction

In the contextual bandit problem, a learning agent repeatedly makes decisions based on contextual information, with the goal of learning a decision-making policy that minimizes their total loss over time. This model captures simple reinforcement learning tasks in which the agent must learn to make high-quality decisions in an uncertain environment, but need not engage in long-term planning and credit assignment. Owing to the availability of high-quality engineered reward metrics, contextual bandit algorithms are now routinely deployed in production for online personalization systems [4, 61].

Contextual bandits embed both the general problem of statistical learning with function approximation (specifically, cost-sensitive classification) and the classical multi-armed bandit problem, yet present algorithmic challenges greater than the sum of both parts. In spite of these difficulties, extensive research effort over the past decade has resulted in efficient, general-purpose algorithms, as well as a tight understanding of the optimal worst-case sample complexity [9, 12, 3, 29, 58].

While the algorithmic and statistical foundations for contextual bandits are beginning to take shape, we still lack an understanding of *adaptive* or *data-dependent* algorithms that can go *beyond the worst case* and exploit nice properties of real-world instances for better performance. This is in sharp contrast with supervised statistical learning, where adaptivity is well understood, and where

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

standard algorithms (e.g., empirical risk minimization) are known to automatically adapt to nice data [17]. For contextual bandits, adaptivity poses new challenges that seem to require algorithmic innovation, and a major frontier for research is to develop both algorithmic principles for adaptivity and an understanding of the fundamental limits.

To highlight the lack of understanding for adaptive and data-dependent algorithms, a COLT 2017 open problem posed by Agarwal, Krishnamurthy, Langford, and Luo [5] asks whether there exist contextual bandit algorithms that achieve a certain data-dependent *first-order* regret bound, which scales with the cumulative loss $L^\star$ of the best policy, rather than with the time horizon $T$. First-order regret bounds (also known as *small-loss bounds* or *fast rates*) typically scale as $\sqrt{L^\star}$ and imply faster convergence for "easy" problems, interpolating between the optimal $\sqrt{T}$ rate for worst-case instances and constant/logarithmic regret for noise-free instances. Agarwal et al. [5] observed that existing techniques appear to be inadequate to achieve this type of guarantee. Beyond simply asking whether first-order regret can be achieved, they also asked whether it can be achieved *efficiently*, which is essential for real-world deployment. Subsequently, Allen-Zhu, Bubeck, and Li [6] provided an inefficient algorithm with an optimal first-order regret guarantee, thereby resolving the former question, but the existence of efficient first-order algorithms remained open.

**Contributions.** We provide the first optimal and efficient contextual bandit algorithm with a first-order regret guarantee, thereby resolving the second open problem raised by Agarwal et al. [5]. Our algorithm, FastCB, builds on a recent line of research that develops efficient contextual bandit algorithms based on the computational primitive of (online/offline) *supervised regression* [43, 32, 29, 58], and is efficient in terms of queries to an *online oracle* for regression with the logarithmic loss. Beyond attaining first-order regret, FastCB inherits all the benefits of recent algorithms based on regression: it is simple, practical, accommodates flexible function classes, requires no statistical assumptions beyond realizability, and enjoys strong empirical performance.

**Technical highlights.** By invoking the framework of regression oracles, our algorithm design approach deviates sharply from prior work on first-order regret guarantees and necessitates the use of techniques that are novel even in the context of statistical learning. At a high-level, the design of FastCB leverages two key techniques:

1. *First-order regret for classification via logarithmic loss.* We show that algorithms based on regression with least-squares, as used in prior work [29, 58, 34, 23], fail to attain first-order regret, even for the simpler problem of cost-sensitive classification in statistical learning. In spite of this apparent setback, we show that regression with the logarithmic loss *does* lead to first-order regret for statistical learning. This is established through a new analysis based on an information-theoretic quantity called the *triangular discrimination* [66, 44, 62].
2. *Reweighted inverse gap weighting.* Moving from statistical learning to contextual bandits, we transform predictions into distributions over actions using a scale-sensitive refinement to the *inverse-gap weighting scheme* used in the SquareCB algorithm [1, 29]; we show that the error of this scheme is controlled by the triangular discrimination.

Summarizing, our approach leverages **prediction** via the logarithmic loss, **allocation** via reweighted inverse gap weighting, and **triangular discrimination** as the bridge from prediction to allocation.

**Empirical results.** In Section 4, we evaluate FastCB on the large-scale contextual bandit benchmark of Bietti et al. [13] and find it typically outperforms SquareCB and other non-adaptive baselines [35].

**Organization.** Section 2 contains our algorithm and main theorem. Section 3 describes motivation and analysis ideas behind FastCB, beginning from new techniques for statistical learning with regression-based classifiers. Experimental results are given in Section 4. Detailed discussion of related work and additional examples are deferred to Appendix A.1 and Appendix C.2, respectively.

**Additional notation.** For functions $f, g : \mathcal{X} \to \mathbb{R}_+$, we write $f = \mathcal{O}(g)$ if there exists a constant $C > 0$ such that $f(x) \le Cg(x)$ for all $x \in \mathcal{X}$. We write $f = \widetilde{\mathcal{O}}(g)$ if $f = \mathcal{O}(g \max\{1, \mathrm{polylog}(g)\})$.

## 2 Main Result: An Efficient First-Order Algorithm for Contextual Bandits

We begin by formally introducing the contextual bandit model. At each round $t \in [T]$, the learner observes a context $x_t \in \mathcal{X}$, selects an action $a_t \in \mathcal{A}$, then observes a loss $\ell_t(a_t) \in [0, 1]$ for the

action they selected. We assume that each loss function $\ell_t : \mathcal{A} \to [0,1]$ is drawn independently from a fixed distribution $\mathbb{P}_{\ell_t}(\cdot \mid x_t)$, where $\mathbb{P}_{\ell_1}, \ldots, \mathbb{P}_{\ell_T}$ and $x_1, \ldots, x_T$ are selected by a potentially adaptive adversary.

We make a standard *realizability* assumption [24, 2, 32, 29]. Namely, we assume that the learner has access to a class of value functions $\mathcal{F} \subset (\mathcal{X} \times \mathcal{A} \to [0,1])$ (e.g., neural networks, kernels, or random forests) that models the mean of the loss distribution.

