Adaptive Sampling Probabilities for Non-Smooth Optimization

Hongseok Namkoong [1] **Aman Sinha** [2] **Steve Yadlowsky** [2] **John C. Duchi** [2 3]

Abstract

Standard forms of coordinate and stochastic gradient methods do not adapt to structure in data; their good behavior under random sampling is predicated on uniformity in data. When gradients in certain blocks of features (for coordinate descent) or examples (for SGD) are larger than others, there is a natural structure that can be exploited for quicker convergence. Yet adaptive variants often suffer nontrivial computational overhead. We present a framework that discovers and leverages such structural properties at a low computational cost. We employ a bandit optimization procedure that "learns" probabilities for sampling coordinates or examples in (non-smooth) optimization problems, allowing us to guarantee performance close to that of the optimal stationary sampling distribution. When such structures exist, our algorithms achieve tighter convergence guarantees than their non-adaptive counterparts, and we complement our analysis with experiments on several datasets.

1. Introduction

Identifying and adapting to structural aspects of problem data can often improve performance of optimization algorithms. In this paper, we study two forms of such structure: variance in the relative importance of different features and observations (as well as blocks thereof). As a motivating concrete example, consider the ℓ_p regression problem

$$\underset{x}{\text{minimize}} \left\{ f(x) := \|Ax - b\|_p^p = \sum_{i=1}^{n} |a_i^T x - b_i|^p \right\}, \quad (1)$$

where a_i denote the rows of $A \in \mathbb{R}^{n \times d}$. When the columns (features) of A have highly varying norms—say because certain features are infrequent—we wish to leverage this during optimization. Likewise, when rows a_i have disparate norms, "heavy" rows of A influence the objective more than others. We develop optimization algorithms that automatically adapt to such irregularities for general non-smooth convex optimization problems.

Standard (stochastic) subgradient methods (Nemirovski et al., 2009), as well as more recent accelerated variants for smooth, strongly convex incremental optimization problems (e.g. Johnson and Zhang, 2013; Defazio et al., 2014), follow deterministic or random procedures that choose data to use to compute updates in ways that are oblivious to conditioning and structure. As our experiments demonstrate, choosing blocks of features or observations—for instance, all examples belonging to a particular class in classification problems—can be advantageous. Adapting to such structure can lead to substantial gains, and we propose a method that adaptively updates the sampling probabilities from which it draws blocks of features/observations (columns/rows in problem (1)) as it performs subgradient updates. Our method applies to both coordinate descent (feature/column sampling) and mirror descent (observation/row sampling). Heuristically, our algorithm learns to sample informative features/observations using their gradient values and requires overhead only logarithmic in the number of blocks over which it samples. We show that our method optimizes a particular bound on convergence, roughly sampling from the optimal stationary probability distribution in hindsight, and leading to substantial improvements when the data has pronounced irregularity.

When the objective $f(\cdot)$ is smooth and the desired solution accuracy is reasonably low, (block) coordinate descent methods are attractive because of their tractability (Nesterov, 2012; Necoara et al., 2011; Beck and Tetruashvili, 2013; Lee and Sidford, 2013; Richtárik and Takáč, 2014; Lu and Xiao, 2015). In this paper, we consider potentially *non-smooth* functions and present an adaptive block coordinate descent method, which iterates over b blocks of coordinates, reminiscent of AdaGrad (Duchi et al., 2011). Choosing a good sampling distribution for coordinates in coordinate descent procedures is nontrivial (Lee and Sidford, 2013; Necoara et al., 2011; Shalev-Shwartz and Zhang, 2012; Richtárik and Takáč, 2015; Csiba et al., 2015; Allen-Zhu and Yuan, 2015). Most work focuses on choos-

[1]Management Science & Engineering, Stanford University, USA [2]Electrical Engineering, Stanford University, USA [3]Statistics, Stanford University, USA. Correspondence to: Hongseok Namkoong <hnamk@stanford.edu>, Aman Sinha <amans@stanford.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

ing a good stationary distribution using problem-specific knowledge, which may not be feasible; this motivates automatically adapting to individual problem instances. For example, Csiba et al. (2015) provide an updating scheme for the probabilities in stochastic dual ascent. However, the update requires $O(b)$ time per iteration, making it impractical for large-scale problems. Similarly, Nutini et al. (2015) observe that the Gauss-Southwell rule (choosing the coordinate with maximum gradient value) achieves better performance, but this also requires $O(b)$ time per iteration. Our method roughly emulates this behavior via careful adaptive sampling and bandit optimization, and we are able to provide a number of *a posteriori* optimality guarantees.

In addition to coordinate descent methods, we also consider the finite-sum minimization problem

$$\minimize_{x \in \mathcal{X}} \frac{1}{n} \sum_{i=1}^{n} f_i(x),$$

where the f_i are convex and may be non-smooth. Variance-reduction techniques for finite-sum problems often yield substantial gains (Johnson and Zhang, 2013; Defazio et al., 2014), but they generally require smoothness. More broadly, importance sampling estimates (Strohmer and Vershynin, 2009; Needell et al., 2014; Zhao and Zhang, 2014; 2015; Csiba and Richtárik, 2016) can yield improved convergence, but the only work that allows online, problem-specific adaptation of sampling probabilities of which we are aware is Gopal (2016). However, these updates require $O(b)$ computation and do not have optimality guarantees.

We develop these ideas in the coming sections, focusing first in Section 2 on adaptive procedures for (non-smooth) coordinate descent methods and developing the necessary bandit optimization and adaptivity machinery. In Section 3, we translate our development into convergence results for finite-sum convex optimization problems. Complementing our theoretical results, we provide a number of experiments in Section 4 that show the importance of our algorithmic development and the advantages of exploiting block structures in problem data.

2. Adaptive sampling for coordinate descent

We begin with the convex optimization problem

$$\minimize_{x \in \mathcal{X}} f(x) \tag{2}$$

where $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_b \subset \mathbb{R}^d$ is a Cartesian product of closed convex sets $\mathcal{X}_j \subset \mathbb{R}^{d_j}$ with $\sum_j d_j = d$, and f is convex and Lipschitz. When there is a natural block structure in the problem, some blocks have larger gradient norms than others, and we wish to sample these blocks more often in the coordinate descent algorithm. To that end, we develop an adaptive procedure that exploits variability in block "importance" online. In the coming sections, we show that we obtain certain near-optimal guarantees, and that the computational overhead over a simple random choice of block $j \in [b]$ is at most $O(\log b)$. In addition, under some natural structural assumptions on the blocks and problem data, we show how our adaptive sampling scheme provides convergence guarantees polynomially better in the dimension than those of naive uniform sampling or gradient descent.

Notation for coordinate descent Without loss of genrality we assume that the first d_1 coordinates of $x \in \mathbb{R}^d$ correspond to $\mathcal{X}_1$, the second d_2 to $\mathcal{X}_2$, and so on. We let $U_j \in \{0,1\}^{d \times d_j}$ be the matrix identifying the jth block, so that $I_d = [U_1 \cdots U_d]$. We define the projected subgradient vectors for each block j by

$$G_j(x) = U_j U_j^\top f'(x) \in \mathbb{R}^d,$$

where $f'(x) \in \partial f(x)$ is a *fixed* element of the subdifferential $\partial f(x)$. Define $x_{[j]} := U_j^\top x \in \mathbb{R}^{d_j}$ and $G_{[j]}(x) = U_j^\top G_j(x) = U_j^\top f'(x) \in \mathbb{R}^{d_j}$. Let ψ_j denote a differentiable 1-strongly convex function on $\mathcal{X}_j$ with respect to the norm $\|\cdot\|_j$, meaning for all $\Delta \in \mathbb{R}^{d_j}$ we have

$$\psi_j\left(x_{[j]} + \Delta\right) \geq \psi_j\left(x_{[j]}\right) + \nabla\psi_j(x_{[j]})^\top \Delta + \frac{1}{2}\|\Delta\|_j^2,$$

and let $\|\cdot\|_{j,*}$ be the dual norm of $\|\cdot\|_j$. Let $B_j(u,v) = \psi_j(u) - \psi_j(v) - \nabla\psi_j(v)^\top(u-v)$ be the Bregman divergence associated with ψ_j, and define the tensorized divergence $B(x,y) := \sum_{j=1}^b B_j(x_{[j]}, y_{[j]})$. Throughout the paper, we assume the following.

