Stochastic Gradient Methods for Distributionally Robust Optimization with f-divergences

Hongseok Namkoong
Stanford University
hnamk@stanford.edu

John C. Duchi
Stanford University
jduchi@stanford.edu

Abstract

We develop efficient solution methods for a robust empirical risk minimization problem designed to give calibrated confidence intervals on performance and provide optimal tradeoffs between bias and variance. Our methods apply to distributionally robust optimization problems proposed by Ben-Tal et al., which put more weight on observations inducing high loss via a worst-case approach over a non-parametric uncertainty set on the underlying data distribution. Our algorithm solves the resulting minimax problems with nearly the same computational cost of stochastic gradient descent through the use of several carefully designed data structures. For a sample of size n, the per-iteration cost of our method scales as $O(\log n)$, which allows us to give optimality certificates that distributionally robust optimization provides at little extra cost compared to empirical risk minimization and stochastic gradient methods.

1 Introduction

In statistical learning or other data-based decision-making problems, it is desirable to give solutions that come with guarantees on performance, at least to some specified confidence level. For tasks such as driving or medical diagnosis where safety and reliability are crucial, confidence levels have additional importance. Classical techniques in machine learning and statistics, including regularization, stability, concentration inequalities, and generalization guarantees [6, 25] provide such guarantees, though often a more fine-tuned certificate—one with *calibrated* confidence—is desirable. In this paper, we leverage techniques from the robust optimization literature [e.g. 2], building an uncertainty set around the empirical distribution of the data and studying worst case performance in this uncertainty set. Recent work [15, 13] shows how this approach can give (i) calibrated statistical optimality certificates for stochastic optimization problems, (ii) performs a natural type of regularization based on the variance of the objective and (iii) achieves fast rates of convergence under more general conditions than empirical risk minimization by trading off bias (approximation error) and variance (estimation error) optimally. In this paper, we propose efficient algorithms for such distributionally robust optimization problems.

We now provide our formal setting. Let $\mathcal{X} \subset \mathbb{R}^d$ be a compact convex set, and for a convex function $f : \mathbb{R}_+ \to \mathbb{R}$ with $f(1) = 0$, define the f-divergence between distributions P and Q by $D_f(P\|Q) = \int f(\frac{dP}{dQ})dQ$. Letting $\mathcal{P}_{\rho,n} := \{p \in \mathbb{R}^n : p^\top \mathbb{1} = 1, p \geq 0, D_f(p\|\mathbb{1}/n) \leq \frac{\rho}{n}\}$ be an *uncertainty set* around the uniform distribution $\mathbb{1}/n$, we develop methods for solving the robust empirical risk minimization problem

$$\operatorname*{minimize}_{x \in \mathcal{X}} \sup_{p \in \mathcal{P}_{\rho,n}} \sum_{i=1}^{n} p_i \ell_i(x). \tag{1}$$

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

In problem (1), the functions $\ell_i : \mathcal{X} \to \mathbb{R}_+$ are convex and subdifferentiable, and we consider the situation in which $\ell_i(x) = \ell(x; \xi_i)$ for $\xi_i \stackrel{\text{iid}}{\sim} P_0$. We let $\ell(x) = [\ell_1(x) \ \cdots \ \ell_n(x)]^\top \in \mathbb{R}^n$ denote the vector of convex losses, so the robust objective (1) is $\sup_{p \in \mathcal{P}_{\rho,n}} p^T \ell(x)$.

A number of authors show how the robust formulation (1) provides guarantees. Duchi et al. [15] show that the objective (1) is a convex approximation to regularizing the empirical risk by variance,

$$\sup_{p \in \mathcal{P}_{\rho,n}} \sum_{i=1}^n p_i \ell_i(x) = \frac{1}{n} \sum_{i=1}^n \ell_i(x) + \sqrt{\frac{\rho}{n} \mathrm{Var}_{P_0}(\ell(x; \xi))} + o_{P_0}(n^{-\frac{1}{2}}) \tag{2}$$

uniformly in $x \in \mathcal{X}$. Since the right hand side naturally trades off good loss performance (approximation error) and minimizing variance (estimation error) which is usually non-convex, the robust formulation (1) provides a convex regularization for the standard empirical risk minimization (ERM) problem. This trading between bias and variance leads to certificates on the optimal value $\inf_{x \in \mathcal{X}} \mathbb{E}_{P_0}[\ell(x; \xi)]$ so that under suitable conditions, we have

$$\lim_{n \to \infty} \mathbb{P}\left(\inf_{x \in \mathcal{X}} \mathbb{E}_{P_0}[\ell(x; \xi)] \le u_n \right) = \mathbb{P}\left(W \ge -\sqrt{\rho}\right) \ \text{ for } W \sim \mathsf{N}(0, 1) \tag{3}$$

where $u_n = \inf_{x \in \mathcal{X}} \sup_{p \in \mathcal{P}_{\rho,n}} p^T \ell(x)$ is the optimal robust objective. Duchi and Namkoong [13] provide finite sample guarantees for the special case that $f(t) = \frac{1}{2}(t-1)^2$, making the expansion (2) more explicit and providing a number of consequences for estimation and optimization based on this expansion (including fast rates for risk minimization). A special case of their results [13, §3.1] is as follows. Let $\widehat{x}^{\text{rob}} \in \mathop{\mathrm{argmin}}_{x \in \mathcal{X}} \sup_{p \in \mathcal{P}_{\rho,n}} p^T \ell(x)$, let $\mathsf{VC}(\mathcal{F})$ denote the VC-(subgraph)-dimension of the class of functions $\mathcal{F} := \{\ell(x; \cdot) \mid x \in \mathcal{X}\}$, assume that $M \ge \ell(x; \xi)$ for all $x \in \mathcal{X}, \xi \in \Xi$, and for some fixed $\delta > 0$, define $\rho = \log \frac{1}{\delta} + 10\, \mathsf{VC}(\mathcal{F}) \log \mathsf{VC}(\mathcal{F})$. Then, with probability at least $1 - \delta$,

$$\mathbb{E}_{P_0}[\ell(\widehat{x}^{\text{rob}}; \xi)] \le u_n + O(1)\frac{M\rho}{n} \le \inf_{x \in \mathcal{X}} \left\{ \mathbb{E}_{P_0}[\ell(x; \xi)] + 2\sqrt{\frac{2\rho \mathrm{Var}_{\widehat{P}_n}(\ell(x; \xi))}{n}} \right\} + O(1)\frac{M\rho}{n} \tag{4}$$

For large n, evaluating the objective (1) may be expensive; with fixed $p = \mathbb{1}/n$, this has motivated an extensive literature in stochastic and online optimization [27, 23, 19, 16, 18]. The problem (1) does not admit quite such a straightforward approach. A first idea, common in the robust optimization literature [3], is to obtain a problem that may be written as a sum of individual terms by taking the dual of the inner supremum, yielding the convex problem

$$\inf_{x \in \mathcal{X}} \sup_{p \in \mathcal{P}_{\rho,n}} p^\top \ell(x) = \inf_{x \in \mathcal{X}, \lambda \ge 0, \eta \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n \lambda f^*\left(\frac{\ell_i(x) - \eta}{\lambda} \right) + \frac{\rho}{n}\lambda + \eta. \tag{5}$$

Here $f^*(s) = \sup_{t \ge 0}\{st - f(t)\}$ is the Fenchel conjugate of the convex function f. While the above dual reformulation is jointly convex in (x, λ, η), canonical stochastic gradient descent (SGD) procedures [23] generally fail because the variance of the objective (and its subgradients) explodes as $\lambda \to 0$. (This is not just a theoretical issue: in extensive simulations that we omit because they are a bit boring, SGD and other heuristic approaches that impose shrinking bounds of the form $\lambda_t \ge c_t > 0$ at each iteration t all fail to optimize the objective (5).)

Instead, we view the robust ERM problem (1) as a game between the x (minimizing) player and p (maximizing) player. Each player performs a variant of mirror descent (ascent), and we show how such an approach yields strong convergence guarantees, as well as good empirical performance. In particular, we show (for many suitable divergences f) that if ℓ_i is L-Lipschitz and $\mathcal{X}$ has radius bounded by R, then our procedure requires at most $O(\frac{R^2 L^2 + \rho}{\epsilon^2})$ iterations to achieve an ϵ-accurate solution to problem (1), which is comparable to the number of iterations required by SGD [23]. Our solution strategy builds off of similar algorithms due to Nemirovski et al. [23, Sec. 3] and Ben-Tal et al. [4], and more directly procedures developed by Clarkson et al. [10] for solving two-player convex games. Most directly relevant to our approach is that of Shalev-Shwartz and Wexler [26], which solves problem (1) under the assumption that $\mathcal{P}_{\rho,n} = \{p \in \mathbb{R}^n_+ : p^T \mathbb{1} = 1\}$ and that there is some x with perfect loss performance, that is, $\sum_{i=1}^n \ell_i(x) = 0$. We generalize these approaches to more challenging f-divergence-constrained problems, and, for the χ^2 divergence with $f(t) = \frac{1}{2}(t-1)^2$, develop efficient data structures that give a total run-time for solving problem (1) to ϵ-accuracy scaling as $O((\mathsf{Cost}(\mathsf{grad}) + \log n)\frac{R^2L^2+\rho}{\epsilon^2})$. Here $\mathsf{Cost}(\mathsf{grad})$ is the cost to compute the gradient of a single term $\nabla \ell_i(x)$ and perform a mirror descent step with x. Using SGD to solve the empirical minimization problem to ϵ-accuracy has run-time $O(\mathsf{Cost}(\mathsf{grad})\frac{R^2L^2}{\epsilon^2})$, so we see that we can achieve the guarantees (3)–(4) offered by the robust formulation (1) at little additional computational cost.

The remainder of the paper is organized as follows. We present our abstract algorithm in Section 2 and give guarantees on its performance in Section 3. In Section 4, we give efficient computational schemes for the case that $f(t) = \frac{1}{2}(t-1)^2$, presenting experiments in Section 5.