**Assumption 1** (Realizability)**.** *There exists a regression function $f^\star \in \mathcal{F}$ such that for all $t$, $f^\star(x,a) = \mathbb{E}[\ell_t(a) \mid x_t = x]$.*

The aim of the learner is to minimize their *regret* to the optimal policy $\pi^\star(x) := \operatorname{argmin}_{a\in\mathcal{A}} f^\star(x,a)$:

$$\mathbf{Reg}_{\mathsf{CB}}(T) := \sum_{t=1}^{T} \ell_t(a_t) - \sum_{t=1}^{T} \ell_t(\pi^\star(x_t)). \tag{1}$$

For each $f \in \mathcal{F}$, we let $\pi_f(x) := \operatorname{argmin}_{a\in\mathcal{A}} f(x,a)$ be the induced policy, and let $\Pi := \{\pi_f \mid f \in \mathcal{F}\}$ be the induced policy class.

### 2.1 Algorithm and Main Result

FastCB builds on the SquareCB algorithm of Foster and Rakhlin [29], which provides an efficient, minimax-optimal reduction to online regression with the square loss. Compared to SquareCB and other subsequent algorithms based on online regression [34, 23], the first twist here is that we instead build on the computational primitive of online regression with *the logarithmic loss*; while this point is inconsequential for worst-case guarantees, we establish through upper and lower bounds (Section 3.1) that it is a fundamental distinction where first-order guarantees are concerned.

**Online regression oracles.** In more detail, an online regression oracle, which we denote by $\mathbf{Alg}_{\mathsf{KL}}$ (for "Kullback-Leibler") operates in the following protocol: For each time $t$, the algorithm receives a context-action pair $(x_t, a_t)$, produces a prediction $\widehat{y}_t \in [0,1]$, then receives a response $y_t$. The algorithm's prediction error is measured through the logarithmic/cross-entropy loss

$$\ell_{\log}(\widehat{y}, y) := y \log(1/\widehat{y}) + (1-y)\log(1/(1-\widehat{y})). \tag{2}$$

Its goal is to ensure that the log-loss regret to the function class $\mathcal{F}$ is minimized for all sequences.

**Assumption 2.** *The algorithm $\mathbf{Alg}_{\mathsf{KL}}$ guarantees that for every (possibly adaptively chosen) sequence $x_{1:T}, a_{1:T}, y_{1:T}$, the log-loss regret is bounded by a function $\mathbf{Reg}_{\mathsf{KL}}(T)$:*

$$\sum_{t=1}^{T} \ell_{\log}(\widehat{y}_t, y_t) - \inf_{f\in\mathcal{F}} \sum_{t=1}^{T} \ell_{\log}(f(z_t), y_t) \le \mathbf{Reg}_{\mathsf{KL}}(T). \tag{3}$$

Online regression with the logarithmic loss—also known as sequential probability assignment—is a fundamental and well-studied problem in online learning, and there are many efficient algorithms available for function classes of interest [25, 67, 40, 37, 52, 55, 33, 48]; see Appendix C.2 for examples. While log-loss regret is a more stringent notion of performance than square loss regret, it nonetheless has a relatively mature theory [57, 51, 20, 14] characterizing optimal rates.

**The algorithm.** FastCB (Algorithm 1) is a reduction that efficiently transforms any online regression oracle satisfying Assumption 2 into a contextual bandit algorithm with an optimal first-order regret bound. At each round $t$, the algorithm first (Line 4) computes the estimated loss

$$\widehat{y}_t(x_t, a) := \mathbf{Alg}^{(t)}_{\mathsf{KL}}(x_t, a\,; \{(x_i, a_i, \ell_i(a_i))\}_{i=1}^{t-1}) \tag{4}$$

predicted by the regression oracle for each action $a$.[1] Next (Line 6), FastCB uses these estimates to assign a probability of being played to each action $a$ via a scale-sensitive refinement to the inverse gap weighting strategy used in SquareCB [1, 29], which we call *reweighted inverse gap weighting*. Let $b_t := \operatorname{argmin}_{a\in\mathcal{A}} \widehat{y}_t(x_t, a)$ be the greedy action, then define

$$p_{t,a} := \frac{\widehat{y}_t(x_t, b_t)}{A\widehat{y}_t(x_t, b_t) + \gamma(\widehat{y}_t(x_t, a) - \widehat{y}_t(x_t, b_t))} \quad \forall a \neq b_t, \quad \text{and} \quad p_{t,b_t} := 1 - \sum_{a\neq b_t} p_{t,a}. \tag{5}$$

[1] See Appendix C.1 for a more detailed formal description of the oracle model.

**Algorithm 1** FastCB ("Fast Rates for Contextual Bandits")

**parameters**:
  Learning rate $\gamma > 0$.
  Online regression oracle $\mathbf{Alg}_{\mathsf{KL}}$.
**for** $t = 1, \ldots, T$ **do**
  Receive context $x_t$.
  `// Compute oracle's predictions (Eq. (4)).`
  For each action $a \in \mathcal{A}$, compute $\widehat{y}_t(x_t, a) := \mathbf{Alg}^{(t)}_{\mathsf{KL}}(x_t, a\,; \{(x_i, a_i, \ell_i(a_i))\}_{i=1}^{t-1})$.
  Let $b_t \in \operatorname{argmin}_{a\in\mathcal{A}} \widehat{y}_{t,a}$.
  `// Reweighted inverse gap weighting.`
  For each $a \neq b_t$, define $p_{t,a} = \frac{\widehat{y}_t(x_t,b_t)}{A\widehat{y}_t(x_t,b_t)+\gamma(\widehat{y}_t(x_t,a)-\widehat{y}_t(x_t,b_t))}$. Let $p_{t,b_t} = 1 - \sum_{a\neq b_t} p_{t,a}$.
  Sample $a_t \sim p_t$ and observe loss $\ell_t(a_t)$.
  Update $\mathbf{Alg}_{\mathsf{KL}}$ with example $(x_t, a_t, \ell_t(a_t))$.
**end for**

Given this distribution, FastCB simply samples $a_t \sim p_t$, then updates the oracle with the resulting tuple $(x_t, a_t, \ell_t(a_t))$. Our main theorem shows that this leads to an optimal first-order regret bound.[2]

**Theorem 1** (Main theorem)**.** *Suppose Assumptions 1 and 2 hold. Then Algorithm 1 guarantees that for all sequences with* $\mathbb{E}\big[\sum_{t=1}^{T} \ell_t(\pi^\star(x_t))\big] \leq L^\star$*, by choosing* $\gamma = \sqrt{AL^\star/3\mathbf{Reg}_{\mathsf{KL}}(T)} \vee 10A$*,*

$$\mathbb{E}[\mathbf{Reg}_{\mathsf{CB}}(T)] \leq 40\sqrt{L^\star \cdot A\mathbf{Reg}_{\mathsf{KL}}(T)} + 600A\mathbf{Reg}_{\mathsf{KL}}(T). \tag{6}$$