Assumption 1. *For all $x, y \in \mathcal{X}$, we have $B(x,y) \leq R^2$ and $\left\|G_{[j]}(x)\right\|_{j,*}^2 \leq L^2/b$ for $j = 1, \ldots, b$.*

2.1. Coordinate descent for non-smooth functions

The starting point of our analysis is the simple observation that if a coordinate $J \in [b]$ is chosen according to a probability vector $p > 0$, then the importance sampling estimator

$$G_J(x)/p_J \text{ satisfies } \mathbb{E}_p[G_J(x)/p_J] = f'(x) \in \partial f(x).$$

Thus the randomized coordinate subgradient method of Algorithm 1 is essentially a stochastic mirror descent method (Nemirovski and Yudin, 1983; Beck and Teboulle, 2003; Nemirovski et al., 2009), and as long as $\sup_{x \in \mathcal{X}} \mathbb{E}[\|p_J^{-1} G_J(x)\|_*^2] < \infty$ it converges at rate $O(1/\sqrt{T})$. With this insight, a variant of standard stochastic mirror descent analysis yields the following convergence guarantee for Algorithm 1 with non-stationary probabilities (*cf.* Dang and Lan (2015), who do not quite as carefully track dependence on the sampling distribution

Algorithm 1 Non-smooth Coordinate Descent

Input: Stepsize $\alpha_x > 0$, Probabilities $p^1, \dots, p^T$.
Initialize: $x^1 = x$
for $t \leftarrow 1, \dots, T$
  Sample $J_t \sim p^t$
  Update x:
  $x^{t+1}_{[J_t]} \leftarrow \operatorname{argmin}_{x \in \mathcal{X}_{J_t}} \left\{ \left\langle \frac{G_{[J_t]}(x^t)}{p^t_{J_t}}, x \right\rangle + \frac{1}{\alpha_x} B_{J_t}\left(x, x^t_{[J_t]}\right) \right\}$
return $\bar{x}_T \leftarrow \frac{1}{T}\sum_{t=1}^T x^t$

Algorithm 2 Stepsize Doubling Coordinate Descent

Initialize: $x^1 = x$, $p^1 = p$, $k = 1$
while $t \le T$ **do**
  while $\sum_{l=1}^t (p^l_{J_l})^{-2} \left\|G_{[J_l]}(x^l)\right\|^2_{J_l,*} \le 4^k$, $t \le T$ **do**
    Run inner loop of Algorithm 1 with
    $\alpha_{x,k} = \sqrt{2}R\left(4^k + \frac{L^2}{bp^2_{\min}}\right)^{-\frac{1}{2}}$
    $t \leftarrow t + 1$
  $k \leftarrow k + 1$
return $\bar{x}_T \leftarrow \frac{1}{T}\sum_{t=1}^T x^t$

p). Throughout, we define the expected sub-optimality gap of an algorithm outputing an estimate $\widehat{x}$ by $S(f, \widehat{x}) := \mathbb{E}[f(\widehat{x})] - \inf_{x^* \in \mathcal{X}} f(x^*)$. See Section A.1 for the proof.

Proposition 1. *Under Assumption 1, Algorithm 1 achieves*

$$S(f, \overline{x}^T) \le \frac{R^2}{\alpha_x T} + \frac{\alpha_x}{2T}\sum_{t=1}^T \mathbb{E}\left[\sum_{j=1}^b \frac{\left\|G_{[j]}(x^t)\right\|^2_{j,*}}{p^t_j}\right]. \quad (3)$$

where $S(f, \bar{x}_T) = \mathbb{E}[f(\bar{x}_T)] - \inf_{x \in \mathcal{X}} f(x)$.

As an immediate consequence, if $p^t \ge p_{\min} > 0$ and $\alpha_x = \frac{R}{L}\sqrt{\frac{2p_{\min}}{T}}$, then $S(f, \bar{x}_T) \le RL\sqrt{\frac{2}{Tp_{\min}}}$. To make this more concrete, we consider sampling from the uniform distribution $p^t \equiv \frac{1}{b}\mathbb{1}$ so that $p_{\min} = 1/b$, and assume homogeneous block sizes $d_j = d/b$ for simplicity. Algorithm 1 solves problem (2) to ϵ-accuracy within $O(bR^2L^2/\epsilon^2)$ iterations, where each iteration approximately costs $O(d/b)$ plus the cost of projecting into $\mathcal{X}_j$. In contrast, mirror descent with the same constraints and divergence B achieves the same accuracy within $O(R^2L^2/\epsilon^2)$ iterations, taking $O(d)$ time plus the cost of projecting to $\mathcal{X}$ per iteration. As the projection costs are linear in the number b of blocks, the two algorithms are comparable.

In practice, coordinate descent procedures can significantly outperform full gradient updates through efficient memory usage. For huge problems, coordinate descent methods can leverage data locality by choosing appropriate block sizes so that each gradient block fits in local memory.

2.2. Optimal stepsizes by doubling

In the the upper bound (3), we wish to choose the optimal stepsize α_x that minimizes this bound. However, the term $\sum_{t=1}^T \mathbb{E}\left[\sum_{j=1}^b \frac{\|G_{[j]}(x^t)\|^2_{j,*}}{p^t_j}\right]$ is unknown *a priori*. We circumvent this issue by using the doubling trick (e.g. Shalev-Shwartz, 2012, Section 2.3.1) to achieve the best possible rate in hindsight. To simplify our analysis, we assume that there is some $p_{\min} > 0$ such that

$$p^t \in \Delta_b := \left\{p \in \mathbb{R}^b_+ : p^\top \mathbb{1} = 1, p \ge p_{\min}\right\}.$$

Maintaining the running sum $\sum_{l=1}^t p^{-2}_{l,J_l}\left\|G_{[J_l]}(x_l)\right\|^2_{J_l,*}$ requires incremental time $O(d_{J_t})$ at each iteration t, choosing the stepsizes adaptively via Algorithm 2 only requires a constant factor of extra computation over using a fixed step size. The below result shows that the doubling trick in Algorithm 2 acheives (up to log factors) the performance of the optimal stepsize that minimizes the regret bound (3).

Proposition 2. *Under Assumption 1, Algorithm 2 achieves*

$$S(f, \bar{x}_T) \le 6\frac{R}{T}\left(\sum_{t=1}^T \mathbb{E}\left[\sum_{j=1}^b \frac{\left\|G_{[j]}(x^t)\right\|^2_{j,*}}{p^t_j}\right]\right)^{\frac{1}{2}} + \sqrt{\frac{2}{b}}\frac{RL}{p_{\min}T\log 4}\log\left(\frac{4bTL^2}{p_{\min}}\right)$$

where $S(f, \bar{x}_T) = \mathbb{E}[f(\bar{x}_T)] - \inf_{x \in \mathcal{X}} f(x)$.