2 A bandit mirror descent algorithm for the minimax problem

Under the conditions that ℓ is convex and $\mathcal{X}$ is compact, standard results [7] show that there exists a saddle point $(x^\star, p^\star) \in \mathcal{X} \times \mathcal{P}_{\rho,n}$ for the robust problem (1) satisfying

$$\sup\left\{p^\top \ell(x^\star) \mid p \in \mathcal{P}_{\rho,n}\right\} \le p^{\star\top}\ell(x^\star) \le \inf\left\{p^{\star\top}\ell(x) \mid x \in \mathcal{X}\right\}.$$

We now describe a procedure for finding this saddle point by alternating a linear bandit-convex optimization procedure [8] for p and a stochastic mirror descent procedure for x. Our approach builds off of Nemirovski et al.'s [23] development of mirror descent for two-player stochastic games.

To describe our algorithm, we require a few standard tools. Let $\|\cdot\|_{\mathsf{x}}$ denote a norm on the space $\mathcal{X}$ with dual norm $\|y\|_{\mathsf{x},*} = \sup\{\langle x, y\rangle : \|x\| \le 1\}$, and let ψ_{x} be a differentiable strongly convex function on $\mathcal{X}$, meaning $\psi_{\mathsf{x}}(x+\Delta) \ge \psi_{\mathsf{x}}(x) + \nabla\psi_{\mathsf{x}}(x)^\top\Delta + \frac{1}{2}\|\Delta\|_{\mathsf{x}}^2$ for all Δ. Let ψ_{p} a differentiable strictly convex function on $\mathcal{P}_{\rho,n}$. For a differentiable convex function h, we define the Bregman divergence $B_h(x,y) = h(x) - h(y) - \langle \nabla h(y), x-y\rangle$ 0. The Fenchel conjugate ψ_{p}^* of ψ_{p} is

$$\psi_{\mathsf{p}}^*(s) := \sup_p\{\langle s,p\rangle - \psi_{\mathsf{p}}(p)\} \text{ and } \nabla\psi_{\mathsf{p}}^*(s) = \operatorname*{argmax}_p\{\langle s,p\rangle - \psi_{\mathsf{p}}(p)\}.$$

(ψ_{p}^* is differentiable because ψ_{p} is strongly convex [20, Chapter X].) We let $g_i(x) \in \partial\ell_i(x)$ be a particular subgradient selection.

With this notation in place, we now give our algorithm, which alternates between gradient ascent steps on p and subgradient descent steps on x. Roughly, we would like to alternate gradient ascent steps for p, $p_{t+1} \leftarrow p_t + \alpha_{\mathsf{p}}\ell(x_t)$, and descent steps $x_{t+1} \leftarrow x_t - \alpha_{\mathsf{x}} g_i(x_t)$ for x, where i is a random index drawn according to p_t. This procedure is inefficient—requiring time of order $n\mathsf{Cost}(\mathsf{grad})$ in each iteration—so that we use stochastic estimates of the loss vector $\ell(x_t)$ developed in the linear bandit literature [8] and variants of mirror descent to implement our algorithm.

Algorithm 1 Two-player Bandit Mirror Descent

Input: Stepsize $\alpha_{\mathsf{x}}, \alpha_{\mathsf{p}} > 0$, initialize: $x_1 \in \mathcal{X}$, $p_1 = \mathbb{1}/n$
for $t = 1, 2, \ldots, T$ **do**
  Sample $I_t \sim p_t$, that is, set $I_t = i$ with probability $p_{t,i}$
  Compute estimated loss for $i \in [n]$: $\widehat{\ell}_{t,i}(x) = \frac{\ell_i(x)}{p_{i,t}}\mathbf{1}\{I_t = i\}$
  Update p: $w_{t+1} \leftarrow \nabla\psi_{\mathsf{p}}^*(\nabla\psi_{\mathsf{p}}(p_t) + \alpha_{\mathsf{p}}\widehat{\ell}_t(x_t))$, $p_{t+1} \leftarrow \operatorname{argmin}_{p\in\mathcal{P}_{\rho,n}} B_{\psi_{\mathsf{p}}}(p, w_{t+1})$
  Update x: $y_{t+1} \leftarrow \nabla\psi_{\mathsf{x}}^*(\psi_{\mathsf{x}}(x_t) - \alpha_x g_{I_t}(x_t))$, $x_{t+1} \leftarrow \operatorname{argmin}_{x\in\mathcal{X}} B_{\psi_{\mathsf{x}}}(x, y_{t+1})$
end for

We specialize this general algorithm for specific choices of the divergence f and the functions ψ_{x} and ψ_{p} presently, first briefly discussing the algorithm. Note that in Step 5, the updates for p depend only on a single index $I_t \in \{1, \ldots, n\}$ (the vector $\widehat{\ell}(x_t)$ is 1-sparse), which, as long as the updates for p are efficiently computable, can yield substantial performance benefits.

3 Regret bounds

With our algorithm described, we now describe its convergence properties, specializing later to specific families of f-divergences. We begin with the following result on *pseudo*-regret, which (with minor modifications) is known [23, 10, 26]. We provide a proof for completeness in Appendix A.1.

Lemma 1. *Let the sequences x_t and p_t be generated by Algorithm 1. Define $\widehat{x}_T := \frac{1}{T}\sum_{t=1}^T x_t$ and $\widehat{p}_T := \frac{1}{T}\sum_{t=1}^T p_t$. Then for the saddle point $(x^\star, p^\star)$ we have*

$$T\mathbb{E}[p^{\star\top}\ell(\widehat{x}_T) - \widehat{p}_T^\top \ell(x^\star)] \le \underbrace{\frac{1}{\alpha_{\mathsf{x}}} B_{\psi_{\mathsf{x}}}(x^\star, x_1) + \frac{\alpha_{\mathsf{x}}}{2}\sum_{t=1}^T \mathbb{E}[\|g_{I_t}(x_t)\|_{\mathsf{x},*}^2]}_{\mathcal{T}_1:\ \textit{ERM regret}} + \underbrace{\sum_{t=1}^T \mathbb{E}[\widehat{\ell}_t(x_t)^\top (p^\star - p_t)]}_{\mathcal{T}_2:\ \textit{robust regret}}$$

where the expectation is taken over the random draws $I_t \sim p_t$. Moreover, $\mathbb{E}[\widehat{\ell}_t(x_t)^\top(p - p_t)] = \mathbb{E}[\ell(x_t)^\top(p - p_t)]$ for any vector p.

In the lemma, $\mathcal{T}_1$ is the standard regret when applying mirror descent to the ERM problem. In particular, if $B_{\psi_{\mathsf{x}}}(x^\star, x_1) \le R^2$ and $\ell_i(x)$ is L-Lipschitz, then choosing $\alpha_{\mathsf{x}} = \frac{R}{L}\sqrt{2/T}$ yields $\mathcal{T}_1 \le RL\sqrt{T}$. Because it is (relatively) easy to bound the term $\mathcal{T}_1$, the remainder of our arguments focus on bounding the the second term $\mathcal{T}_2$, which is the regret that comes as a consequence of the random sampling for the loss vector $\widehat{\ell}_t$. This regret depends strongly on the distance-generating function ψ_{p}. To the end of bounding $\mathcal{T}_2$, we use the following bound for the pseudo-regret of p, which is standard [9, Chapter 11], [8, Thm 5.3]. For completeness we outline the proof in Appendix A.2.

Lemma 2. *For any $p \in \mathcal{P}_{\rho,n}$, Algorithm 1 satisfies*

$$\sum_{t=1}^T \widehat{\ell}_t(x_t)^\top (p - p_t) \le \frac{B_{\psi_{\mathsf{p}}}(p, p_1)}{\alpha_{\mathsf{p}}} + \frac{1}{\alpha_{\mathsf{p}}}\sum_{t=1}^T B_{\psi_{\mathsf{p}}^*}\left(\nabla\psi_{\mathsf{p}}(p_t) + \alpha_{\mathsf{p}}\widehat{\ell}_t(x_t), \nabla\psi_{\mathsf{p}}(p_t)\right). \tag{6}$$

Lemma 2 shows that controlling the Bregman divergences $B_{\psi_{\mathsf{p}}}$ and $B_{\psi_{\mathsf{p}}^*}$ is sufficient to bound $\mathcal{T}_2$ in the basic regret bound of Lemma 1.

Now, we narrow our focus slightly to a specialized—but broad—family of divergences for which we can give more explicit results. For $k \in \mathbb{R}$, the Cressie-Read divergence [12] of order k is

$$f_k(t) = \frac{t^k - kt + k - 1}{k(k-1)}, \tag{7}$$

where $f_k(t) = \infty$ for $t < 0$, and for $k \in \{0, 1\}$ we define f_k by its limits as $k \to 0$ or 1 (we have $f_1(t) = t\log t - t + 1$ and $f_0(t) = -\log t + t - 1$). Inspecting expression (6), we might hope that careful choices of ψ_{p} could yield regret bounds that grow slowly with T and have small dependence on the sample size n. Indeed, this is the case, as we show in the sequel: for each divergence f_k, we may carefully choose ψ_{p} to achieve small regret. To prove our bounds, however, it is crucial that the importance sampling estimator $\widehat{\ell}_t$ has small variance, which in turn necessitates that $p_{t,i}$ is not too small. Generally, this means that in the update (Alg. 1, Line 5) to construct p_{t+1}, we choose $\psi(p)$ to grow quickly as $p_i \to 0$ (e.g. $|\frac{\partial}{\partial p_i}\psi_{\mathsf{p}}(p)| \to \infty$), but there is a tradeoff in that this may cause large Bregman divergence terms (6). In the coming sections, we explore this tradeoff for various k, providing regret bounds for each of the Cressie-Read divergences (7).

To control the $B_{\psi_{\mathsf{p}}^*}$ terms in the bound (6), we use the curvature of ψ_{p} (dually, smoothness of ψ_{p}^*) to show that $B_{\psi_{\mathsf{p}}^*}(u, v) \approx \sum(u_i - v_i)^2$. For this approximation to hold, we shift our loss functions based on the f-divergence. When $k \ge 2$, we assume that $\ell(x) \in [0,1]^n$. If $k < 2$, we instead apply Algorithm 1 with shifted losses $\ell'(x) = \ell(x) - \mathbb{1}$, so that $\ell'(x) \in [-1, 0]^n$. We call the method with ℓ' Algorithm 1', noting that $\widehat{\ell}_{t,i}(x_t) = \frac{\ell_i(x_t) - 1}{p_{t,i}}\mathbf{1}\{I_t = i\}$ in this case.