The dominant term in this regret bound scales with $\sqrt{L^\star}$ whenever the oracle $\mathbf{Alg}_{\mathsf{KL}}$ attains a fast $\log(T)$-type regret bound. As a simple example, whenever $\mathcal{F}$ is finite, we can instantiate $\mathbf{Alg}_{\mathsf{KL}}$ so that $\mathbf{Reg}_{\mathsf{KL}}(T) \leq \log|\mathcal{F}|$ [67], whereby FastCB enjoys optimal [2] first-order regret:

$$\mathbb{E}[\mathbf{Reg}_{\mathsf{CB}}(T)] \leq \mathcal{O}\Big(\sqrt{L^\star \cdot A\log|\mathcal{F}|} + A\log|\mathcal{F}|\Big).$$

Beyond first-order regret, FastCB inherits all of the advantages of online regression-based algorithms:

- *Efficiency and simplicity.* The memory and runtime used by the algorithm—on top of what is required by the regression oracle—scales only as $\mathcal{O}(A)$ per step; implementation is trivial.
- *Flexibility.* Working with regression as a primitive means that the algorithm easily accomodates rich, potentially nonparametric function classes, and we can instantiate Theorem 1 to get provable end-to-end regret guarantees for concrete classes of interest. For example, for linear models in $\mathbb{R}^d$ we can efficiently attain $\mathbf{Reg}_{\mathsf{KL}}(T) \leq \mathcal{O}(d\log(T))$ [25, 40], which yields a first-order bound of $\mathbf{Reg}_{\mathsf{CB}}(T) \lesssim \sqrt{L^\star \cdot Ad}$; our result is new even for this simple special case. Similar guarantees are available for kernels, generalized linear models, and beyond. On the other hand, even for function classes where provable algorithms are not available, regression is amenable to practical heuristics (e.g., gradient descent for non-convex models). See Appendix C.2 for detailed examples.

Additionally, FastCB enjoys strong empirical performance on a standard large-scale contextual bandit benchmark suite (cf. Section 4).

**On the regression oracle model.** As a disclaimer, we caution that our algorithm is efficient in terms of an oracle for online regression, while Agarwal et al. [5] originally asked for an algorithm that is efficient in terms of a *cost-sensitive classification oracle* capable of solving the policy optimization problem $\operatorname{argmin}_{\pi\in\Pi} \sum_{t=1}^{T} \ell_t(\pi(x_t))$. Hence, while FastCB is the first algorithm with first-order regret that is efficient in *any* oracle model, it does not formally solve the open problem. Nonetheless, there are strong reasons to prefer a solution based on regression over one based on classification. First, cost-sensitive classification is intractable to implement even for simple function classes for which regression can be solved efficiently [29]. Setting this issue aside, (online) regression-based algorithms are typically simpler and faster than classification-based algorithms, and multiple empirical evaluations have shown that algorithms based on regression dominate those based on classification [32, 13, 35].

[2] While we assume that an upper bound on the optimal loss is known for simplicity, it is trivial to extend to the case where $L^\star$ is unknown by running the algorithm in epochs, setting $\gamma$ in terms of the learner's esimated loss $L_t = \sum_{\tau=1}^{t} \ell_\tau(a_\tau)$, and applying the doubling trick. Theorem 1 also readily extends to high-probability.

# 3 Overview of Analysis

We now outline the key algorithmic principles and analysis ideas behind FastCB. First, in Section 3.1, we take a step back and consider the sub-problem of cost-sensitive classification in statistical learning. We establish that approaches based on least-squares fail to attain first-order regret (Theorem 2) for cost-sensitive classification, then show how to fix this problem using log-loss regression (Theorem 3). With this result in hand, we move to the contextual bandit setting and transform predictions into distributions over actions using the reweighted inverse-gap weighting scheme in (5), which exploits small losses. Our main result here shows that this scheme satisfies a first-order variant of the "per-round" minimax inequality of Foster and Rakhlin [29], which links the instantaneous contextual bandit regret to the triangular divergence of the regression oracle on a per-round basis (Theorem 4). Full proofs for all theorems are deferred to Appendices B and C.

## 3.1 Warmup: First-order Regret Bounds for Plug-In Classifiers

In the simpler problem of cost-sensitive classification in statistical learning, the literature on *plug-in classification* shows that whenever realizability conditions such as Assumption 1 hold, we can obtain optimal worst-case regret by taking the greedy policy/classifier induced by a least-squares estimator. We will show that this approach fails to attain first-order regret, and how to fix it.

The statistical learning setup we consider is as follows. We receive a dataset $D_n$ consisting of $n$ context-loss pairs $(x_t, \ell_t) \sim \mathcal{D}$ i.i.d., where the entire loss function $\ell_t : \mathcal{A} \to [0,1]$ is observed. Analogously to Assumption 1, we assume access to a function class $\mathcal{F} \subseteq (\mathcal{X} \times \mathcal{A} \to [0,1])$ such that $\mathbb{E}_{\mathcal{D}}[\ell(a) \mid x] = f^\star(x,a)$ for some $f^\star \in \mathcal{F}$, and take $\Pi := \{\pi_f \mid f \in \mathcal{F}\}$ as the induced class of policies. Our goal is to learn a policy $\widehat{\pi} : \mathcal{X} \to \mathcal{A}$ such that the regret (or excess risk)

$$\mathbf{Reg}_{\mathsf{CSC}}(n) := L(\widehat{\pi}) - L^\star \tag{7}$$

is small, where $L(\pi) := \mathbb{E}_{\mathcal{D}}[\ell(\pi(x))]$ and $L^\star := L(\pi^\star)$, with $\pi^\star := \pi_{f^\star}$. Formally, this an easier problem than contextual bandits, since any algorithm with a regret bound for contextual bandits yields a bound on the cost-sensitive classification regret (7) via online-to-batch conversion.