2.3. Adaptive probabilities

We now present an adaptive updating scheme for p^t, the sampling probabilities. From Proposition 2, the stationary distribution achieving the smallest regret upper bound minimizes the criterion

$$\sum_{t=1}^T \mathbb{E}\left[\sum_{j=1}^b \frac{\left\|G_{[j]}(x^t)\right\|^2_{j,*}}{p_j}\right] = \sum_{t=1}^T \mathbb{E}\left[\frac{\left\|G_{[J_t]}(x^t)\right\|^2_{J_t,*}}{p^2_{J_t}}\right],$$

where the equality follows from the tower property. Since x^t depends on the p^t, we view this as an online convex optimization problem and choose $p^1, \dots, p^T$ to minimize the regret

$$\max_{p \in \Delta_b} \sum_{t=1}^T \mathbb{E}\left[\sum_{j=1}^b \left\|G_{[j]}(x^t)\right\|^2_{j,*}\left(\frac{1}{p^t_j} - \frac{1}{p_j}\right)\right]. \quad (4)$$

Note that due to the block coordinate nature of Algorithm 1, we only compute $\left\|G_{[j]}(x^t)\right\|^2_{j,*}$ for the sampled $j = J_t$ at each iteration. Hence, we treat this as a multi-armed bandit problem where the arms are the blocks $j = 1, \dots, b$ and we only observe the loss $\left\|G_{[j]}(x^t)\right\|^2_{j,*}/(p^t_{J_t})^2$ associated with the arm J_t pulled at time t.

Algorithm 3 Coordinate Descent with Adaptive Sampling

Input: Stepsize $\alpha_p > 0$, Threshold $p_{\min} > 0$ with $\mathcal{P} = \{p \in R_+^b : p^\top \mathbb{1} = 1, p \geq p_{\min}\}$
Initialize: $x^1 = x$, $p^1 = p$
for $t \leftarrow 1, \ldots, T$
  Sample $J_t \sim p^t$
  Choose $\alpha_{x,k}$ according to Algorithm 2
  Update x:
  $x_{[J_t]}^{t+1} \leftarrow \operatorname{argmin}_{x \in \mathcal{X}_{J_t}} \left\{ \left\langle \frac{G_{[J_t]}(x^t)}{p_{J_t}^t}, x \right\rangle + \frac{1}{\alpha_{x,k}} B\left(x, x_{[J_t]}^t\right) \right\}$
  Update p: for $\widehat{\ell}_{t,j}(x)$ defined in (5),
  $w^{t+1} \leftarrow p^t \exp(-(\alpha_p \widehat{\ell}_{t,J_t}(x^t)/p_{J_t}^t) e_{J_t})$,
  $p^{t+1} \leftarrow \operatorname{argmin}_{q \in \mathcal{P}} D_{\mathrm{kl}}\left(q \| w^{t+1}\right)$
return $\bar{x}_T \leftarrow \frac{1}{T} \sum_{t=1}^T x^t$

By using a bandit algorithm—*another* coordinate descent method— to update p, we show that our updates achieve performance comparable to the best stationary probability in Δ_b in hindsight. To this end, we first bound the regret (4) by the regret of a linear bandit problem. By convexity of $x \mapsto 1/x$ and $\frac{d}{dx} x^{-1} = -x^{-2}$, we have

$$\sum_{t=1}^T \mathbb{E}\left[\sum_{j=1}^b \left\|G_{[j]}(x^t)\right\|_{j,*}^2 \left(\frac{1}{p_j^t} - \frac{1}{p_j}\right)\right] \leq \sum_{t=1}^T \mathbb{E}\left[\left\langle \underbrace{-\left\{\left\|G_{[j]}(x^t)\right\|_{j,*}^2 / (p_j^t)^2\right\}_{j=1}^b}_{(*)}, p^t - p \right\rangle\right].$$

Now, let us view the vector $(*)$ as the loss vector for a constrained linear bandit problem with feasibility region Δ_b. We wish to apply EXP3 (due to Auer et al. (2002)) or equivalently, a 1-sparse mirror descent to p with $\psi_P(p) = p \log p$ (see, for example, Section 5.3 of Bubeck and Cesa-Bianchi (2012) for the connections). However, EXP3 requires the loss values be positive in order to be in the region where ψ_P is strongly convex, so we scale our problem using the fact that p and p^t's are probability vectors. Namely,

$$\sum_{t=1}^T \mathbb{E}\left[\left\langle -\left\{\left\|G_{[j]}(x^t)\right\|_{j,*}^2 / (p_j^t)^2\right\}_{j=1}^b, p^t - p \right\rangle\right] = \sum_{t=1}^T \mathbb{E}\left[\left\langle \widehat{\ell}_t(x^t), p^t - p \right\rangle\right],$$

where

$$\widehat{\ell}_{t,j}(x) := -\frac{\left\|G_{[j]}(x)\right\|_{j,*}^2}{(p_j^t)^2} + \frac{L^2}{b p_{\min}^2}. \tag{5}$$

Using scaled loss values, we perform EXP3 (Algorithm 3). Intuitively, we penalize the probability of the sampled block by the strength of the signal on the block. The scaling (5) ensures that we penalize blocks with low signal (as opposed to rewarding those with high signal) which enforces diversity in the sampled coordinates as well. In Section A.3, we will see how this scaling plays a key role in proving optimality of Algorithm 3. Here, the signal is measured by the relative size of the gradient in the block against the probability of sampling the block. This means that blocks with large "surprises"—those with higher gradient norms relative to their sampling probability—will get sampled more frequently in the subsequent iterations. Algorithm 3 guarantees low regret for the online convex optimization problem (4) which in turn yields the following guarantee for Algorithm 3.

Theorem 3. *Under Assumption 1, the adaptive updates in Algorithm 3 with* $\alpha_p = \frac{p_{\min}^2}{L^2}\sqrt{\frac{2b\log b}{T}}$ *achieve*

$$S(f, \bar{x}_T) \leq \underbrace{\frac{6R}{T}\sqrt{\min_{p \in \Delta_b} \sum_{t=1}^T \mathbb{E}\left[\sum_{j=1}^b \frac{\|G_{[j]}(x^t)\|_{j,*}^2}{p_j}\right]}}_{(a):\textit{best in hindsight}} + \underbrace{\frac{8LR}{T p_{\min}}\left(\frac{T\log b}{b}\right)^{\frac{1}{4}}}_{(b):\textit{regret for bandit problem}} + \frac{2RL}{\sqrt{b} T p_{\min}} \log\left(\frac{4bTL^2}{p_{\min}}\right). \tag{6}$$

where $S(f, \bar{x}_T) = \mathbb{E}[f(\bar{x}_T)] - \inf_{x \in \mathcal{X}} f(x)$.

See Section A.3 for the proof. Note that there is a trade-off in the regret bound (6) in terms of $p_{\min}$: for small $p_{\min}$, the first term is small, as the the set Δ_b is large, but second (regret) term is large, and vice versa. To interpret the bound (6), take $p_{\min} = \delta/b$ for some $\delta \in (0, 1)$. The first term dominates the remainder as long as $T = \Omega(b \log b)$; we require $T \asymp (bR^2L^2/\epsilon^2)$ to guarantee convergence of coordinate descent in Proposition 1, so that we roughly expect the first term in the bound (6) to dominate. Thus, Algorithm 3 attains the best convergence guarantee for the optimal stationary sampling distribution in hindsight.