3.1 Power divergences when $k \notin \{0, 1\}$

For our first results, we prove a generic regret bound for Algorithm 1 when $k \notin \{0, 1\}$ by taking the distance-generating function $\psi_{\mathsf{p}}(p) = \frac{1}{k(k-1)}\sum_{i=1}^n p_i^k$, which is differentiable and strictly convex on $\mathbb{R}_+^n$. Before proceeding further, we first note that for $p \in \mathcal{P}_{\rho,n}$ and $p_1 = \frac{1}{n}\mathbb{1}$, we have

$$\begin{aligned} B_{\psi_{\mathsf{p}}}(p, p_1) &= \psi_{\mathsf{p}}(p) - \psi_{\mathsf{p}}(p_1) - \nabla\psi_{\mathsf{p}}(p_1)^\top(p - p_1) \\ &= \frac{n^{-k}}{k(k-1)}\sum_{i=1}^n \left\{(np_i)^k - knp_i + k - 1\right\} = n^{-k} D_f\left(p \| \mathbb{1}/n\right) \le n^{-k}\rho \end{aligned} \tag{8}$$

bounding the first term in expression (6). From Lemma 2, it remains to bound the Bregman divergence terms $B_{\psi_{\mathsf{p}}^*}$. Using smoothness of ψ_{p}^* in the positive orthant, we obtain the following bound.

Theorem 1. *Assume that $\ell(x) \in [0,1]^n$. For any real-valued $k \ge 2$ and any $p \in \mathcal{P}_{\rho,n}$, Algorithm 1 satisfies*

$$\sum_{t=1}^{T} \mathbb{E}[\ell(x_t)^\top (p - p_t)] = \sum_{t=1}^{T} \mathbb{E}[\widehat{\ell}_t(x_t)^\top (p - p_t)] \le \frac{n^{-k}\rho}{\alpha_{\mathsf{p}}} + \frac{\alpha_{\mathsf{p}}}{2} \sum_{t=1}^{T} \mathbb{E}\left[\sum_{i:p_{t,i}>0} p_{t,i}^{1-k}\right]. \tag{9}$$

For $k \le 2$ with $k \notin \{0,1\}$, an identical bound holds for Algorithm 1' with $\ell'(x) = \ell(x) - \mathbb{1}$.

See Appendix A.3 for the proof. We now use Theorem 1 to obtain concrete convergence guarantees for Cressie-Read divergences with parameter $k < 1$, giving sublinear (in T) regret bounds independent of n. In the corollary, whose proof we provide in Appendix A.4, we let $C_{k,\rho} = \frac{(1-k)(1-k\rho)}{-k}$, which is positive for $k < 0$.

Corollary 1. *For $k \in (-\infty, 0)$ and $\alpha_{\mathsf{p}} = C_{k,\rho}^{\frac{k-1}{2}} n^{-k} \sqrt{2\rho/T}$ Algorithm 1' with $\ell'(x) = \ell(x) - \mathbb{1} \in [-1,0]^n$ acheives the regret bound*

$$\sum_{t=1}^{T} \mathbb{E}[\ell(x_t)^\top (p - p_t)] = \sum_{t=1}^{T} \mathbb{E}[\widehat{\ell}_t(x_t)^\top (p - p_t)] \le \sqrt{2 C_{k,\rho}^{1-k} \rho T}.$$

For $k \in (0,1)$ and $\alpha_{\mathsf{p}} = n^{-k}\sqrt{2\rho/T}$, Algorithm 1' with $\ell'(x) = \ell(x) - \mathbb{1} \in [-1,0]^n$ acheives the regret bound

$$\sum_{t=1}^{T} \mathbb{E}[\ell(x_t)^\top (p - p_t)] = \sum_{t=1}^{T} \mathbb{E}[\widehat{\ell}_t(x_t)^\top (p - p_t)] \le \sqrt{2\rho T}.$$

It is worth noting that despite the robustification, the above regret is independent of n. In the special case that $k \in (0,1)$, Theorem 1 is the regret bound for the implicitly normalized forecaster of Audibert and Bubeck [1] (cf. [8, Ch 5.4]).

3.2 Regret bounds using the KL divergences ($k = 1$ and $k = 0$)

The choice $f_1(t) = t \log t - t + 1$ yields $D_f(P\|Q) = D_{\mathrm{kl}}(P\|Q)$, and in this case, we take $\psi_{\mathsf{p}}(p) = \sum_{i=1}^{n} p_i \log p_i$, which means that Algorithm 1 performs entropic gradient ascent. To control the divergence $B_{\psi_{\mathsf{p}}^*}$, we use the rescaled losses $\ell'(x) = \ell(x) - \mathbb{1}$ (as we have $k < 2$). Then we have the following bound, whose proof we provide in Appendix A.5.

Theorem 2. *Algorithm 1' with loss $\ell'(x) = \ell(x) - \mathbb{1}$ yields*

$$\sum_{t=1}^{T} \mathbb{E}[\ell(x_t)^\top (p - p_t)] = \sum_{t=1}^{T} \mathbb{E}[\widehat{\ell}_t(x_t)^\top (p - p_t)] \le \frac{\rho}{n\alpha_{\mathsf{p}}} + \frac{\alpha_{\mathsf{p}}}{2} nT. \tag{10}$$

In particular, when $\alpha_{\mathsf{p}} = \frac{1}{n}\sqrt{\frac{2\rho}{T}}$, we have $\sum_{t=1}^{T} \mathbb{E}[\ell(x_t)^\top (p - p_t)] \le \sqrt{2\rho T}$.

Using $k = 0$, so that $f_0(t) = -\log t + t - 1$, we obtain $D_f(P\|Q) = D_{\mathrm{kl}}(Q\|P)$, which results in a robustification technique identical to Owen's original empirical likelihood [24]. We again use the rescaled losses $\ell'(x) = \ell(x) - \mathbb{1}$, but in this scenario we use the proximal function $\psi_{\mathsf{p}}(p) = -\sum_{i=1}^{n} \log p_i$ in Algorithm 1'. Then we have the following regret bound (see Appendix A.6).

Theorem 3. *Algorithm 1' with loss $\ell'(x) = \ell(x) - \mathbb{1}$ yields*

$$\sum_{t=1}^{T} \mathbb{E}[\ell(x_t)^\top (p - p_t)] = \sum_{t=1}^{T} \mathbb{E}[\widehat{\ell}_t(x_t)^\top (p - p_t)] \le \frac{\rho}{\alpha_{\mathsf{p}}} + \frac{\alpha_{\mathsf{p}}}{2} T.$$

In particular, when $\alpha_{\mathsf{p}} = \sqrt{\frac{2\rho}{T}}$, we have $\sum_{t=1}^{T} \mathbb{E}[\ell(x_t)^\top (p - p_t)] \le \sqrt{2\rho T}$.

In both of these cases, the expected pseudo-regret of our robust gradient procedure is independent of n and grows as $\sqrt{T}$, which is essentially identical to that achieved by pure online gradient methods.

3.3 Power divergences ($k > 1$)

Corollary 1 provides convergence guarantees for power divergences f_k with $k < 1$, but says nothing about the case that $k > 1$; the choice $\psi_{\mathsf{p}}(p) = \frac{1}{k(k-1)} \sum_{i=1}^n p_i^k$ allows the individual probabilities $p_{t,i}$ to be too small, which can cause excess variance of $\widehat{\ell}$. To remedy this, we regularize the robust problem (1) by re-defining our robust empirical distributions set, taking

$$\mathcal{P}_{\rho,n,\delta} := \Big\{ p \in \mathbb{R}_+^n \mid p \geq \frac{\delta}{n}, \sum_{i=1}^n f(np_i) \leq \rho \Big\},$$

where we no longer constrain the weights p to satisfy $\mathbb{1}^\top p = 1$. Nonetheless, it is still possible to show that the guarantees (2) and (3) hold with $\mathcal{P}_{\rho,n,\delta}$ replacing $\mathcal{P}_{\rho,n}$. Indeed, we may give bounds for the pseudo-regret of the regularized problem with $\mathcal{P}_{\rho,n,\delta}$, where we apply Algorithm 1 with a slightly modified sampling strategy, drawing indices i according to the normalized distribution $p_t / \sum_{i=1}^n p_{t,i}$ and appropriately normalizing the loss estimate via

$$\widehat{\ell}_{t,i}(x_t) = \left(\sum_{i=1}^n p_{t,i} \right) \frac{\ell_i(x_t)}{p_{t,i}} \mathbf{1}\left\{ I_t = i \right\}.$$

This vector is still unbiased for $\ell(x_t)$. Define the constant $C_k := \max\left\{ t : f_k(t) \leq t \right\} \vee \frac{\rho}{n} < \infty$ (so $C_2 = 2 + \sqrt{3}$). With our choice $\psi_{\mathsf{p}}(p) = \frac{1}{k(k-1)} \sum_{i=1}^n p_i^k$ and for $\delta > 0$, we obtain the following result, whose proof we provide in Appendix A.7.

Theorem 4. *For $k \in [2, \infty)$, any $p \in \mathcal{P}_{\rho,n,\delta}$, Algorithm 1 with $\alpha_{\mathsf{p}} = n^{-k}\sqrt{\rho\delta^{k-1}/\left(4C_k^3 T\right)}$ yields*

$$\sum_{t=1}^T \mathbb{E}[\ell(x_t)^\top (p - p_t)] = \sum_{t=1}^T \mathbb{E}[\widehat{\ell}_t(x_t)^\top (p - p_t)] \leq 2C_k \sqrt{\rho C_k \delta^{1-k} T}$$

For $k \in (1, 2)$, assume that $\ell(x) \in [-1, 0]^n$. Then, Algorithm 1 gives identical bounds.

4 Efficient updates when $k = 2$

The previous section shows that Algorithm 1 with careful choice of ψ_{p} yields sublinear regret bounds. The projection step $p_{t+1} = \mathrm{argmin}_{p \in \mathcal{P}_{\rho,n,\delta}} B_{\psi_{\mathsf{p}}}(p, w_{t+1})$, however, can still take time linear in n despite the sparsity of $\widehat{\ell}(x_t)$ (see Appendix B for concrete updates for each of our cases). In this section, we show how to compute the bandit mirror descent update in Alg. 1, line 5, in time $O(\log n)$ time for $f_2(t) = \frac{1}{2}(t-1)^2$ and $\psi_{\mathsf{p}}(p) = \frac{1}{2}\sum_{i=1}^n p_i^2$. Building off of Duchi et al. [14], we use carefully designed balanced binary search trees (BSTs) to this end.