A classical result in statistical learning [65, 54, 59] shows that if we compute the policy/classifier $\widehat{\pi} := \operatorname{argmin}_{\pi \in \Pi} \sum_{i=1}^n \ell_i(\pi(x_i))$ that minimizes the empirical risk, we are guaranteed to obtain a first-order regret bound of the form[3]

$$\mathbb{E}[L(\widehat{\pi})] - L^\star \lesssim \sqrt{\frac{L^\star \cdot \log|\mathcal{F}|}{n}} + \frac{\log|\mathcal{F}|}{n}. \tag{8}$$

This is an optimal first-order guarantee, but computing $\widehat{\pi}$ is typically computationally intractable, even for relatively simple policy classes. As an alternative, the approach of plug-in classification aims to use the realizability assumption to develop algorithms based on the more tractable primitive of regression. Here, another classical result (e.g., [8]), shows that if we perform least squares[4]

$$\widehat{f}_{\mathsf{LS}} := \operatorname*{argmin}_{f \in \mathcal{F}} \sum_{t=1}^n \sum_{a \in \mathcal{A}} (f(x_t, a) - \ell_t(a))^2,$$

and take $\widehat{\pi}_{\mathsf{LS}} := \pi_{\widehat{f}_{\mathsf{LS}}}$ as our classifier, then under the realizability assumption we are guaranteed

$$\mathbb{E}[L(\widehat{\pi}_{\mathsf{LS}})] - L^\star \lesssim \sqrt{\frac{A \log|\mathcal{F}|}{n}}. \tag{9}$$

While this rate is minimax optimal, it is not first-order, and first-order regret bounds for plug-in classification are conspicuously absent from the literature. We show that this is fundamental.

**Theorem 2** (Failure of least-squares for plug-in classification)**.** *Let $\mathcal{A} = [2]$ and $\mathcal{X} = [2]$. For every $n > 10^8$, there exists a function class $\mathcal{F} \subseteq (\mathcal{X} \times \mathcal{A} \to [0,1])$ with $|\mathcal{F}| = 2$, and a realizable distribution $\mathcal{D}$ such that $L^\star \leq 2^7 \cdot \frac{1}{n} < 1$, yet $L(\widehat{\pi}_{\mathsf{LS}}) - L^\star \geq 2^{-5}\sqrt{\frac{1}{n}}$ with probability at least $1/10$.*

[3] Following the convention in contextual bandit literature, we focus on finite classes with $|\mathcal{F}| < \infty$ in this discussion, but one can extend our observations to general classes, e.g., using the machinery of Zhang [70].

[4] This result is well-known in the binary setting. We are not aware of a reference for the multiclass/cost-sensitive version here, though it is implicit in many recent works on contextual bandits.

Since the instance in this theorem always has $\sqrt{\frac{L^\star \cdot A \log|\mathcal{F}|}{n}} \lesssim \frac{1}{n}$, we conclude that plug-in classification with least-squares fails to attain the first-order regret bound in (8).[5]

### 3.1.1 Fast Rates for Plug-In Classifiers: Triangular Discrimination and Logarithmic Loss

It would appear we are at an impasse, as Theorem 2 shows that square loss regression oracles of the type used in Foster and Rakhlin [29] are unlikely to attain first-order regret bounds on their own. However, the plug-in classification approach is not completely doomed. All we need to do to fix this issue is change the loss function, and instead perform regression with the *logarithmic loss*.

To understand why plug-in least-squares fails, and how it can be improved, it will be helpful to review the key steps in the analysis leading to the rate (9).

**Step 1**. First, using a generic regret decomposition based on realizability, for any $f$ we have

$$L(\pi_f) - L^\star \lesssim \max_{\pi \in \{\pi_f, \pi^\star\}} \mathbb{E}_{\mathcal{D}}\big|f(x,\pi(x)) - f^\star(x,\pi(x))\big|. \tag{10}$$

**Step 2**. Next, by Cauchy-Schwarz, for any policy $\pi$ we have

$$\mathbb{E}_{\mathcal{D}}\big|f(x,\pi(x)) - f^\star(x,\pi(x))\big| \leq \Big(\mathbb{E}_{\mathcal{D}}\big|f(x,\pi(x)) - f^\star(x,\pi(x))\big|^2\Big)^{1/2}, \tag{11}$$

which we may further upper bound by $\Big(\sum_{a\in\mathcal{A}} \mathbb{E}_{\mathcal{D}}\big|f(x,a) - f^\star(x,a)\big|^2\Big)^{1/2}$.

**Step 3**. Finally, under realizability, a standard concentration argument based on Bernstein's inequality implies that the least-squares estimator satisfies

$$\mathbb{E}\left[\sum_{a\in\mathcal{A}} \mathbb{E}_{\mathcal{D}}\big|\widehat{f}_{\mathsf{LS}}(x,a) - f^\star(x,a)\big|^2\right] \lesssim \frac{A\log|\mathcal{F}|}{n}. \tag{12}$$

Combining this bound with **Step 2**, we conclude that $\mathbb{E}[L(\widehat{\pi}_{\mathsf{LS}})] - L^\star \leq \sqrt{A\log|\mathcal{F}|/n}$.

The issue here is that even in the presence of low noise, the squared error in (12) shrinks no faster than $\frac{1}{n}$. This holds even if $L^\star \propto \frac{1}{n}$, as in the lower bound construction for Theorem 2. Consequently, once we apply Cauchy-Schwarz in **Step 2**, we lose all hope of attaining a first-order bound.

Our starting point toward improving this result is a refined application of Cauchy-Schwarz, by which we can replace the right hand side of (11) with

$$\left(\mathbb{E}_{\mathcal{D}}[f(x,\pi(x)) + f^\star(x,\pi(x))] \cdot \mathbb{E}_{\mathcal{D}}\left[\frac{\big(f(x,\pi(x)) - f^\star(x,\pi(x))\big)^2}{f(x,\pi(x)) + f^\star(x,\pi(x))}\right]\right)^{1/2}. \tag{13}$$

The ratio term above is closely related to an information-theoretic divergence measure known as the *triangular discrimination*, which we define for $p, q \in \mathbb{R}^A_+$ as[6]

$$D_\Delta(p \,\|\, q) := \sum_a \frac{(p_a - q_a)^2}{p_a + q_a}. \tag{14}$$

The triangular discrimination—also known as the symmetric $\chi^2$-divergence and Vincze-Le Cam distance—is a fundamental, oft-overlooked quantity in information theory [66, 44, 62]. Since readers may be unfamiliar, we record some basic facts.

**Proposition 1** (Topsøe [62])**.** *The triangular discrimination $D_\Delta$, over the domain $\Delta_A$, i) is the $f$-divergence given by $f(t) = \frac{(t-1)^2}{t+1}$, ii) is the square of a distance metric, and iii) is equivalent (up to a multiplicative constant) to both the Hellinger distance and Jensen-Shannon divergence.*

[5] Since regret for $\widehat{\pi}_{\mathsf{LS}}$ is non-negative, Theorem 2 also implies a lower bound of $\sqrt{\frac{1}{n}}$ on the expected regret.