2.4. Efficient updates for p

The updates for p in Algorithm 3 can be done in $O(\log b)$ time by using a balanced binary tree. Let $D_{\mathrm{kl}}(p\|q) := \sum_{i=1}^d p_i \log \frac{p_i}{q_i}$ denote the Kullback-Leibler divergence between p and q. Ignoring the subscript on t so that $w = w^{t+1}$, $p = p^t$ and $J = J_t$, the new probability vector q is given by the minimizer of

$$D_{\mathrm{kl}}(q\|w) \quad \text{s.t. } q^\top \mathbb{1} = 1, q \geq p_{\min}, \tag{7}$$

where w is the previous probability vector p modified only at the index J. We store w in a binary tree, keeping values up to their normalization factor. At each node, we also store the sum of elements in the left/right subtree for

Algorithm 4 KL Projection

Input: J, p_J, w_J, mass $= \sum_i w_i$
$w_{\text{cand}} \leftarrow p_J \cdot \text{mass}$.
if $w_{\text{cand}}/(\text{mass} - w_J + w_{\text{cand}}) \le p_{\min}$ **then**
  $w_{\text{cand}} \leftarrow \frac{p_{\min}}{1-p_{\min}} (\text{mass} - w_J)$
Update(w_{cand}, J)

efficient sampling (for completeness, the pseudo-code for sampling from the binary tree in $O(\log b)$ time is given in Section B.3). The total mass of the tree can be accessed by inspecting the root of the tree alone.

The following proposition shows that it suffices to touch at most one element in the tree to do the update. See Section B for the proof.

Proposition 4. *The solution to* (7) *is given by*

$$q_{j \neq J} = \begin{cases} \frac{1}{1-p_J+w_J} w_j & \text{if } w_J \ge \frac{p_{\min}(1-p_J)}{1-p_{\min}} \\ \frac{1-p_{\min}}{1-p_J} w_j & \text{otherwise} \end{cases},$$

$$q_J = \begin{cases} \frac{1}{1-p_J+w_J} w & \text{if } w_J \ge \frac{p_{\min}(1-p_J)}{1-p_{\min}} \\ p_{\min} & \text{otherwise.} \end{cases}$$

As seen in Algorithm 4, we need to modify at most one element in the binary tree. Here, the update function modifies the value at index J and propagates the value up the tree so that the sum of left/right subtrees are appropriately updated. We provide the full pseudocode in Section B.2.

2.5. Example

The optimality guarantee given in Theorem 3 is not directly interpretable since the term (a) in the upper bound (6) is only optimal given the iterates $x^1, \ldots, x^T$ despite the fact that x^t's themselves depend on the sampling probabilities. Hence, we now study a setting where we can further bound (6) to get a explicit regret bound for Algorithm 3 that is provably better than non-adaptive counterparts. Indeed, under certain structural assumptions on the problem similar to those of McMahan and Streeter (2010) and Duchi et al. (2011), our adaptive sampling algorithm provably achieves regret polynomially better in the dimension than either using a uniform sampling distribution or gradient descent.

Consider the SVM objective

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} \left(1 - y_i z_i^\top x\right)_+$$

where n is small and d is large. Here, $\partial f(x) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\left\{1 - y_i z_i^\top x \ge 0\right\} z_i$. Assume that for some fixed $\alpha \in (1, \infty)$ and $L_j := \beta j^{-\alpha}$, we have $|\partial_j f(x)|^2 \le \frac{1}{n} \sum_{i=1}^n |z_{i,j}|^2 \le L_j^2$. In particular, this is the case if we have sparse features $z_U \in \{-1, 0, +1\}^d$ with power law tails $P(|z_{U,j}| = 1) = \beta j^{-\alpha}$ where U is a uniform random variable over $\{1, \ldots, n\}$.

Algorithm	$\alpha \in [2, \infty)$	$\alpha \in (1, 2)$
ACD	$\left(\frac{R}{\epsilon}\right)^2 \log^2 d + \left(\frac{R}{\epsilon}\right)^{\frac{4}{3}} d \log^{\frac{5}{3}} d$	$\left(\frac{R}{\epsilon}\right)^2 d^{2-\alpha} + \left(\frac{R}{\epsilon}\right)^{\frac{4}{3}} d \log^{\frac{5}{3}} d$
UCD	$\left(\frac{R}{\epsilon}\right)^2 d \log d$	
GD	$\left(\frac{R}{\epsilon}\right)^2 d \log d$	

Table 1. Runtime comparison (computations needed to guarantee ϵ-optimality gap) under heavy-tailed block structures. ACD=adaptive coordinate descent, UCD=uniform coordinate descent, GD=gradient descent

Take $C_j = \{j\}$ for $j = 1, \ldots, d$ (and $b = d$). First, we show that although for the uniform distribution $p = \mathbb{1}/d$

$$\sum_{j=1}^{d} \frac{\mathbb{E}[\|G_j(x^t)\|_*^2]}{1/d} \le d \sum_{j=1}^{d} L_j^2 = O(d \log d),$$

the term (a) in (6) can be orders of magnitude smaller.

Proposition 5. *Let $b = d$, $p_{\min} = \delta/d$ for some $\delta \in (0, 1)$, and $C_j = \{j\}$. If $\|G_j(x)\|_*^2 \le L_j^2 := \beta j^{-\alpha}$ for some $\alpha \in (1, \infty)$, then*

$$\min_{p \in \Delta_b, p \ge p_{\min}} \sum_{j=1}^{d} \frac{\mathbb{E}[\|G_j(x^t)\|_*^2]}{p_j} = \begin{cases} O(\log d), & \text{if } \alpha \in [2, \infty) \\ O(d^{2-\alpha}), & \text{if } \alpha \in (1, 2). \end{cases}$$

We defer the proof of the proposition to Section A.5. Using this bound, we can show explicit regret bounds for Algorithm 3. From Theorem 3 and Proposition 5, we have that Algorithm 3 attains

$$S(f, \bar{x}_T) \le \begin{cases} O(\frac{R \log d}{\sqrt{T}}), & \text{if } \alpha \in [2, \infty) \\ \frac{R}{\sqrt{T}} O(d^{1-\frac{\alpha}{2}}), & \text{if } \alpha \in (1, 2) \end{cases} + O\left(R d^{3/4} T^{-3/4} \log^{5/4} d\right).$$

Setting above to be less than ϵ and inverting with respect to T, we obtain the iteration complexity in Table 1.

To see the runtime bounds for uniformly sampled coordinate descent and gradient descent, recall the regret bound (3) given in Proposition 1. Plugging $p_j^t = 1/d$ in the bound, we obtain

$$S(f, \bar{x}_T) \le O(R \sqrt{\log d} \sqrt{2dT}).$$

for $\alpha_x = \sqrt{2R^2/(L^2 T d)}$ where $L^2 := \sum_{j=1}^d L_j^2$. Similarly, gradient descent with $\alpha_x = \sqrt{2R^2/(L^2 T)}$ attains

$$S(f, \bar{x}_T) \le O(R \sqrt{\log d} \sqrt{2T}).$$

Since each gradient descent update takes $O(d)$, we obtain the same runtime bound.

While non-adaptive algorithms such as uniformly-sampled coordinate descent or gradient descent have the same runtime for all α, our adaptive sampling method automatically tunes to the value of α. Note that for $\alpha \in (1, \infty)$, the first term in the runtime bound for our adaptive method given in Table 1 is strictly better than that of uniform coordinate descent or gradient descent. In particular, for $\alpha \in [2, \infty)$ the best stationary sampling distribution in hindsight yields an improvement that is at most $O(d)$ better in the dimension. However, due to the remainder terms for the bandit problem, this improvement only matters (*i.e.*first term is larger than second) when

$$\epsilon = \begin{cases} O\left(Rd^{-\frac{3}{2}}\sqrt{\log d}\right) & \text{if } \alpha \in [2, \infty) \\ O\left(Rd^{\frac{3}{2}(1-\alpha)}\log^{-5/2} d\right) & \text{if } \alpha \in (1, 2). \end{cases}$$

In Section 4, we show that these remainder terms can be made smaller than what their upper bounds indicate. Empirically, our adaptive method outperforms the uniformly-sampled counterpart for larger values of ϵ than above.