The Lagrangian for the update $p_{t+1} = \mathrm{argmin}_{p \in \mathcal{P}_{\rho,n,\delta}} B_{\psi_{\mathsf{p}}}(p, w_{t+1})$ (suppressing t) is

$$\mathcal{L}(p, \lambda, \theta) = B_{\psi_{\mathsf{p}}}(p, w) - \frac{\lambda}{n^2} \left(\rho - \sum_{i=1}^n f_2(np_i) \right) - \theta^\top \left(p - \frac{\delta}{n} \mathbb{1} \right)$$

where $\lambda \geq 0, \theta \in \mathbb{R}_+^n$. The KKT conditions imply $(1+\lambda)p = w + \frac{\lambda}{n}\mathbb{1} + \theta$, and strict complementarity yields

$$p(\lambda) = \left(\frac{1}{1+\lambda} w + \frac{\lambda}{1+\lambda}\frac{1}{n} - \frac{\delta}{n}\mathbb{1} \right)_+ + \frac{\delta}{n}\mathbb{1}, \tag{11}$$

where $p(\lambda) = \mathrm{argmin}_{p \in \mathcal{P}_{\rho,n,\delta}} \inf_{\theta \in \mathbb{R}_+^n} \mathcal{L}(p, \lambda, \theta)$. Substituting this into the Lagrangian, we obtain the concave dual objective

$$g(\lambda) := \sup_{\theta} \inf_{p \in \mathcal{P}_{\rho,n,\delta}} \mathcal{L}(p, \lambda, \theta) = B_{\psi_{\mathsf{p}}}(p(\lambda), w) - \lambda \left(\rho - \sum_{i=1}^n f_k(np_i(\lambda)) \right).$$

We can run a bisection search on the nondecreasing function $g'(\lambda)$ to find λ such that $g'(\lambda) = 0$. After algebraic manipulations, we have that

$$\frac{\partial}{\partial \lambda} g(\lambda) = g_1(\lambda) \sum_{i \in I(\lambda)} w_i^2 + g_2(\lambda) \sum_{i \in I(\lambda)} w_i + g_3(\lambda)|I(\lambda)| + \frac{(1-\delta)^2}{2n} - \frac{\rho}{n^2},$$

where $I(\lambda) := \{1 \le i \le n : w_i \ge \frac{\delta}{n} + (\frac{\delta}{n} - 1)\lambda\}$ and (see expression (18) in Appendix B.4)

$$g_1(\lambda) = \frac{1}{(1+\lambda)^2}, \;\; g_2(\lambda) = \frac{-2}{n(1+\lambda)^2}, \;\; g_3(\lambda) = \frac{1}{n^2(1+\lambda)^2} - \frac{(1-\delta)^2}{2n}.$$

To see that we can solve for λ^* that acheives $|g'(\lambda^*)| \le \epsilon$ in $O(\log n + \log \frac{1}{\epsilon})$ time, it suffices to evaluate $\sum_{i \in I(\lambda)} w_i^q$ for $q = 0, 1, 2$ in time $O(\log n)$. To this end, we store the w's in a balanced search tree (*e.g.*, red-black tree) keyed on the weights up to a multiplicative and an additive constant. A key ingredient in our implementation is that the BST stores in each node the sum of the appropriate powers of values in the left and right subtree [14]. See Appendix C for detailed pseudocode for all operations required in Algorithm 1: each subroutine (sampling $I_t \sim p_t$, updating w, computing λ^*, and updating $p(\lambda^*)$) require time $O(\log n)$ using standard BST operations.

5 Experiments

In this section, we present experimental results demonstrating the efficiency of our algorithm. We first compare our method with existing algorithms for solving the robust problem (1) on a synthetic dataset, then investigating the robust formulation on real datasets to show how the calibrated confidence guarantees behave in practice, especially in comparison to the ERM. We experiment on natural high dimensional datasets as well as those with many training examples.

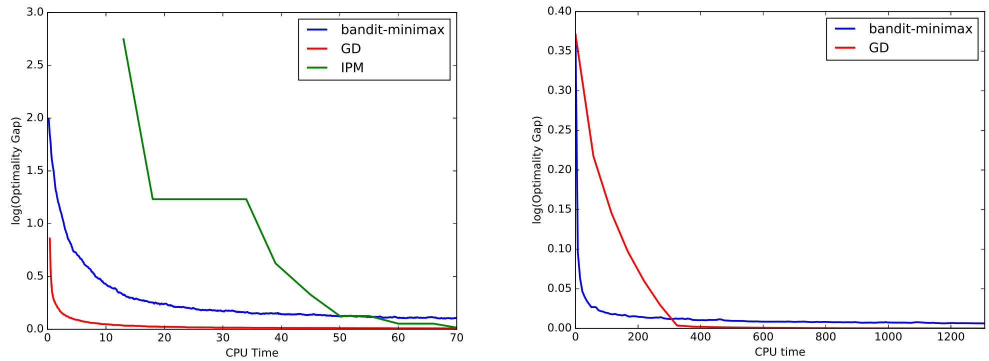
Our implementation uses the efficient updates outlined in Section 4. Throughout our experiments, we use the best tuned step sizes for all methods. For the first two experiments, we set $\rho = \chi^2_{1,.9}$ so that the resulting robust objective (1) will be a calibrated 95% upper confidence bound on the optimal population risk. For our last experiment, the asymptotic regime (3) fails to hold due to the high dimensional nature of the problem, so we choose $\rho = 50$ (somewhat arbitrarily, but other ρ give similar behavior). We take $\mathcal{X} = \left\{x \in \mathbb{R}^d : \|x\|_2 \le R\right\}$ for our experiments.

For the experiment with synthetic data, we compare our algorithm against two benchmark methods for solving the robust problem (1). The first is the interior point method for the dual reformulation (5) using the Gurobi solver [17]. The second is using gradient descent, viewing the robust formulation (1) as a minimization problem with the objective $x \mapsto \sup_{p \in \mathcal{P}_{\rho,n,\delta}} p^\top \ell(x)$. To efficiently compute the gradient, we bisect over the dual form (5) with respect to $\lambda \ge 0, \eta$. We use the best step sizes for both our proposed bandit-based algorithm and gradient descent.

To generate the data, we choose a true classifier $x^* \in \mathbb{R}^d$ and sample the feature vectors $a_i \overset{\text{iid}}{\sim} \mathsf{N}(0, I)$ for $i \in [n]$. We set the labels to be $b_i = \text{sign}(a_i^\top x^*)$ and flip them with probability 10%. We use the hinge loss $\ell_i(x) = \left(1 - b_i a_i^\top x\right)_+$ with $n = 2000$, $d = 500$ and $R = 10$ in our experiment. In Figure 1a, we plot the log optimality ratio (log of current objective value over optimal value) with respect to the runtime for the three algorithms. While the interior point method (IPM) obtains accurate solutions, it scales relatively poorly in n and d (the initial flat region in the plot is due to pre-computations for factorizing within the solver). Gradient descent performs quite well in this moderate sized example although each iteration takes time $\Omega(n)$.

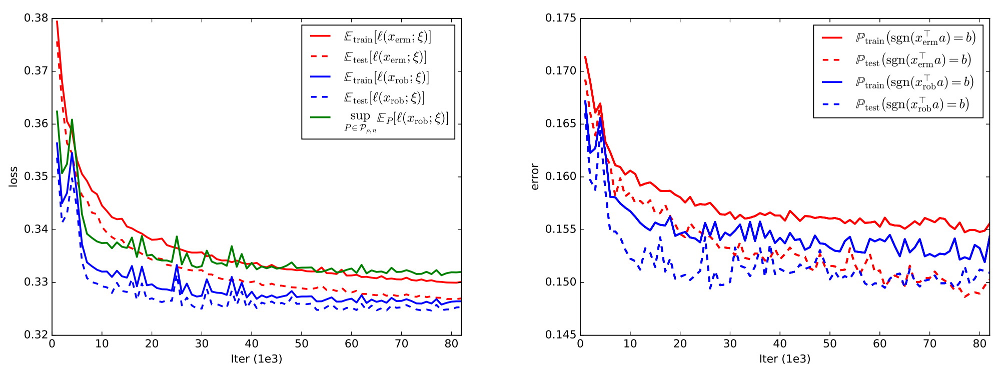
We also perform experiments on two datasets with larger n: the Adult dataset [22] and the Reuters RCV1 Corpus [21]. The Adult dataset has $n = 32{,}561$ training and 16,281 test examples with 123-dimensional features. We use binary logistic loss $\ell_i(x) = \log(1 + \exp(-b_i a_i^\top x))$ to classify whether the income level is greater than \$5K. For the Reuters RCV1 Corpus, our task is to classify whether a document belongs to the Corporate category. With $d = 47{,}236$ features, we randomly split the 804,410 examples into 723,969 training (90% of data) and 80,441 (10% of data) test examples. We use the hinge loss and solve the binary classification problem for the document type. To test the efficiency of our method in large scale settings, we plot the log ratio $\log \frac{R_n(x)}{R_n(x^*)}$, where $R_n(x) = \sup_{p \in \mathcal{P}_{\rho,n,\delta}} p^\top \ell(x)$, versus CPU time for our algorithm and gradient descent in Figure 1b. As is somewhat typical of stochastic gradient-based methods, our bandit-based optimization algorithm quickly obtains a solution with small optimality gap (about 2% relative error), while the gradient descent method eventually achieves better loss.