[6] The triangular discrimination is traditionally defined over the simplex $\Delta_A$, but for our purposes it is useful to work with the entire positive orthant.

The triangular discrimination turns out to be "just right" for our purposes, in that it is both i) large enough to facilitate the scale-sensitive application of Cauchy-Schwarz in (13), and ii) small enough (compared to the more standard $\chi^2$-divergence) to facilitate minimizing from samples.

Returning to (13), we can upper bound in terms of the triangular discrimination and leverage a certain *self-bounding* property that it satisfies to arrive at the following improvement on **Step 1**/**Step 2**.

**Lemma 1** (Regret decomposition for triangular discrimination)**.** *For any* $f : \mathcal{X} \times \mathcal{A} \to [0, 1]$,

$$L(\pi_f) - L^\star \leq 8(L^\star \cdot \mathbb{E}_{\mathcal{D}}[D_\Delta(f^\star(x, \cdot) \,\|\, f(x, \cdot))])^{1/2} + 17\,\mathbb{E}_{\mathcal{D}}[D_\Delta(f^\star(x, \cdot) \,\|\, f(x, \cdot))]. \tag{15}$$

Lemma 1 shows that low triangular discrimination (i.e. $\mathbb{E}_{\mathcal{D}}[D_\Delta(f^\star(x, \cdot) \,\|\, f(x, \cdot))] \propto 1/n$) suffices for an optimal first-order regret bound. What remains is to find an estimator $\widehat{f}$ that minimizes this quantity given only samples. Our key observation here is that the triangular discrimination satisfies a refined variant of Pinsker's inequality (originally due to Topsøe [62]), which allows us to bound it in terms of the Kullback-Leibler divergence:[7]

$$D_\Delta(f^\star(x, \cdot) \,\|\, f(x, \cdot)) = \sum_a \frac{(f(x, a) - f^\star(x, a))^2}{f(x, a) + f^\star(x, a)} \leq 2 \sum_a d_{\mathrm{KL}}(f^\star(x, a) \,\|\, f(x, a)), \tag{16}$$

where $d_{\mathrm{KL}}(p \,\|\, q) := p \log(p/q) + (1-p) \log((1-p)/(1-q))$ is the binary KL-divergence. This bound suggests that we should minimize the logarithmic loss, since—under the realizability assumption—this loss is closely related to the KL- divergence. In particular, we show (Theorem 6 in Appendix B), that by taking the estimator

$$\widehat{f}_{\mathsf{KL}} := \operatorname*{argmin}_{f \in \mathcal{F}} \sum_{t=1}^{n} \sum_{a \in \mathcal{A}} \ell_{\log}(f(x_t, a), \ell_t(a)),$$

we are guaranteed that with high probability, $\mathbb{E}_{\mathcal{D}}\big[D_\Delta(f^\star(x, \cdot) \,\|\, \widehat{f}_{\mathsf{KL}}(x, \cdot))\big] \lesssim \frac{A \log|\mathcal{F}|}{n}$. Putting everything together, we arrive at a first-order regret bound for the plug-in classifier $\widehat{\pi}_{\mathsf{KL}} := \pi_{\widehat{f}_{\mathsf{KL}}}$.[8]

**Theorem 3** (First-order regret bound for plug-in classification)**.** *Let* $\delta \in (0, 1)$. *Suppose that Assumption 3 holds. Then with probability at least* $1 - \delta$, *we have*

$$L(\widehat{\pi}_{\mathsf{KL}}) - L^\star \leq 16\sqrt{\frac{L^\star \cdot A\,(\log|\mathcal{F}| + \log(A/\delta))}{n}} + 68\frac{A\,(\log|\mathcal{F}| + \log(A/\delta))}{n}.$$

Interestingly, applications of the triangular discrimination similar to Lemma 1 have been discovered recently across many branches of mathematics, including theoretical computer science (communication complexity) [69], probability, and group theory (construction of group homomorphisms) [28, 11, 53]. Additionally, Bubeck and Sellke [18] use a related *non-negative* $\chi^2$*-divergence* to provide first-order Bayesian regret bounds for Thompson sampling for the multi-armed bandit.

### 3.2 Moving to Contextual Bandits: Inverse Gap Weighting meets Triangular Discrimination

FastCB builds on the development for plug-in classifiers in Section 3.1, but with two key differences. First, since we need to make decisions on the fly for arbitrary sequences of contexts, the algorithm estimates losses using an *online* regression oracle for the logarithmic loss, as described in Assumption 2. Second, and more importantly, since the algorithm receives partial feedback, the strategy for selecting actions is critical. Here our main technical result shows that the reweighted inverse gap weighting strategy (5) satisfies a certain *per-round* inequality that links the instantaneous contextual bandit error to the triangular discrimination between the oracle's prediction $\widehat{y}_t$ and the true loss function $f^\star$.

**Theorem 4** (First-order per-round inequality)**.** *Let* $y \in [0, 1]^A$ *be given and* $b \in \operatorname{argmin}_a y_a$. *Define* $p_a = \frac{y_b}{A y_b + \gamma(y_a - y_b)}$ *for* $a \neq b$, *and* $p_b = 1 - \sum_{a \neq b} p_a$. *If* $\gamma \geq 2A$, *then for all* $f \in [0, 1]^A$ *and* $a^\star \in \operatorname{argmin}_a f_a$, *we have*

$$\underbrace{\sum_a p_a(f_a - f_{a^\star})}_{\text{CB regret}} \leq \underbrace{\frac{5A}{\gamma} \sum_a p_a f_a}_{\text{bias from } \textit{exploring}} + \underbrace{7\gamma \sum_a p_a \frac{(y_a - f_a)^2}{y_a + f_a}}_{\text{error from } \textit{exploiting}}. \tag{17}$$

[7] Note that the triangular discrimination is critical here as the *opposite* inequality holds for the $\chi^2$-divergence.

[8] The dependence on $A$ in this result can be improved under additional assumptions on the loss distribution. As an example, in Appendix B we remove the leading $A$ factor for the special case of multiclass classification.