3. Adaptive probabilities for stochastic gradient descent

Consider the empirical risk minimization problem

$$\underset{x \in \mathcal{X}}{\text{minimize}} \left\{ \frac{1}{n} \sum_{i=1}^{n} f_i(x) =: f(x) \right\}$$

where $\mathcal{X} \in \mathbb{R}^d$ is a closed convex set and $f_i(\cdot)$ are convex functions. Let $C_1, \ldots, C_b$ be a partition of the n samples so that each example belongs to some C_j, a set of size $n_j := |C_j|$ (note that the index j now refers to blocks of examples instead of coordinates). These block structures naturally arise, for example, when C_j's are the examples with the same label in a multi-class classification problem. In this stochastic optimization setting, we now sample a block $J_t \sim p^t$ at each iteration t, and perform gradient updates using a gradient estimate on the block C_{J_t}. We show how a similar adaptive updating scheme for p^t's again achieves the optimality guarantees given in Section 2.

3.1. Mirror descent with non-stationary probabilities

Following the approach of (Nemirovski et al., 2009), we run mirror descent for the updates on x. At iteration t, a block J_t is drawn from a b-dimensional probability vector p^t. We assume that we have access to unbiased stochastic gradients $G_j(x)$ for each block. That is, $\mathbb{E}[G_j(x)] = \frac{1}{n_j}\sum_{i \in C_j} \partial f_i(x)$. In particular, the estimate $G_{J_t}(x^t) := \partial f_{I_t}(x)$ where I_t is drawn uniformly in C_{J_t} gives the usual unbiased stochastic gradient of minibatch size 1. The other extreme is obtained by using a minibatch size of n_j where $G_{J_t}(x^t) := \frac{1}{n_{J_t}}\sum_{i \in C_{J_t}} \partial f_i(x)$. Then, the importance sampling estimator $\frac{n_{J_t}}{np^t_{J_t}} G_{J_t}(x^t)$ is an unbiased estimate for the subgradient of the objective.

Let ψ be a differentiable 1-strongly convex function on $\mathcal{X}$ with respect to the norm $\|\cdot\|$ as before and denote by $\|\cdot\|_*$ the dual norm of $\|\cdot\|$. Let $B(x, y) = \psi(x) - \psi(y) - \nabla\psi(y)^\top(x - y)$ be the Bregman divergence associated with ψ. In this section, we assume the below (standard) bound.

Assumption 2. *For all $x, y \in \mathcal{X}$, we have $B(x, y) \le R^2$ and $\|G_j(x)\|_*^2 \le L$ for $j = 1, \ldots, b$.*

We use these stochastic gradients to perform mirror updates, replacing the update in Algorithm 1 with the update

$$x^{t+1} \leftarrow \underset{x \in \mathcal{X}}{\text{argmin}} \left\{ \frac{n_{J_t}}{np^t_{J_t}} \langle G_{J_t}(x^t), x \rangle + \frac{1}{\alpha_x} B(x, x^t) \right\}. \quad (8)$$

From a standard argument (*e.g.*, (Nemirovski et al., 2009)), we obtain the following convergence guarantee. The proof follows an argument similar to that of Proposition 1.

Proposition 6. *Under Assumption 2, the updates (8) attain*

$$S(f, \bar{x}_T) \le \frac{R^2}{\alpha_x T} + \frac{\alpha_x}{2T}\sum_{t=1}^{T} \mathbb{E}\left[\sum_{j=1}^{b} \frac{n_j^2 \|G_j(x^t)\|_*^2}{n^2 p_j^t}\right]. \quad (9)$$

where $S(f, \bar{x}_T) = \mathbb{E}[f(\bar{x}_T)] - \inf_{x \in \mathcal{X}} f(x)$.

Again, we wish to choose the optimal step size α_x that minimizes the regret bound (9). To this end, modify the doubling trick given in Algorithm 2 as follows: use $\sum_{l=1}^{t} \frac{n_{J_l}^2}{n^2 p_{l,J_l}^2} \left\|G_{J_l}(x^l)\right\|_*^2$ for the second while condition, and stepsizes $\alpha_{x,k} = \sqrt{2}R\left(4^k + \frac{L^2 \max_j n_j^2}{n^2 p_{\min}^2}\right)^{-\frac{1}{2}}$. Then, similar to Proposition 2, we have

$$S(f, \bar{x}_T) \le 6\frac{R}{T}\left(\sum_{t=1}^{T} \mathbb{E}\left[\sum_{j=1}^{b} \frac{n_j^2}{n^2 p_j^t} \left\|G_j(x^t)\right\|_*^2\right]\right)^{\frac{1}{2}} + \frac{\sqrt{2}RL}{p_{\min} T \log 4} \frac{\max_j n_j}{n} \log\left(\frac{4TL^2}{p_{\min}} \sum_{j=1}^{b} \frac{n_j^2}{n^2}\right).$$

3.2. Adaptive probabilities

Now, we consider an adaptive updating scheme for p^t's similar to Section 2.3. Using the scaled gradient estimate

$$\widehat{\ell}_{t,j}(x) := -\left(\frac{n_j}{np_j^t}\|G_j(x)\|_*\right)^2 + \frac{L^2 \max_j n_j^2}{n^2 p_{\min}^2} \quad (10)$$

to run EXP3, we obtain Algorithm 5. Again, the additive scaling $L^2(\max_j n_j / np_{\min})^2$ is to ensure that $\widehat{\ell} \ge 0$. As in Section 2.4, the updates for p in Algorithm 5 can be done in $O(\log b)$ time. We can also show similar optimality guarantees for Algorithm 5 as before. The proof is essentially the same to that given in Section A.3.

Algorithm 5 Mirror Descent with Adaptive Sampling

Input: Stepsize $\alpha_p > 0$
Initialize: $x^1 = x$, $p^1 = p$
for $t \leftarrow 1, \ldots, T$
  Sample $J_t \sim p^t$
  Choose $\alpha_{x,k}$ according to (modified) Algorithm 2.
  Update x:
  $x_{J_t}^{t+1} \leftarrow \operatorname{argmin}_{x \in \mathcal{X}} \left\{ \frac{1}{p_{J_t}^t} \langle G_{J_t}(x^t), x \rangle + \frac{1}{\alpha_{x,k}} B\left(x, x_{J_t}^t\right) \right\}$
  Update p:
  $w^{t+1} \leftarrow p^t \exp(-(\alpha_p \widehat{\ell}_{t,J_t}(x^t)/p_{J_t}^t) e_{J_t})$
  $p^{t+1} \leftarrow \operatorname{argmin}_{q \in \mathcal{P}} D_{\mathrm{kl}}\left(q \| w^{t+1}\right)$
return $\bar{x}_T \leftarrow \frac{1}{T} \sum_{t=1}^T x^t$

Theorem 7. *Let* $W := \frac{L \max_j n_j}{p_{\min} n}$. *Under Assumption 2, Algorithm 5 with* $\alpha_p = \frac{1}{W^2} \sqrt{\frac{2 \log b}{bT}}$ *achieves*

$$S(f, \bar{x}_T) \leq \frac{6R}{T} \min_{p \in \Delta_b} \sum_{t=1}^T \mathbb{E}\left[\sum_{j=1}^b \frac{n_j^2}{n^2 p_j} \left\| G_{J_t}(x^t) \right\|_*^2 \right] + W(2Tb \log b)^{\frac{1}{4}} + \frac{\sqrt{2} RW}{T \log 4} \log\left(\frac{4TL^2}{p_{\min}} \sum_{j=1}^b \frac{n_j^2}{n^2} \right)$$

where $S(f, \bar{x}_T) = \mathbb{E}[f(\bar{x}_T)] - \inf_{x \in \mathcal{X}} f(x)$.