In Figures 2a–2d, we plot the loss value and the classification error compared with applying pure stochastic gradient descent to the standard empirical loss, plotting the confidence bound for the robust

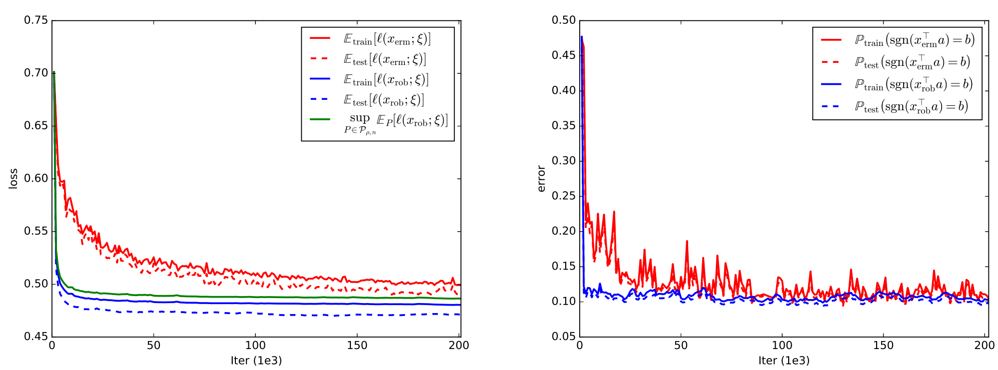


(a) Synthetic Data ($n = 2000, d = 500$) (b) Reuters Corpus ($n = 7.2 \cdot 10^5$, $d \approx 5 \cdot 10^4$)

Figure 1: Comparison of Solvers



(a) Adult: Logistic Loss (b) Adult: Classification Error



(c) Reuters: Hinge Loss (d) Reuters: Classification Error

Figure 2: Comparison with ERM

method as well. As the theory suggests [15, 13], the robust objective provides upper confidence bounds on the true risk (approximated by the average loss on the test sample).

References

[1] J.-Y. Audibert and S. Bubeck. Regret bounds and minimax policies under partial monitoring. In *Journal of Machine Learning Research*, pages 2635–2686, 2010.

[2] A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

[3] A. Ben-Tal, D. den Hertog, A. D. Waegenaere, B. Melenberg, and G. Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2): 341–357, 2013.

[4] A. Ben-Tal, E. Hazan, T. Koren, and S. Mannor. Oracle-based robust optimization via online learning. *Operations Research*, 63(3):628–638, 2015.

[5] J. Borwein, A. J. Guirao, P. Hájek, and J. Vanderwerff. Uniformly convex functions on Banach spaces. *Proceedings of the American Mathematical Society*, 137(3):1081–1091, 2009.

[6] S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification: a survey of some recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005.

[7] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[8] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

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

[10] K. Clarkson, E. Hazan, and D. Woodruff. Sublinear optimization for machine learning. *Journal of the Association for Computing Machinery*, 59(5), 2012.

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2001.

[12] N. Cressie and T. R. Read. Multinomial goodness-of-fit tests. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 440–464, 1984.

[13] J. C. Duchi and H. Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *arXiv:1610.02581 [stat.ML]*, 2016. URL `https://arxiv.org/abs/1610.02581`.

[14] J. C. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, 2008.

[15] J. C. Duchi, P. W. Glynn, and H. Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *arXiv:1610.03425 [stat.ML]*, 2016. URL `https://arxiv.org/abs/1610.03425`.

[16] S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, I: a generic algorithmic framework. *SIAM Journal on Optimization*, 22(4):1469–1492, 2012.

[17] I. Gurobi Optimization. Gurobi optimizer reference manual, 2015. URL `http://www.gurobi.com`.

[18] E. Hazan. The convex optimization approach to regret minimization. In *Optimization for Machine Learning*, chapter 10. MIT Press, 2012.

[19] E. Hazan and S. Kale. An optimal algorithm for stochastic strongly convex optimization. In *Proceedings of the Twenty Fourth Annual Conference on Computational Learning Theory*, 2011.

[20] J. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I & II*. Springer, New York, 1993.

[21] D. Lewis, Y. Yang, T. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[22] M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

[23] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[24] A. B. Owen. *Empirical likelihood*. CRC press, 2001.

[25] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[26] S. Shalev-Shwartz and Y. Wexler. Minimizing the maximal loss: How and why? In *Proceedings of the 32nd International Conference on Machine Learning*, 2016.

[27] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*, 2003.

A Proofs of Regret Bounds

A.1 Proof of Lemma 1

From convexity of the loss function $\ell(\cdot)$, we have

$$p^\top \ell(\widehat{x}_T) - \widehat{p}_T^\top \ell(x) \le \frac{1}{T}\sum_{t=1}^T p^\top \ell(x_t) - p_t^\top \ell(x) \le \frac{1}{T}\sum_{t=1}^T \left\{\ell(x_t)^\top (p - p_t) + p_t^\top g(x_t)(x - x_t)\right\} \tag{12}$$

where we have used $g(x_t) \in R^{n \times d}$ to denote the n-by-d matrix whose rows are $g_i(x_t)^\top$. Note that $\mathbb{E}[\widehat{\ell}_t(x_t)^\top (p - p_t) | I_1^{t-1}, x_1^t] = \ell(x_t)^\top (p - p_t)$, $\mathbb{E}[g_{I_t}(x_t)^\top (x - x_t) | I_1^{t-1}, x_1^t] = p_t^\top g(x_t)(x - x_t)$ since $I_t \sim p_t$ and p_t is $\sigma(I_1^{t-1}, x_1^t)$-measurable. Further, from the standard mirror descent result (*e.g.*, [23, Section 2.3]), we have

$$\sum_{t=1}^T \mathbb{E}[g_{I_t}(x_t)^\top (x - x_t)] \le \frac{1}{\alpha_x} B_{\psi_{\mathsf{x}}}(x^\star, x_1) + \frac{\alpha_x}{2}\sum_{t=1}^T \mathbb{E}\left\|g_{I_t}(x_t)\right\|_{\mathsf{x},*}^2.$$

Taking expectation in (12) and applying these facts, desired result follows.

A.2 Proof of Lemma 2

From Algorithm 1, we have

$$\begin{aligned}\alpha_{\mathsf{p}} \widehat{\ell}_t(x_t)^\top (p - p_t) &= \left(\nabla\psi_{\mathsf{p}}(w_{t+1}) - \nabla\psi_{\mathsf{p}}(x_t)\right)^\top (p - p_t) \\ &= B_{\psi_{\mathsf{p}}}(p, p_t) + B_{\psi_{\mathsf{p}}}(p_t, w_{t+1}) - B_{\psi_{\mathsf{p}}}(p, w_{t+1}).\end{aligned} \tag{13}$$

For any $p \in \mathcal{P}_{\rho,n}$, we have for all $p \in \mathcal{P}_{\rho,n}$,

$$B_{\psi_{\mathsf{p}}}(p, w_{t+1}) \ge B_{\psi_{\mathsf{p}}}(p, p_{t+1}) + B_{\psi_{\mathsf{p}}}(p_{t+1}, w_{t+1}) \;\equiv\; \left(\nabla\psi_{\mathsf{p}}(p) - \nabla\psi_{\mathsf{p}}(w_{t+1})\right)^\top (p - p_{t+1}) \ge 0.$$

The latter inequality is just the optimality condition for $p_{t+1} = \mathrm{argmin}_{p \in \mathcal{P}_{\rho,n}} B_{\psi_{\mathsf{p}}}(p, w_{t+1})$. Applying the first equality in (13) and summing for $t = 1, \ldots, T$, we obtain

$$\begin{aligned}\alpha_{\mathsf{p}} \sum_{t=1}^T \widehat{\ell}_t(x_t)^\top (p - p_t) &\le B_{\psi_{\mathsf{p}}}(p, p_1) - B_{\psi_{\mathsf{p}}}(p, p_{T+1}) + \sum_{t=1}^T \left(B_{\psi_{\mathsf{p}}}(p_t, w_{t+1}) - B_{\psi_{\mathsf{p}}}(p_{t+1}, w_{t+1})\right) \\ &\le B_{\psi_{\mathsf{p}}}(p, p_1) + \sum_{t=1}^T B_{\psi_{\mathsf{p}}}(p_t, w_{t+1}) \\ &= B_{\psi_{\mathsf{p}}}(p, p_1) + \sum_{t=1}^T B_{\psi_{\mathsf{p}}^*}(\nabla\psi_{\mathsf{p}}(w_{t+1}), \nabla\psi_{\mathsf{p}}(p_t)).\end{aligned}$$

Now, noting that $\nabla\psi_{\mathsf{p}}(w_{t+1}) = \nabla\psi_{\mathsf{p}}(p_t) + \alpha_{\mathsf{p}}\widehat{\ell}_t(x_t)$, we obtain the result.

A.3 Proof of Theorem 1

The conjugate of ψ_{p} is

$$\psi_{\mathsf{p}}^*(s) = \frac{1}{k}\left((k-1)s\right)_+^{k_*} + \infty \cdot \mathbf{1}\left\{s \le 0, k < 1\right\}.$$

From Taylor's theorem, we have

$$\begin{aligned}B_{\psi_{\mathsf{p}}^*}(u, v) &= \frac{1}{k}\sum_{i=1}^n \left(((k-1)u_i)_+^{k_*} - ((k-1)v_i)_+^{k_*}\right) - \sum_{i=1}^n ((k-1)v_i)_+^{k_*-1}(u_i - v_i) \\ &= \sum_{i=1}^n \left(\int_{v_i}^{u_i} ((k-1)t)_+^{k_*-1}\, dt - ((k-1)v_i)_+^{k_*-1}(u_i - v_i)\right) \\ &\le \frac{1}{2}\sum_{i=1}^n \max_{t \in [v_i, u_i]} ((k-1)t)_+^{k_*-2}(u_i - v_i)^2\end{aligned}$$

which gives the following useful lemma.

Lemma 3 (Bubeck and Cesa-Bianchi [8], Lemma 5.9)**.**

$$B_{\psi_{\mathsf{p}}^*}(u,v) \le \frac{1}{2}\sum_{i=1}^{n} \max_{t\in[v_i,u_i]} \left((k-1)t\right)_+^{k_*-2}(u_i-v_i)^2.$$

For later use, we define the conjugate $k_* = \frac{k}{k-1}$ and note that

$$k_* = \frac{k}{k-1} = \begin{cases} <1 & \text{if } k\in(-\infty,0) \\ <0 & \text{if } k\in(0,1) \\ >0 & \text{if } k\in(1,2) \\ <2 & \text{if } k\in(2,\infty) \end{cases}$$

Now, define $u := \nabla\psi_{\mathsf{p}}(p_t) + \alpha_{\mathsf{p}}\widehat{\ell}_t(x_t) = \frac{1}{k-1}p_t^{k-1} + \alpha_{\mathsf{p}}\widehat{\ell}_t(x_t)$ and $v := \nabla\psi_{\mathsf{p}}(p_t) = \frac{1}{k-1}p_t^{k-1}$, where p^{k-1} indicates the vector with each of its entries raised to the power $k-1$.