The inequality (17) may be thought of as an algorithmic analogue of the refined Cauchy-Schwarz lemma (15) used in our analysis for statistical learning, with the learning rate $\gamma$ modulating the tradeoff between exploration and exploitation. Applying the inequality for each step $t$ (with $p = p_t$, $y = \widehat{y}_t(x_t, \cdot)$, and $f = f^\star(x_t, \cdot)$), and using the Pinsker-type inequality (16), we are guaranteed that

$$\mathbb{E}[\mathbf{Reg}_{\mathsf{CB}}(T)] \leq \frac{5A}{\gamma}\,\mathbb{E}[L_T] + 14\gamma \cdot \mathbf{Reg}_{\mathsf{KL}}(T), \tag{18}$$

where $L_T := \sum_{t=1}^{T} \ell_t(a_t)$. By a standard argument, this implies the main result in Theorem 1.

Compared to the per-round inequality used to analyze the original version of SquareCB in Foster and Rakhlin [29], the main improvement given by Theorem 4 is that, by reweighting—which leads to less exploration when the optimal loss is small—we are able to replace the constant exploration bias of order $\frac{A}{\gamma}$ incurred by SquareCB with the scale-sensitive bias term $\frac{A}{\gamma} \cdot \sum_a p_a f_a$ in (17), leading to a first-order bound. The price for this improvement is that we must now minimize the triangular discrimination rather than the squared error used by SquareCB, but this is taken care of by the log-loss oracle.

# 4 Experiments

We compared the performance of FastCB to that of the de-facto alternative, SquareCB [29] in the large-scale contextual bandit evaluation suite ("bake-off") of Bietti et al. [13]. We found that FastCB typically enjoys improved performance, particularly on datasets where the optimal loss $L^\star$ is small. As a secondary observation, we found that using generalized linear models with the log loss rather than a linear model with the square loss (as in prior work [13, 35]) leads to substantial improvements, even for SquareCB. We summarize the results here; further details are given in Appendix E.

**Datasets.** The *contextual bandit bake-off* is a collection of over 500 multiclass, multilabel, and cost-sensitive classification datasets available on the `openml.org` platform [64]. The collection was introduced in Bietti et al. [13] for the purpose of benchmarking oracle-based contextual bandit algorithms. Following Bietti et al. [13] we use the multiclass classification datasets (each context $x$ has a "correct" label $y$ associated with it) to simulate bandit feedback by assigning loss $0$ if the learner predicts the correct label and $1$ otherwise.

**Algorithms and oracle.** We use the standard implementation of SquareCB in the Vowpal Wabbit (VW) online learning library,[9] as used by Foster et al. [35]. We also implement FastCB in VW.

For both algorithms, we instantiate the oracle as performing online logistic regression with a fixed dataset-dependent feature map. This choice is convenient because i) it naturally produces predictions in $[0, 1]$, as required by FastCB, and ii), it formally meets our oracle requirements as it is equivalent to online log-loss regression with a generalized linear model. It can also be viewed as an admissible online square loss oracle, as required by SquareCB (cf. further discussion in Appendix E). We additionally instantiate SquareCB with a linear model and the square loss, which was shown to be the strongest non-adaptive method in prior evaluations [35]. We do not compare with high-performing adaptive algorithms like RegCB and AdaCB (as used in [13, 35]) as these algorithmic modifications are somewhat complementary, and we expect they can be incorporated into FastCB. All oracles are trained with the default VW learning rule, which performs online gradient descent with adaptive updates [27, 41, 56].

For both FastCB and SquareCB, we apply inverse gap weighting (the reweighted and original version, respectively) with a time-varying learning rate $\gamma_t$.[10] Following Foster et al. [35], we set $\gamma_t = \gamma_0 t^\rho$, where $\gamma_0 \in \{10, 50, 100, 400, 700, 10^3\}$ and $\rho \in \{.25, .5\}$ are hyperparameters.

**Evaluation.** We evaluate the performance of each algorithm using *progressive validation* (PV) loss, defined as $L_{\mathsf{PV}}(T) = \frac{1}{T}\sum_{t=1}^{T} \ell_t(a_t)$ [15]. Following Bietti et al. [13], we define a given algorithm as beating another algorithm *significantly* on a given dataset using an approximate Z-test. See Appendix E for details. For each pair $(a, b)$ of algorithms, Figure 1 (top row) displays the number of datasets where $a$ beats $b$ significantly, minus the number of datasets where $b$ beats $a$ significantly.

[9] `https://vowpalwabbit.org`

[10] Concretely, for FastCB, we set $\gamma = \gamma_t$ in Line 6 in round $t$, and likewise for SquareCB.

| ↓ vs → | S.S | S.L | F.L |
|---|---|---|---|
| SquareCB.S | - | -55 | -66 |
| SquareCB.L | 55 | - | -11 |
| **FastCB.L** | **66** | **11** | - |

| ↓ vs → | S.S | S.L | F.L |
|---|---|---|---|
| SquareCB.S | - | -54 | -64 |
| SquareCB.L | 54 | - | -3 |
| **FastCB.L** | **64** | **3** | - |

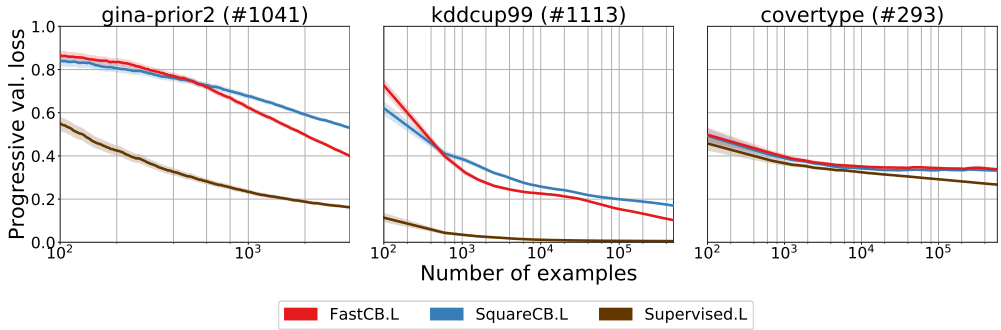


Figure 1: *Top:* Head-to-head win-loss differences. Each entry indicates the statistically significant win-loss difference between the row algorithm and the column algorithm. *Top-Left:* All hyperparameters are optimized on each dataset. *Top-Right:* Best fixed hyperparameter configuration across all datasets; only the oracle's learning rate is optimized per-dataset. *Bottom:* Progressive validation results for representative datasets depicting significant wins for FastCB.L (left, center) and a loss (right).