With equal block sizes $n_j = n/b$ and $p_{\min} = \delta/b$ for some $\delta \in (0, 1)$, the first term in the boudn of Theorem 7 is $O(TL^2)$ which dominates the second term if $T = \Omega(b \log b)$. Since we usually have $T = \Theta(n)$ for SGD, as long as $n = \Omega(b \log b)$ we have

$$S(f, \bar{x}_T) \leq O\left(\frac{R}{T} \sqrt{ \min_{p \in \Delta_b} \sum_{t=1}^T \mathbb{E}\left[\sum_{j=1}^b \frac{\left\| G_{[j]}(x^t) \right\|_{j,*}^2}{p_j} \right] } \right).$$

That is, Algorithm 5 attains the best regret bound achieved by the optimal stationary distribution in hindsight had the x^t's had remained the same. Further, under similar structural assumptions $\|G_j(x)\|_*^2 \propto j^{-\alpha}$ as in Section 2.5, we can prove that the regret bound for our algorithm is better than that of the uniform distribution.

4. Experiments

We compare performance of our adaptive approach with stationary sampling distributions on real and synthetic datasets. To minimize parameter tuning, we fix α_p at the value suggested by theory in Theorems 3 and 7. However, we make a heuristic modification to our adaptive algorithm since rescaling the bandit gradient (5) and (10) dwarfs the signals in gradient values if L is too large. We present performance of our algorithm with respect to multiple estimates of the Lipschitz constant $\hat{L} = L/c$ for $c > 1$, where L is the actual upper bound.[1] We tune the stepsize α_x for both methods, using the form $\beta/\sqrt{t}$ and tuning β.

For all our experiments, we compare our method against the uniform distribution and blockwise Lipschitz sampling distribution $p_j \propto L_j$ where L_j is the Lipschitz constant of the j-th block (Zhao and Zhang, 2015). We observe that the latter method often performs very well with respect to iteration count. However, since computing the blockwise Lipschitz sampling distribution takes $O(nd)$, the method is not competitive in large-scale settings. Our algorithm, on the other hand, adaptively learns the latent structure and often outperforms stationary counterparts with respect to runtime. While all of our plots are for a single run with a random seed, we can reject the null hypothesis $f(x_{\text{uniform}}^T) < f(x_{\text{adaptive}}^T)$ at 99% confidence for all instances where our theory guarantees it. We take $\|\cdot\| = \|\cdot\|_2$ throughout this section.

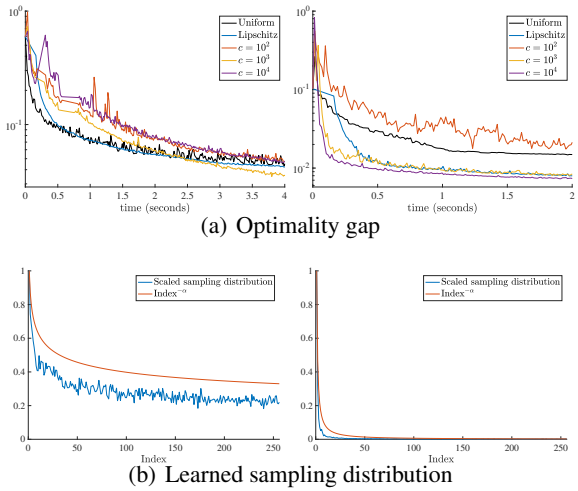
4.1. Adaptive sampling for coordinate descent

Synthetic Data We begin with coordinate descent, first verifying the intuition of Section 2.5 on a synthetic dataset. We consider the problem $\text{minimize}_{\|x\|_\infty \leq 1} \frac{1}{n} \|Ax - b\|_1$, and we endow $A \in \mathbb{R}^{n \times d}$ with the following block structure: the columns are drawn as $a_j \sim j^{-\alpha/2} N(0, I)$. Thus, the gradients of the columns decay in a heavy-tailed manner as in Section 2.5 so that $L_j^2 = j^{-\alpha}$. We set $n = d = b = 256$; the effects of changing ratios n/d and b/d manifest themselves via relative norms of the gradients in the columns, which we control via α instead. We run all experiments with $p_{\min} = 0.1/b$ and multiple values of c.

Results are shown in Figure 1, where we show the optimality gap vs. runtime in (a) and the learned sampling distribution in (b). Increasing α (stronger block structure) improves our relative performance with respect to uniform sampling and our ability to accurately learn the underlying block structure. Experiments over more α and c in Section C further elucidate the phase transition from uniform-like behavior to regimes learning/exploiting structure.

We also compare our method with (non-preconditioned) SGD using leverage scores $p_j \propto \|a_j\|_1$ given by (Yang et al., 2016). The leverage scores (*i.e.*, sampling distribution proportional to blockwise Lipschitz constants) roughly correpond to using $p_j \propto j^{-\alpha/2}$, which is the stationary distribution that minimizes the bound (3); in this synthetic setting, this sampling probability coincides with the actual block structure. Although this is expensive to compute, taking $O(nd)$ time, it exploits the latent block structure very well as expected. Our method quickly learns the structure and performs comparably with this "optimal" distribution.

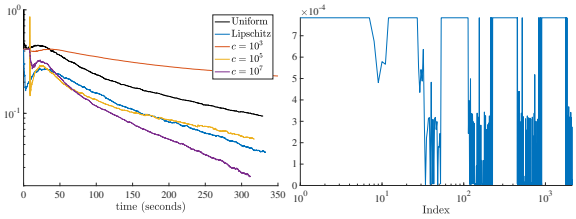
[1] We guarantee a positive loss by taking $\max(\widehat{\ell}_{t,j}(x), 0)$.



(a) Optimality gap

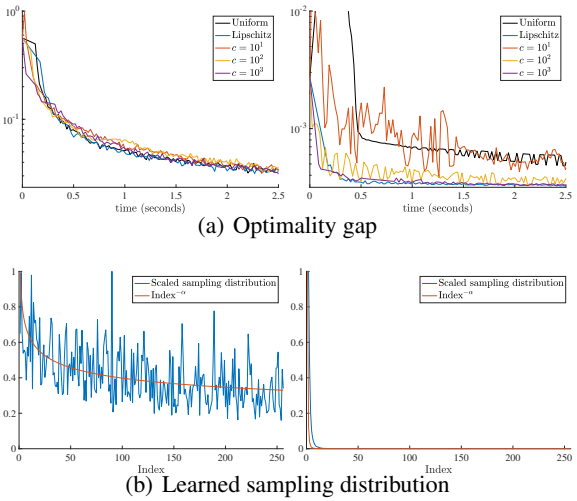
(b) Learned sampling distribution

Figure 1. Adaptive coordinate descent (left to right: $\alpha = 0.4, 2.2$)



(a) Optimality gap (b) Learned distribution

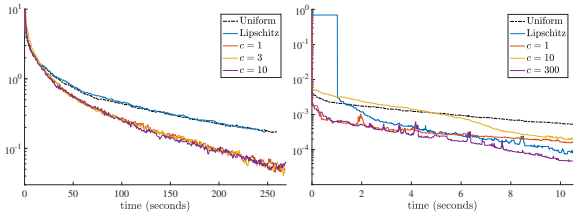
Figure 2. Model selection for nucleotide sequences