When $k\ge 2$, we have from Lemma 3 that

$$B_{\psi_{\mathsf{p}}^*}\left(\nabla\psi_{\mathsf{p}}(p_t) + \alpha_{\mathsf{p}}\widehat{\ell}_t(x_t), \nabla\psi_{\mathsf{p}}(p_t)\right) \le \frac{p_{t,I_t}^{2-k}}{2}\left(\alpha_{\mathsf{p}}\frac{\ell_{I_t}(x_t)}{p_{t,I_t}}\right)^2 \le \frac{\alpha_{\mathsf{p}}^2}{2}p_{t,I_t}^{-k} \tag{14}$$

where we have used that $\ell_i(x)\in[0,1]$. Substituting this in the bound (6) and taking expectations, we obtain the result by noting that $I_t\sim p_t$.

For $k<2$, note that since p, p_t are probability vectors and p_t is $\sigma(I_1^{t-1}, x_1^t)$-measurable, we have

$$\mathbb{E}[\widehat{\ell}_t(x_t)^\top(p-p_t)|I_1^{t-1},x_1^t] = \ell'(x_t)^\top(p-p_t) = (\ell(x_t)-\mathbb{1})^\top(p-p_t) = \ell(x_t)^\top(p-p_t) \tag{15}$$

from which the first equality of the theorem follows. Following the proof of Lemma 2 verbatim, we have the usual regret bound

$$\sum_{t=1}^{T}\widehat{\ell}_t(x_t)^\top(p-p_t) \le \frac{B_{\psi_{\mathsf{p}}}(p,p_1)}{\alpha_{\mathsf{p}}} + \frac{1}{\alpha_{\mathsf{p}}}\sum_{t=1}^{T} B_{\psi_{\mathsf{p}}^*}\left(\nabla\psi_{\mathsf{p}}(p_t)+\alpha_{\mathsf{p}}\widehat{\ell}_t(x_t), \nabla\psi_{\mathsf{p}}(p_t)\right) \tag{16}$$

where we now have $\widehat{\ell}_{t,i}(x_t) = \frac{\ell_i'(x_t)}{p_{t,i}}\mathbf{1}\{I_t=i\} = \frac{\ell_i(x_t)-1}{p_{t,i}}\mathbf{1}\{I_t=i\}$. Now, note that if $k\le 2$ with $k\notin\{0,1\}$, we have that $((k-1)s)_+^{k_*-2}$ is nondecreasing in s. Hence, we again obtain the bound (14) from Lemma 3.

A.4 Proof of Corollary 1

When $k\in(-\infty,0)$, the f-divergence constraint

$$\frac{1}{nk(k-1)}\sum_{i=1}^{n}\left\{(np_i)^k - k(np_i-1)-1\right\} \le \frac{\rho}{n}$$

implies that $-knp_i \le (np_i)^k - knp_i \le (1-k)(1-k\rho)$ and hence $p_i\le\frac{C_k}{n}$. Using this to bound the sum in (9), we get

$$\sum_{t=1}^{T}\mathbb{E}[\widehat{\ell}_t(x_t)^\top(p-p_t)] \le \frac{n^{-k}\rho}{\alpha_{\mathsf{p}}} + \frac{\alpha_{\mathsf{p}}}{2}Tn^kC_k^{1-k}.$$

Minimizing with respect to $\alpha_{\mathsf{p}}>0$ gives the first result. When $k\in(0,1)$, we use Holder inequality with $p=\frac{1}{1-k}>1$ and $q=\frac{1}{k}>1$:

$$\sum_{i=1}^{n}p_{t,i}^{1-k} \le \left(\sum_{i=1}^{n}(p_{t,i}^{1-k})^{\frac{1}{1-k}}\right)^{1-k}\left(\sum_{i=1}^{n}1\right)^{k} = n^k.$$

Applying this bound in (9) and minimizing with respect to α_{p}, result follows.

A.5 Proof of Theorem 2

Proceeding as in Section A.3 for $k \le 2$, we obtain the regret bound (16). First, note that $B_{\psi_{\mathsf{p}}}(p, p_1) = \sum_{i=1}^n p_i \log np_i \le \frac{\rho}{n}$. We now bound $B_{\psi_{\mathsf{p}}^*}\left(\nabla\psi_{\mathsf{p}}(p_t) + \alpha_{\mathsf{p}}\widehat{\ell}_t(x_t), \nabla\psi_{\mathsf{p}}(p_t)\right)$. Using $\exp(-x) - 1 + x \le \frac{x^2}{2}$ for $x \ge 0$, we have

$$
\begin{aligned}
&B_{\psi_{\mathsf{p}}^*}\left(\log p_t + \mathbb{1} + \alpha_{\mathsf{p}}\widehat{\ell}_t(x_t), \log p_t + \mathbb{1}\right) \\
&= \sum_{i \ne I_t} \exp(\log p_{t,i}) + \exp(\log p_{t,I_t} + \alpha_{\mathsf{p}}\widehat{\ell}_{t,I_t}(x_t)) - \sum_{i=1}^n \exp(\log p_{t,i}) - \exp(\log p_{t,I_t})\alpha_{\mathsf{p}}\widehat{\ell}_{t,I_t}(x_t) \\
&= p_{t,I_t}\left\{\exp(\alpha_{\mathsf{p}}\widehat{\ell}_{t,I_t}(x_t)) - 1 - \alpha_{\mathsf{p}}\widehat{\ell}_{t,I_t}(x_t)\right\} \le \frac{1}{2}p_{t,I_t}\widehat{\ell}_{t,I_t}(x_t)^2 = \frac{(\ell_{I_t}(x_t) - 1)^2}{2p_{t,I_t}}
\end{aligned}
$$

where we used $x = -\alpha_{\mathsf{p}}\widehat{\ell}_{t,I_t}(x_t) \ge 0$. Plugging the above observations into (6) and taking expectations, we obtain

$$
\sum_{t=1}^T \mathbb{E}[\widehat{\ell}_t(x_t)^\top(p - p_t)] \le \frac{\rho}{n\alpha_{\mathsf{p}}} + \frac{\alpha_{\mathsf{p}}}{2}\sum_{t=1}^T \mathbb{E}\left[\sum_{i=1}^n (\ell_i(x_t) - 1)^2\right].
$$

Bounding $(\ell_i(x_t) - 1)^2 \le 1$, the first claim follows. Optimizing the bound with respect to $\alpha_{\mathsf{p}} > 0$, we obtain the second result.

A.6 Proof of Theorem 3

As in Section A.3, the first equality and the interim regret bound follows from (15), (16). Now, note that $B_{\psi_{\mathsf{p}}}(p, p_1) = -\sum_{i=1}^n(\log(np_i) - np_i + 1) \le \rho$ to bound the first term. Next, we use $x - \log(1 + x) \le \frac{x^2}{2}$ for $x \ge 0$ to get

$$
\begin{aligned}
B_{\psi_{\mathsf{p}}^*}\left(\nabla\psi_{\mathsf{p}}(p_t) + \alpha_{\mathsf{p}}\widehat{\ell}_t(x_t), \nabla\psi_{\mathsf{p}}(p_t)\right) &= B_{\psi_{\mathsf{p}}^*}\left(-\frac{1}{p_t} + \alpha_{\mathsf{p}}\widehat{\ell}_t(x_t), -\frac{1}{p_t}\right) \\
&= -\log\left(1 - \alpha_{\mathsf{p}}\ell'_{I_t}(x_t)\right) - \alpha_{\mathsf{p}}\ell'_{I_t}(x_t) \le \frac{\alpha_{\mathsf{p}}^2\ell'_{I_t}(x_t)^2}{2}
\end{aligned}
$$

where we have used $x = -\alpha_{\mathsf{p}}\ell'_{I_t}(x_t) \ge 0$ and $\ell'_{I_t}(x_t) \in [-1, 0]$. Plugging these into the bound (6) and taking expectations, we have

$$
\sum_{t=1}^T \mathbb{E}[\widehat{\ell}_t(x_t)^\top(p - p_t)] \le \frac{\rho}{\alpha_{\mathsf{p}}} + \frac{\alpha_{\mathsf{p}}}{2}\sum_{t=1}^T\sum_{i=1}^n p_{t,i}(\ell_i(x_t) - 1)^2.
$$

Bounding $(\ell_i(x_t) - 1)^2 \le 1$, the first claim follows. Minimizing with respect to α_{p} gives the final claim.

A.7 Proof of Theorem 4

For $k \in [2, \infty)$, we proceed identically as in Section A.3 to obtain

$$
\sum_{t=1}^T \mathbb{E}[\ell(x_t)^\top(p - p_t)] = \sum_{t=1}^T \mathbb{E}[\widehat{\ell}_t(x_t)^\top(p - p_t)] \le \frac{B_{\psi_{\mathsf{p}}}(p_t, p_1)}{\alpha_{\mathsf{p}}} + \frac{\alpha_{\mathsf{p}}}{2}\sum_{t=1}^T \mathbb{E}\left[\left(\sum_{i=1}^n p_{t,i}\right)^3 \sum_{i: p_{t,i} > 0} p_{t,i}^{1-k}\right]
$$

where the extra summation term appeared since p_t's are no longer normalized. We note that $B_{\psi_{\mathsf{p}}}(p_t, p_1) \le n^{-k}\rho$ since (8) still holds.

From the definition of $C_k = \max\{t : f_k(t) \le t\} \vee \frac{\rho}{n}$, we have

$$
\sum_{i=1}^n np_i \le \sum_{i: np_i \le C_k} np_i + \sum_{i: np_i > C_k} f(np_i) \le nC_k + \rho \le 2nC_k
$$

for all $p \in \mathcal{P}_{\rho,n,\delta}$. Hence, it follows that

$$\sum_{t=1}^{T} \mathbb{E}[\widehat{\ell}_t(x_t)^\top (p - p_t)] \le \frac{n^{-k}\rho}{\alpha_{\mathsf{p}}} + 8\alpha_{\mathsf{p}} T C_k^3 \delta^{1-k} n^k.$$

Minimizing with respect to α_{p}, we obtain the first result.