**Results.** We find (Figure 1, top row) that FastCB with the logistic loss oracle (FastCB.L) has a positive win-loss difference against SquareCB with both logistic and square loss oracles (SquareCB.L/SquareCB.S), suggesting the strongest overall performance. This holds both when hyperparameters are optimized on a per-dataset basis, and for the best global hyperparameter configuration.

Interestingly, our results suggest that the largest gains come from switching from the square loss oracle to the logistic loss oracle (SquareCB.S vs. SquareCB.L), while the gains from switching from the original inverse gap weighting strategy to our reweighted version (SquareCB.L vs. FastCB.L) are more marginal. Inspecting the results in more detail, we find that when we compare FastCB.L and SquareCB.L with hyperparameters optimized on a per-dataset basis, FastCB.L wins on 14/17 of the datasets in which either algorithm wins significantly, and that all but two of these 14 datasets have $L^\star \leq 0.2$. This suggests that the reweighted inverse gap weighting strategy is indeed helpful when $L^\star$ is small. Figure 1 (bottom row) displays progressive validation performance for FastCB.L and SquareCB.L for three representative datasets which illustrate this phenomenon.

The fact that FastCB.L does not strictly improve over SquareCB.L on every dataset, in spite of being very similar, might be attributed to the fact that the constants in the per-round inequality (17) are worse than those in the corresponding inequality for SquareCB.L, suggesting worse performance when $L^\star$ is not small. Thus, a fruitful future direction might be to find a strategy with optimal constants for (17).

## 5 Discussion

We have given the first efficient algorithm with optimal first-order regret for contextual bandits, resolving a variant of the open problem posed by Agarwal et al. [5]. Here we briefly mention some extensions. First, we believe that our techniques can also be used to obtain first-order guarantees for stochastic contextual bandits with an *offline* log-loss oracle (à la Simchi-Levi and Xu [58])—albeit with a more technical analysis—and plan to address this in an extended version of the paper. As another extension, in Appendix D, we show how to efficiently obtain a first-order regret bound when working with rewards rather than losses, which is useful when no policy accumulates much reward, as is common in personalization applications. Several other extensions appear to be straightforward, including working with infinite action spaces [34] and infinite function classes.

We close with some directions for future work. Directly relevant to our theoretical results is to continue the investigation into adaptivity in contextual bandits and reinforcement learning. More broadly, while triangular discrimination has been used in various mathematics disciplines, we are not aware of many applications in algorithm design. Are there other uses for the triangular discrimination in machine learning? We look forward to pursuing these directions.

# References


[1] N. Abe and P. M. Long. Associative reinforcement learning using linear probabilistic concepts. In *International Conference on Machine Learning*, 1999.

[2] A. Agarwal, M. Dudík, S. Kale, J. Langford, and R. E. Schapire. Contextual bandit learning with predictable rewards. In *International Conference on Artificial Intelligence and Statistics*, 2012.

[3] A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, 2014.

[4] A. Agarwal, S. Bird, M. Cozowicz, L. Hoang, J. Langford, S. Lee, J. Li, D. Melamed, G. Oshri, O. Ribas, S. Sen, and A. Slivkins. Making contextual decisions with low technical debt. *arXiv:1606.03966*, 2016.

[5] A. Agarwal, A. Krishnamurthy, J. Langford, and H. Luo. Open problem: First-order regret bounds for contextual bandits. In *Conference on Learning Theory*, 2017.

[6] Z. Allen-Zhu, S. Bubeck, and Y. Li. Make the minority great again: First-order regret bound for contextual bandits. *International Conference on Machine Learning*, 2018.

[7] C. Allenberg, P. Auer, L. Györfi, and G. Ottucsák. Hannan consistency in on-line learning in case of unbounded losses under partial monitoring. In *International Conference on Algorithmic Learning Theory*, 2006.

[8] J.-Y. Audibert and A. B. Tsybakov. Fast learning rates for plug-in classifiers. *The Annals of statistics*, 2007.

[9] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 2002.

[10] P. Auer, N. Cesa-Bianchi, and C. Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 2002.

[11] I. Benjamini, H. Duminil-Copin, G. Kozma, and A. Yadin. Disorder, entropy and harmonic functions. *Annals of Probability*, 43(5):2332–2373, 2015.

[12] A. Beygelzimer, J. Langford, L. Li, L. Reyzin, and R. Schapire. Contextual bandit algorithms with supervised learning guarantees. In *International Conference on Artificial Intelligence and Statistics*, 2011.

[13] A. Bietti, A. Agarwal, and J. Langford. A contextual bandit bake-off. *arXiv:1802.04064*, 2018.

[14] B. Bilodeau, D. J. Foster, and D. Roy. Tight bounds on minimax regret under logarithmic loss via self-concordance. In *International Conference on Machine Learning*, 2020.

[15] A. Blum, A. Kalai, and J. Langford. Beating the hold-out: Bounds for K-fold and progressive cross-validation. In *Conference on Computational Learning Theory*, 1999.

[16] S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press, 2013.

[17] O. Bousquet, S. Boucheron, and G. Lugosi. Introduction to statistical learning theory. In *Summer School on Machine Learning*, 2003.

[18] S. Bubeck and M. Sellke. First-order bayesian regret analysis of thompson sampling. In *International Conference on Algorithmic Learning Theory*, 2020.

[19] R. J. Carroll. Adapting for heteroscedasticity in linear models. *The Annals of Statistics*, 1982.

[20] N. Cesa-Bianchi and G. Lugosi. Minimax Regret under Log Loss for General Classes of Experts. In *Conference on Computational Learning Theory*, 1999.

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

[22] N. Cesa-Bianchi, Y. Mansour, and G. Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 2007.

[23] S. Chen, F. Koehler, A. Moitra, and M. Yau. Online and distribution-free robustness: Regression and contextual bandits with Huber contamination. *arXiv:2010.04157*, 2020.

[24] W. Chu, L. Li, L. Reyzin, and R. E. Schapire. Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, 2011.

[25] T. M. Cover. Universal portfolios. *Mathematical Finance*, 1991.

[26] L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory of pattern recognition*. Springer Science & Business Media, 2013.

[27] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 2011.

[28] A. Erschler and A. Karlsson. Homomorphisms to $\mathbb{R}$ constructed from random walks. *Annales de l'Institut Fourier*, 60(6):2095–2113, 2010.

[29] D. J. Foster and A. Rakhlin. Beyond UCB: Optimal and efficient contextual bandits with regression oracles. In *International Conference on Machine Learning*, 2020.

[30] D. J. Foster, A. Rakhlin, and K. Sridharan. Adaptive online learning. In *Advances in Neural Information Processing Systems*, 2015.