(a) Optimality gap

(b) Learned sampling distribution

Figure 3. Adaptive SGD (left to right: $\alpha = 0.4, 6$)



(a) CUB-200 (b) ALOI

Figure 4. Optimality gap for CUB-200-2011 and ALOI

Model selection Our algorithm's ability to learn underlying block structure can be useful in its own right as an online feature selection mechanism. We present one example of this task, studying an aptamer selection problem (Cho et al., 2013), which consists of $n = 2900$ nucleotide sequences (aptamers) that are one-hot-encoded with all k-grams of the sequence, where $1 \leq k \leq 5$ so that $d = 105,476$. We train an l_1-regularized SVM on the binary labels, which denote (thresholded) binding affinity of the aptamer. We set the blocksize as 50 features ($b = 2110$) and $p_{\min} = 0.01/b$. Results are shown in Figure 2, where we see that adaptive feature selection certainly improves training time in (a). The learned sampling distribution depicted in (b) for the best case ($c = 10^7$) places larger weight on features known as G-complexes; these features are well-known to affect binding affinities (Cho et al., 2013).

4.2. Adaptive sampling for SGD

Synthetic data We use the same setup as in Section 4.1 but now endow block structure on the rows of A rather than the columns. In Figure 3, we see that when there is little block structure ($\alpha = 0.4$) all sampling schemes perform similarly. When the block structure is apparent ($\alpha = 6$), our adaptive method again learns the underlying structure and outperforms uniform sampling. We provide more experiments in Section C to illustrate behaviors over more c and α. We note that our method is able to handle online data streams unlike stationary methods such as leverage scores.

CUB-200-2011/ALOI We apply our method to two multi-class object detection datasets: Caltech-UCSD Birds-200-2011 (Wah et al., 2011) and ALOI (Geusebroek et al., 2005). Labels are used to form blocks so that $b = 200$ for CUB and $b = 1000$ for ALOI. We use softmax loss for CUB-200-2011 and a binary SVM loss for ALOI, where in the latter we do binary classification between shells and non-shell objects. We set $p_{\min} = 0.5/b$ to enforce enough exploration. For the features, outputs of the last fully-connected layer of ResNet-50 (He et al., 2016) are used for CUB so that we have 2049-dimensional features. Since our classifier x is $(b \cdot d)$-dimensional, this is a fairly large scale problem. For ALOI, we use default histogram features ($d = 128$). In each case, we have $n = 5994$ and $n = 108,000$ respectively. We use $\mathcal{X} := \{x \in \mathbb{R}^m : \|x\|_2 \leq r\}$ where $r = 100$ for CUB and $r = 10$ for ALOI. We observe in Figure 4 that our adaptive sampling method outperforms stationary counterparts.

Acknowledgements

HN was supported by the Samsung Scholarship. AS and SY were supported by Stanford Graduate Fellowships and AS was also supported by a Fannie & John Hertz Foundation Fellowship. JCD was supported by NSF-CAREER-1553086.

References

Z. Allen-Zhu and Y. Yuan. Even faster accelerated coordinate descent using non-uniform sampling. *arXiv preprint arXiv:1512.09103*, 2015.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31:167–175, 2003.

A. Beck and L. Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060, 2013.

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.

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

M. Cho, S. S. Oh, J. Nie, R. Stewart, M. Eisenstein, J. Chambers, J. D. Marth, F. Walker, J. A. Thomson, and H. T. Soh. Quantitative selection and parallel characterization of aptamers. *Proceedings of the National Academy of Sciences*, 110(46), 2013.

D. Csiba and P. Richtárik. Importance sampling for minibatches. *arXiv preprint arXiv:1602.02283*, 2016.

D. Csiba, Z. Qu, and P. Richtárik. Stochastic dual coordinate ascent with adaptive probabilities. *arXiv preprint arXiv:1502.08053*, 2015.

C. D. Dang and G. Lan. Stochastic block mirror descent methods for nonsmooth and stochastic optimization. *SIAM Journal on Optimization*, 25(2):856–881, 2015.

A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems 27*, 2014.

J. C. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

J.-M. Geusebroek, G. J. Burghouts, and A. W. Smeulders. The amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112, 2005.

S. Gopal. Adaptive sampling for sgd by exploiting side information. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 364–372, 2016.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, 2013.

Y. T. Lee and A. Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *54th Annual Symposium on Foundations of Computer Science*, pages 147–156. IEEE, 2013.

Z. Lu and L. Xiao. On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming*, 152(1-2):615–642, 2015.

B. McMahan and M. Streeter. Adaptive bound optimization for online convex optimization. In *Proceedings of the Twenty Third Annual Conference on Computational Learning Theory*, 2010.

I. Necoara, Y. Nesterov, and F. Glineur. A random coordinate descent method on large optimization problems with linear constraints. *University Politehnica Bucharest, Tech. Rep*, 2011.

D. Needell, R. Ward, and N. Srebro. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. In *Advances in Neural Information Processing Systems 27*, pages 1017–1025, 2014.

A. Nemirovski and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley, 1983.

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.

Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

J. Nutini, M. Schmidt, I. H. Laradji, M. Friedlander, and H. Koepke. Coordinate descent converges faster with the gauss-southwell rule than random selection. *arXiv preprint arXiv:1506.00552*, 2015.

P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144 (1-2):1–38, 2014.

P. Richtárik and M. Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, page Online first, 2015. URL http://link.springer.com/article/10.1007/s10107-015-0901-6.

S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

S. Shalev-Shwartz and T. Zhang. Proximal stochastic dual coordinate ascent. *arXiv preprint arXiv:1211.2717*, 2012.

T. Strohmer and R. Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15(2):262–278, 2009.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

J. Yang, Y.-L. Chow, C. Ré, and M. W. Mahoney. Weighted sgd for p regression with randomized preconditioning. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 558–569. Society for Industrial and Applied Mathematics, 2016.

P. Zhao and T. Zhang. Accelerating minibatch stochastic gradient descent using stratified sampling. *arXiv:1405.3080 [stat.ML]*, 2014.

P. Zhao and T. Zhang. Stochastic optimization with importance sampling. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

A. Proofs

A.1. Proof of Proposition 1

Let $\sigma_t := \sigma(x^1, \dots, x^t, J_1, \dots, J_{t-1})$ and $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot|\sigma_t]$. By convexity of f and unbiasedness of our gradient estimator $\mathbb{E}_t\left[\frac{1}{p^t_{J_t}} G_{J_t}(x^t)\right] = g(x^t) \in \partial f(x^t)$, we have

$$f(x^t) - f(x) \leq \mathbb{E}_t\left[\frac{1}{p^t_{J_t}} \left\langle G_{J_t}(x^t), x^t - x\right\rangle\right].$$

We use the following lemma.