When $k \in (1, 2]$, we proceed identically and use the fact that $k_* \ge 2$ and $\ell \in [-1, 0]$ in Lemma 3. Plugging this into the bound (6) and taking expectations, we obtain the second claim by following identical steps as in the case $k \ge 2$.

B Updates for p

In this section, we will explicitly write down the computations required for mirror descent updates in $p \in \mathcal{P}_{\rho,n}$. The updates for p is

$$p_{t+1} := \operatorname*{argmin}_{p \in \mathcal{P}_{\rho,n}} B_{\psi_{\mathsf{p}}}(p, w_{t+1}) \tag{17}$$

where $w_{t+1} = \nabla \psi_{\mathsf{p}}(p_t) + \alpha \widehat{\ell}_t(x_t)$. In the following, we omit subscripts for ease of notation. Note that for $k \le 1$, since $\|\nabla \psi_{\mathsf{p}}(p)\| \to \infty$ as $p_i \to 0$ for any $1 \le i \le n$, we can ignore the nonnegativity constraint in (17).

B.1 Power divergence for $k \in (-\infty, 1) \setminus \{0\}$

Writing down the Lagrangian for the optimization problem (17) with $\psi_{\mathsf{p}}(p) = \frac{1}{k(k-1)} \sum_{i=1}^n p_i^k$, we have

$$\begin{aligned}\mathcal{L}(p, \eta, \lambda) =& \frac{1}{k(k-1)} \sum_{i=1}^n (p_i^k - w_i^k) - \frac{1}{k-1} \sum_{i=1}^n w_i^{k-1}(p_i - w_i) \\ & - \eta(p^\top \mathbb{1} - 1) - n^{-k}\lambda \left(\rho - \frac{1}{k(k-1)} \sum_{i=1}^n ((np_i)^k - 1) \right)\end{aligned}$$

where $\eta \in \mathbb{R}^n$ and $\lambda \ge 0$. In any case, the first order conditions for p yield

$$(1 + \lambda) p^{k-1} = w^{k-1} + (k-1)\eta \mathbb{1}.$$

Plugging this in the constraint f-divergence constraint $\sum_{i=1}^n p_i^k \le n^{-k}(k(k-1)\rho + n)$ and using strict complementarity, we have

$$\lambda(\eta) = \left(\left(\frac{n^k}{k(k-1)\rho + n} \right)^{\frac{1}{k_*}} \left\| w^{k-1} + (k-1)\eta \mathbb{1} \right\|_{k_*} - 1 \right)_+ .$$

Plugging this in the Lagrangian

$$\mathcal{L}(\eta) = \min_{\lambda \ge 0} \max_{p \in \mathcal{P}_{\rho,n}} \mathcal{L}(p, \eta, \lambda) = B_{\psi_{\mathsf{p}}}(p(\eta), w) - \eta(p(\eta)^\top \mathbb{1} - 1)$$

where $p(\eta) = (1 + \lambda(\eta))^{1-k_*}(w^{k-1} + (k-1)\eta\mathbb{1})^{k_*-1}$. Now, it remains to minimize $\mathcal{L}(\eta)$. Noting that $\mathcal{L}(\eta)$ is a concave function, the derivative $\frac{d}{d\eta}\mathcal{L}(\eta)$ is an nondecreasing function. Hence, we can run a bisection search to find η such that $\frac{d}{d\eta}\eta = 0$. To this end, compute

$$\begin{aligned}\frac{d}{d\eta}\mathcal{L}(\eta) =& (1 + \lambda(\eta))^{1-k_*} \left(\frac{\lambda'(\eta)}{k-1} - 1 \right) \sum_{i=1}^n (w_i + (k-1)\eta)^{k_*-1} \\ & - (1 + \lambda(\eta))^{2-k_*} \frac{\lambda'(\eta)}{k-1} \sum_{i=1}^n w_i^{k-1}(w_i^{k-1} + (k-1)\eta)^{k_*-2} \\ & - \eta\lambda'(\eta)(1 + \lambda(\eta))^{2-k_*} \sum_{i=1}^n (w_i^{k-1} + (k-1)\eta)^{k_*-2}\end{aligned}$$

where

$$\lambda'(\eta) = \begin{cases} (k-1)\left(\frac{n^k}{k(k-1)\rho+n}\right)^{\frac{1}{k_*}} \left\|w^{k-1} + (k-1)\eta\mathbb{1}\right\|_{k_*}^{1-k_*} \sum_{i=1}^n (w_i^{k-1} + (k-1)\eta)^{k_*-1} & \text{if } \lambda(\eta) \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Since evaluating $\frac{d}{d\eta}\mathcal{L}(\eta)$ takes $O(n)$ time, the bisection on η will find a ϵ-accurate solution in $O(n \log \frac{1}{\epsilon})$ time. Using this optimal η to to compute $p(\eta)$ takes another $O(n)$ time.

B.2 KL divergence ($k = 1$)

Lagrangian for the optimization problem (17) with $\psi_{\mathsf{p}}(p) = \sum_{i=1}^n p_i \log p_i$ is

$$\mathcal{L}(p, \eta, \lambda) = \sum_{i=1}^n p_i \log \frac{p_i}{w_i} - \eta(p^\top \mathbb{1} - 1) - \frac{\lambda}{n}\left(\rho - \sum_{i=1}^n np_i \log(np_i)\right).$$

The first order conditions for p yield $p_i = w_i^{\frac{1}{1+\lambda}} n^{-\frac{\lambda}{1+\lambda}} \exp\left(\frac{\eta}{1+\lambda}\right)$ and from $p^\top \mathbb{1} = 1$, it follows that $p_i = w_i^{\frac{1}{1+\lambda}} / \sum_{i=1}^n w_i^{\frac{1}{1+\lambda}}$. Plugging this back into the Lagrangian, we have

$$\mathcal{L}(\lambda) = \min_\eta \max_{p \in \mathcal{P}_{\rho,n}} \mathcal{L}(p, \lambda, \eta) = \lambda\left(\log n - \frac{\rho}{n}\right) - \alpha\widehat{\ell}_{I_t}(x_t) - (1+\lambda)\log \sum_{i=1}^n w_i^{\frac{1}{1+\lambda}}.$$

Taking derivatives, we get

$$\frac{d}{d\lambda}\mathcal{L}(\lambda) = \log n - \frac{\rho}{n} - \log \sum_{i=1}^n w_i^{\frac{1}{1+\lambda}} - \frac{\sum_{i=1}^n w_i^{-\frac{\lambda}{1+\lambda}}}{\sum_{i=1}^n w_i^{\frac{1}{1+\lambda}}}$$

which can be computed in $O(n)$ flops. Since $\mathcal{L}(\lambda)$ is concave, $\lambda \geq 0$ such that $\frac{d}{d\lambda}\mathcal{L}(\lambda) = 0$ can be found to ϵ-accuracy in $O(n \log \frac{1}{\epsilon})$. Then, the update $p(\eta)$ takes $O(n)$ to compute.

B.3 EL divergence ($k = 0$)

Lagrangian for the optimization problem (17) with $\psi_{\mathsf{p}}(p) = -\sum_{i=1}^n \log p_i$ is

$$\mathcal{L}(p, \eta, \lambda) = -\sum_{i=1}^n \log \frac{p_i}{w_i} - \eta(p^\top \mathbb{1} - 1) - \lambda\left(\rho + \sum_{i=1}^n \log(np_i)\right).$$

The first order conditions for p yield $p_i = (1+\lambda)(\frac{1}{w_i} - \eta)^{-1}$. Plugging this into the divergence constraint and using strict complementarity, we have

$$\lambda(\eta) = \left(\exp\left(\frac{1}{n}\sum_{i=1}^n \log\left(\frac{1}{nw_i} - \frac{\eta}{n}\right) - \frac{\rho}{n}\right) - 1\right)_+.$$

Then, it suffices to solve

$$\mathcal{L}(\eta) = \min_{\lambda \geq 0} \max_{p \in \mathcal{P}_{\rho,n}} \mathcal{L}(p, \lambda, \eta) = \sum_{i=1}^n p_i(\eta) \log \frac{p_i(\eta)}{w_i} - \eta(p(\eta)^\top \mathbb{1} - 1).$$

From concavity, we can run bisection search on the monotone function $\frac{d}{d\eta}\mathcal{L}(\eta)$ to find its zero. To this end, compute

$$\frac{d}{d\eta}\mathcal{L}(\eta) = \sum_{i=1}^n \left\{p_i'(\eta)\left(\log \frac{p_i(\eta)}{w_i} - \eta - 1\right) + p_i(\eta)\right\} + 1$$

where

$$p_i'(\eta) = (1 + \lambda(\eta))\left(\frac{1}{w_i} - \eta\right)^{-2} + \lambda'(\eta)\left(\frac{1}{w_i} - \eta\right)^{-1}$$

$$\lambda'(\eta) = \begin{cases} -\frac{1}{n}\exp\left(\frac{1}{n}\sum_{i=1}^n \log\left(\frac{1}{nw_i} - \frac{\eta}{n}\right) - \frac{\rho}{n}\right)\sum_{i=1}^n \left(\frac{1}{w_i} - \eta\right)^{-1} & \text{if } \lambda(\eta) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Hence, the update $p(\eta)$ can be computed in $O(n)$ flops.