[31] D. J. Foster, Z. Li, T. Lykouris, K. Sridharan, and É. Tardos. Learning in games: Robustness of fast convergence. In *Advances in Neural Information Processing Systems*, 2016.

[32] D. J. Foster, A. Agarwal, M. Dudík, H. Luo, and R. E. Schapire. Practical contextual bandits with regression oracles. *International Conference on Machine Learning*, 2018.

[33] D. J. Foster, S. Kale, H. Luo, M. Mohri, and K. Sridharan. Logistic regression: The importance of being improper. *Conference on Learning Theory*, 2018.

[34] D. J. Foster, C. Gentile, M. Mohri, and J. Zimmert. Adapting to misspecification in contextual bandits. *Advances in Neural Information Processing Systems*, 2020.

[35] D. J. Foster, A. Rakhlin, D. Simchi-Levi, and Y. Xu. Instance-dependent complexity of contextual bandits and reinforcement learning: A disagreement-based perspective. *arXiv:2010.03104*, 2020.

[36] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 1997.

[37] E. Hazan and S. Kale. An online portfolio selection algorithm with regret logarithmic in price variation. *Mathematical Finance*, 2015.

[38] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 2007.

[39] S. Ito, S. Hirahara, T. Soma, and Y. Yoshida. Tight first-and second-order regret bounds for adversarial linear bandits. *Advances in Neural Information Processing Systems*, 2020.

[40] A. Kalai and S. Vempala. Efficient algorithms for universal portfolios. *Journal of Machine Learning Research*, 2002.

[41] N. Karampatziakis and J. Langford. Online importance weight aware updates. In *Conference on Uncertainty in Artificial Intelligence*, 2011.

[42] W. M. Koolen and T. van Erven. Second-order quantile methods for experts and combinatorial games. In *Conference on Learning Theory*, 2015.

[43] A. Krishnamurthy, A. Agarwal, T.-K. Huang, H. Daumé III, and J. Langford. Active learning for cost-sensitive classification. In *International Conference on Machine Learning*, 2017.

[44] L. Le Cam. *Asymptotic methods in statistical decision theory*. Springer, 1986.

[45] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *International Conference on the World Wide Web*, 2010.

[46] Y. Li, Y. Wang, and Y. Zhou. Nearly minimax-optimal regret for linearly parameterized bandits. In *Conference on Learning Theory*, 2019.

[47] H. Luo and R. E. Schapire. Achieving all with no parameters: AdaNormalHedge. In *Conference on Learning Theory*, 2015.

[48] H. Luo, C.-Y. Wei, and K. Zheng. Efficient online portfolio with logarithmic regret. In *Advances in Neural Information Processing Systems*, 2018.

[49] T. Lykouris, K. Sridharan, and É. Tardos. Small-loss bounds for online learning with partial information. *Conference on Learning Theory*, 2018.

[50] G. Neu. First-order regret bounds for combinatorial semi-bandits. In *Conference on Learning Theory*, 2015.

[51] M. Opper and D. Haussler. Worst Case Prediction over Sequences under Log Loss. In *The Mathematics of Information Coding, Extraction and Distribution*. Springer New York, 1999.

[52] L. Orseau, T. Lattimore, and S. Legg. Soft-bayes: Prod for mixtures of experts with log-loss. In *International Conference on Algorithmic Learning Theory*, 2017.

[53] N. Ozawa. A functional analysis proof of Gromov's polynomial growth theorem. *Ann. Sci. Éc. Norm. Supér.(4)*, 51(3):549–556, 2018.

[54] D. Panchenko. Some extensions of an inequality of Vapnik and Chervonenkis. *Electronic Communications in Probability*, 2002.

[55] A. Rakhlin and K. Sridharan. Sequential probability assignment with binary alphabets and large classes of experts. *arXiv:1501.07340*, 2015.

[56] S. Ross, P. Mineiro, and J. Langford. Normalized online learning. In *Uncertainty in Artificial Intelligence*, 2013.

[57] Y. M. Shtar'kov. Universal sequential coding of single messages. *Problemy Peredachi Informatsii*, 1987.

[58] D. Simchi-Levi and Y. Xu. Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. *arXiv:2003.12699*, 2020.

[59] N. Srebro, K. Sridharan, and A. Tewari. Smoothness, low noise and fast rates. In *Advances in Neural Information Processing Systems*, 2010.

[60] A. Takeshi. *Advanced econometrics*. Harvard university press, 1985.

[61] A. Tewari and S. A. Murphy. From ads to interventions: Contextual bandits in mobile health. In *Mobile Health*, 2017.

[62] F. Topsøe. Some inequalities for information divergence and related measures of discrimination. *IEEE Transactions on Information Theory*, 2000.

[63] S. A. van de Geer. *Empirical Processes in M-Estimation.* Cambridge University Press, 2000.

[64] J. Vanschoren, J. N. Van Rijn, B. Bischl, and L. Torgo. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 2014.

[65] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 1971.

[66] I. Vincze. On the concept and measure of information contained in an observation. In *Contributions to Probability*. Elsevier, 1981.

[67] V. Vovk. A game of prediction with expert advice. In *Conference on Computational Learning Theory*, 1995.

[68] Y. Yang. Minimax nonparametric classification. I. Rates of convergence. *IEEE Transactions on Information Theory*, 1999.

[69] A. Yehudayoff. Pointer chasing via triangular discrimination. *Combinatorics, Probability and Computing*, 29(4):485–494, 2020.

[70] T. Zhang. From $\epsilon$-entropy to KL-entropy: Analysis of minimum information complexity density estimation. *The Annals of Statistics*, 2006.

[71] Z. Zhang, J. Yang, X. Ji, and S. S. Du. Variance-aware confidence set: Variance-dependent bound for linear bandits and horizon-free bound for linear mixture MDP. *arXiv:2101.12745*, 2021.

[72] D. Zhou, Q. Gu, and C. Szepesvari. Nearly minimax optimal reinforcement learning for linear mixture Markov decision processes. *arXiv:2012.08507*, 2020.

# Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 1 and Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix A.2.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] All assumptions are stated in the main body in dedicated assumption environments.
   (b) Did you include complete proofs of all theoretical results? [Yes] All theoretical results are proven in the appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See details and links in Appendix E.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 and Appendix E.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] All figures have error bars. See Appendix E for details.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See the discussion in Appendix E.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Appendix E.
   (b) Did you mention the license of the assets? [Yes] See Appendix E.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]