B.4 Power divergences ($k > 1$)

After some calculations, we have that

$$
\begin{aligned}
g'(\lambda) =& \frac{\partial}{\partial \lambda} B_{\psi_{\mathsf{p}}}(p(\lambda), w) + \lambda \frac{\partial}{\partial \lambda} \sum_{i=1}^{n} f_k(np_i(\lambda)) \\
=& \left\{ \frac{n^{k+1}\lambda}{(k-1)^2} - \frac{n}{k-1} + \lambda(\lambda-1)\frac{n^{2k+1}}{(k-1)^2} \right\} \left(1+n^k\lambda\right)^{k_*-1} \sum_{i \in I(\lambda)} (w_i^{k-1} + n\lambda)^{k_*-1} \\
&+ \left\{ \frac{n\lambda}{(k-1)^2} + \lambda(\lambda-1)\frac{n^{k+2}}{(k-1)^2} \right\} \left(1+n^k\lambda\right)^{k_*} \sum_{i \in I(\lambda)} (w_i^{k-1} + n\lambda)^{k_*-2} \\
&+ \frac{n^k}{k(k-1)} \left(1+n^k\lambda\right)^{k_*} \sum_{i \in I(\lambda)} w_i^{k-1}(w_i^{k-1} + n\lambda)^{k_*} \\
&+ \left\{ \lambda(\lambda-1)\frac{n^{2k+1}}{(k-1)^2} - \frac{n^{2k}\lambda}{(k-1)^2} \right\} \left(1+n^k\lambda\right)^{k_*-1} \sum_{i \in I(\lambda)} w_i^{k-1}(w_i^{k-1} + n\lambda)^{k_*-1} \\
&+ \left\{ \lambda(\lambda-1)\frac{n^{k+2}}{(k-1)^2} - \frac{n^{k+1}\lambda}{(k-1)^2} \right\} \left(1+n^k\lambda\right)^{k_*} \sum_{i \in I(\lambda)} w_i^{k-1}(w_i^{k-1} + n\lambda)^{k_*-2} \\
&- \frac{\delta^k - k\delta}{k(k-1)}|I(\lambda)| - \rho + \frac{n\lambda}{k} + \frac{n(\delta^k - k\delta)}{k(k-1)}
\end{aligned}
$$

where

$$
I(\lambda) = \left\{ 1 \le i \le n : w_i^{k-1} \ge \left(\frac{\delta}{n}\right)^{k-1} (1+n^k\lambda) - \lambda n \right\}.
$$

Hence, we can run bisection search on $\lambda \ge 0$ to find the zero of the monotone function $\frac{\partial}{\partial \lambda}\mathcal{L}(\lambda)$ as before.

When $k = 2$, under the change of variables $\lambda = n^2\lambda$, we have

$$
\begin{aligned}
\frac{\partial}{\partial \lambda} g(\lambda) &= \frac{1}{(1+\lambda)^2} \sum_{i \in I(\lambda)} \left(w_i - \frac{1}{n}\right)^2 - \frac{\rho}{n^2} + \frac{(1-\delta)^2}{2n^2}(n - |I(\lambda)|) \\
&= \frac{1}{(1+\lambda)^2} \sum_{i \in I(\lambda)} w_i^2 - \frac{2}{n(1+\lambda)^2} \sum_{i \in I(\lambda)} w_i \\
&\quad + \left(\frac{1}{n^2(1+\lambda)^2} - \frac{(1-\delta)^2}{2n^2} \right) |I(\lambda)| + \frac{(1-\delta)^2}{2n} - \frac{\rho}{n^2}
\end{aligned}
\tag{18}
$$

Making the additional change of variables $\alpha = \lambda/(1+\lambda)$ and $I(\alpha) = \left\{ i : (1-\alpha)w_i + \alpha/n \ge \frac{\delta}{n} \right\}$, we have

$$
\begin{aligned}
\frac{\partial}{\partial \alpha} g(\alpha) = \frac{1}{2} \sum_{i \in I(\alpha)} w_i^2 - \frac{1}{n} \sum_{i \in I(\alpha)} w_i + \frac{1}{2n^2(1-\alpha)^2} \left((1-\alpha)^2 - (1-\delta)^2\right) |I(\alpha)| \\
+ \frac{1}{2n^2(1-\alpha)^2}(n(1-\delta)^2 - 2\rho),
\end{aligned}
\tag{19}
$$

which is non-increasing in $\alpha \in [0, 1]$.

C Procedures for Efficient Updates when $k = 2$

We detail the operations involving the balanced binary search tree (BST) required for Algorithm 1. The weights w are stored up to multiplicative and additive factors *mult* and *addi*. Each node in the

BST stores the following variables:

$$\begin{aligned}
i &= \text{index in } \{1,\ldots,n\} \text{ of node} \\
\text{left} &= \text{pointer to the left child. } \emptyset \text{ if empty (NULL)} \\
\text{right} &= \text{pointer to the right child. } \emptyset \text{ if empty (NULL)} \\
w &= \text{weight, stored up to multiplicative and additive factors (\textit{mult} and \textit{addi})} \\
N_l &= \text{number of weights in the left subtree (smaller weights)} \\
N_r &= \text{number of weights in the right subtree (bigger weights)} \\
S_l &= \text{sum of weights in the left subtree (smaller weights)} \\
S_r &= \text{sum of weights in the right subtree (bigger weights)} \\
S_l^2 &= \text{sum of squared weights in the left subtree (smaller weights)} \\
S_r^2 &= \text{sum of squared weights in the right subtree (bigger weights)}
\end{aligned}$$

By computing $1 + N_l + N_r$ at the root node, the number of elements in the BST is available in constant time.

We first give the pseudo-code for the sampling procedure used in Line 1.3 of Algorithm 1. *Sample(tree)* samples a node from the given tree with probabilities proportional to the weights of the nodes. At any given node, the procedure decides whether to stay at the current node or recurse down the tree by tossing a coin proportional to the current weight w (stay) and the sum of weights s_l (go left) and s_r (go right). The algorithm returns the node if the coin flip results in a "stay" decision or it reaches a leaf node. By virtue of this recursive strategy, the sampling procedure requires $O(\log n)$.

Algorithm 2 Sample I_t

$coin \leftarrow$ Uniform(0,1)
node $\leftarrow root$
while node is not a leaf **do**
    if $coin < \frac{1}{1+\text{node}.N_l+\text{node}.N_r}$ **then**
        return node
    else if $coin < (1 + \text{node}.N_l)/(1 + \text{node}.N_l + \text{node}.N_r)$ **then**
        node $\leftarrow$ node.left
    else
        node $\leftarrow$ node.right
    end if
end while
return node

Next, we briefly outline the procedure for updating the sampled node with index I_t from p_t to w_{t+1}. Using the standard BST operations `Remove` and `Insert`, this step requires time $O(\log n)$. For example, a red-black tree uses subtree rotations to update and maintain the values $N_l, N_r, S_l, S_r, S_l^2, S_r^2$ along with the weights in logarithmic time [11]. See Duchi et al. [14] for explicitly updates when storing subtree weights and counts, as in our case.

Algorithm 3 Update w

Input: $p_{t,I_t}, w_{t,I_t}, I_t$
`Remove`(p_{t,I_t}, I_t), `Insert`(w_{t,I_t}, I_t)
return $root$

We next give a procedure that computes an ϵ-accurate solution to $\frac{\partial}{\partial\alpha} g(\alpha) = 0$ as in expression (19). We first bisect on the nodes to find the node with its weight at the optimal threshold. Then, we bisect on α to compute the exact value. Since the algorithm proceeds in two bisection steps, it only takes $O(\log n + \log \frac{1}{\epsilon})$ time.

Algorithm 4 Compute α^*

1: node $= root$, node$_r$, node$_l = \emptyset$
2: $c_{num}, c_{sum}, c_{sum^2} = 0$, $\ell_{num}, \ell_{sum}, \ell_{sum^2} = 0$
3: **while** true **do**
4:     $w \leftarrow$ node.w, $\alpha \leftarrow (\delta - nw)/(1 - nw)$
5:     $g(\alpha) \leftarrow \frac{1}{2}(c_{sum^2} + w^2 + \text{node}.S_r^2) - \frac{1}{n}(c_{sum} + w + \text{node}.S_r)$
6:         $+ \frac{1}{2n^2(1-\alpha)^2}((1-\alpha)^2 - (1-\delta)^2)(c_{num} + 1 + \text{node}.N_r) + \frac{1}{2n^2(1-\alpha)^2}(n(1-\delta)^2 - 2\rho)$
7:     **if** $g(\alpha) < 0$ **then** // too small, increase α
8:         node$_r \leftarrow$ node
9:         **if** node.right $= \emptyset$ **then** break
10:         **end if**
11:         node $\leftarrow$ node.right
12:     **else** // too big, decrease α
13:         node$_l \leftarrow$ node
14:         $c_{num} \leftarrow c_{num} + 1 + \text{node}.N_r$, $c_{sum} \leftarrow c_{sum} + \text{node}.w + \text{node}.S_r$
15:         $c_{sum^2} \leftarrow c_{sum^2} + \text{node}.w^2 + \text{node}.S_r^2$
16:         $\ell_{num} \leftarrow c_{num},\ \ell_{sum} \leftarrow c_{sum},\ \ell_{sum^2} \leftarrow c_{sum^2}$
17:         **if** node.left $= \emptyset$ **then** break
18:         **end if**
19:         node $\leftarrow$ node.left
20:     **end if**
21: **end while**
22: **if** node$_l \neq \emptyset$ **then**
23:     $c_{num} = \ell_{num},\ c_{sum} = \ell_{sum},\ c_{sum^2} = \ell_{sum^2}$
24: **end if**
25: $u \leftarrow 1$, $l \leftarrow 0$, $\alpha \leftarrow .5$
26: **while** $u - l > \epsilon$ **do**
27:     **if** $g(\alpha, \ell) < 0$ **then**
28:         $u \leftarrow \alpha$
29:     **else**
30:         $l \leftarrow \alpha$
31:     **end if**
32: **end while**
33: Update mult $\leftarrow (1-\alpha)$mult, addi $\leftarrow (1-\alpha) *$ addi $+ \alpha/n$
34: **return** α

In Line 4.27, we used $g(\alpha, \ell)$ to denote $g(\alpha)$ as computed with $\ell_{num}, \ell_{sum}, \ell_{sum^2}$ as the relevant sums.

Provided $\lambda^* = 1/(1-\alpha^*)$, Algorithm 5 gives a procedure for updating the tree to $p(\lambda^*)$ in $O(\log n)$ time. By virtue of the updates (11), we have $p_i(\lambda) \geq \frac{\delta}{n}$ for $i \neq I_t$ since $w_i \geq \frac{\delta}{n}$. Hence, the only potential truncation is for index I_t, which takes $O(\log n)$ time by removing and reinserting the node into the tree.

Algorithm 5 Update p

Input: λ^*, w_{t,I_t}, I_t
if $w_{t,I_t} < \frac{\delta}{n}$ **then**
    // If modified weight was too low, truncate.
    `Remove`(w_{t,I_t}, I_t), `Insert`($\frac{\delta}{n}, I_t$)
end if