Lemma 1. *For any $t = 1, \dots, T$, we have*

$$\frac{1}{p^t_{J_t}} \left\langle G_{J_t}(x^t), x^t - x\right\rangle \leq \frac{1}{\alpha_x}\left(B(x, x^t) - B(x, x^{t+1})\right) + \frac{\alpha_x}{2(p^t_{J_t})^2}\left\|G_{[J_t]}(x^t)\right\|^2_{J_t,*}.$$

Proof of Lemma

$$\begin{aligned}
&\frac{1}{p^t_{J_t}} \left\langle G_{J_t}(x^t), x^t - x\right\rangle \\
&= \frac{1}{p^t_{J_t}} \left\langle G_{J_t}(x^t), x^{t+1} - x\right\rangle + \frac{1}{p^t_{J_t}} \left\langle G_{J_t}(x^t), x^t - x^{t+1}\right\rangle \\
&= \frac{1}{p^t_{J_t}} \left\langle G_{[J_t]}(x^t), x^{t+1}_{[J_t]} - x_{[J_t]}\right\rangle \\
&\quad + \frac{1}{p^t_{J_t}} \left\langle G_{[J_t]}(x^t), x^t_{[J_t]} - x^{t+1}_{[J_t]}\right\rangle \\
&\overset{(a)}{\leq} \frac{1}{\alpha_x} \left\langle \nabla\psi_{J_t}\left(x^{t+1}_{[J_t]}\right) - \nabla\psi_{J_t}\left(x^t_{[J_t]}\right), x_{[J_t]} - x^{t+1}_{[J_t]}\right\rangle \\
&\quad + \frac{1}{p^t_{J_t}} \left\langle G_{[J_t]}(x^t), x^t_{[J_t]} - x^{t+1}_{[J_t]}\right\rangle \\
&\overset{(b)}{\leq} \frac{1}{\alpha_x}\left(B_{J_t}\left(x_{[J_t]}, x^t_{[J_t]}\right) - B_{J_t}\left(x_{[J_t]}, x^{t+1}_{[J_t]}\right)\right) \\
&\quad - \frac{1}{2\alpha_x}\left\|x^{t+1}_{[J_t]} - x^t_{[J_t]}\right\|^2_{J_t} \\
&\quad + \frac{1}{p^t_{J_t}} \left\langle G_{[J_t]}(x^t), x^t_{[J_t]} - x^{t+1}_{[J_t]}\right\rangle \\
&\overset{(c)}{\leq} \frac{1}{\alpha_x}\left(B_{J_t}\left(x_{[J_t]}, x^t_{[J_t]}\right) - B_{J_t}\left(x_{[J_t]}, x^{t+1}_{[J_t]}\right)\right) \\
&\quad + \frac{\alpha_x}{2\left(p^t_{J_t}\right)^2}\left\|G_{[J_t]}(x^t)\right\|^2_{J_t,*} \\
&\overset{(d)}{=} \frac{1}{\alpha_x}\left(B(x, x^t) - B\left(x, x^{t+1}\right)\right) + \frac{\alpha_x}{2(p^t_{J_t})^2}\left\|G_{[J_t]}(x^t)\right\|^2_{J_t,*}.
\end{aligned}$$

where in step (a) we used the optimality conditions for the mirror update in Algorithm 1. Step (b) follows from the algebraic relation

$$\begin{aligned}
&\left\langle \nabla\psi_{J_t}(x^{t+1}_{[J_t]}) - \nabla\psi_{J_t}(x^t_{[J_t]}), x_{[J_t]} - x^{t+1}_{[J_t]}\right\rangle \\
&= B_{J_t}\left(x_{[J_t]}, x^t_{[J_t]}\right) - B_{J_t}\left(x_{[J_t]}, x^{t+1}_{[J_t]}\right) - B_{J_t}\left(x^{t+1}_{[J_t]}, x^t_{[J_t]}\right),
\end{aligned}$$

and strong convexity of ψ_{J_t} given by $B_{J_t}\left(x^{t+1}_{[J_t]}, x^t_{[J_t]}\right) \geq \frac{1}{2}\left\|x^{t+1}_{[J_t]} - x^t_{[J_t]}\right\|^2_{J_t}$. For step (c), we used Fenchel-Young's inequality:

$$\begin{aligned}
&\frac{1}{p^t_{J_t}} \left\langle G_{[J_t]}(x^t), x^t_{[J_t]} - x^{t+1}_{[J_t]}\right\rangle \\
&\leq \frac{\alpha_x}{2(p^t_{J_t})^2}\left\|G_{[J_t]}(x^t)\right\|^2_{J_t,*} + \frac{1}{2\alpha_x}\left\|x^{t+1}_{[J_t]} - x^t_{[J_t]}\right\|^2_{J_t}.
\end{aligned}$$

Furthermore, due to the fact that x^t and x^{t+1} only differ in the J_t-th block, we have

$$\begin{aligned}
&B_{J_t}\left(x_{[J_t]}, x^t_{[J_t]}\right) - B_{J_t}\left(x_{[J_t]}, x^{t+1}_{[J_t]}\right) \\
&= B(x, x^t) - B(x, x^{t+1})
\end{aligned}$$

from which (d) follows. □

Using convexity and Lemma 1 to bound $f(x^t) - f(x)$ and summing each side over $t = 1, \dots, T$, we conclude

$$\begin{aligned}
&T\mathbb{E}[f(\bar{x}_T) - f(x)] \\
&\leq \mathbb{E}\left[\sum_{t=1}^T (f(x^t) - f(x))\right] \\
&\leq \sum_{t=1}^T \mathbb{E}\left[\frac{1}{p^t_{J_t}} \left\langle G_{J_t}(x^t), x^t - x\right\rangle\right] \\
&\leq \frac{B(x, x^1)}{\alpha_x} + \frac{\alpha_x}{2}\sum_{t=1}^T \mathbb{E}\frac{1}{(p^t_{J_t})^2}\left\|G_{[J_t]}(x^t)\right\|^2_{J_t,*} \\
&\overset{(*)}{\leq} \frac{R^2}{\alpha_x} + \frac{\alpha_x}{2}\sum_{t=1}^T \mathbb{E}\left[\sum_{j=1}^b \frac{\left\|G_{[j]}(x^t)\right\|^2_{j,*}}{p^t_j}\right]
\end{aligned}$$

where in step $(*)$ we used the tower law $\mathbb{E}[\cdot] = \mathbb{E}[\mathbb{E}_t[\cdot]]$. The second result follows from the bound $p^t_j \geq p_{\min}$.

A.2. Proof of Proposition 2

Denote by E_k the indices in epoch k. Let K be the total number of epochs used in Algorithm 2. Applying Lemma 1 to each of the epochs, we obtain

$$\begin{aligned}
&\sum_{t=1}^T \frac{1}{p^t_{J_t}} \left\langle G_{J_t}(x^t), x^t - x\right\rangle \\
&= \sum_{k=1}^K \sum_{t\in E_k} \frac{1}{p^t_{J_t}} \left\langle G_{J_t}(x^t), x^t - x\right\rangle \\
&\leq \sum_{k=1}^K \left\{\frac{R^2}{\alpha_{x,k}} + \sum_{t\in E_k} \frac{\alpha_{x,k}}{2\left(p^t_{J_t}\right)^2}\left\|G_{[J_t]}(x^t)\right\|^2_{J_t,*}\right\} \\
&\overset{(a)}{\leq} \sum_{k=1}^K \left\{\frac{R^2}{\alpha_{x,k}} + \frac{\alpha_{x,k}}{2}\left(4^k + \frac{L^2}{bp^2_{\min}}\right)\right\}
\end{aligned}$$

$$\leq \sqrt{2}R\sum_{k=1}^{K}\sqrt{4^k+\frac{L^2}{bp_{\min}^2}} \overset{(b)}{\leq} \sqrt{2}R\sum_{k=1}^{K}\left(2^k+\frac{L}{p_{\min}\sqrt{b}}\right)$$
$$= \sqrt{2}R\left(2^{K+1}-2+K\frac{L}{p_{\min}\sqrt{b}}\right)$$
$$\overset{(c)}{\leq} \sqrt{2}R\Bigg(4\left(\sum_{t=1}^{T}\frac{\left\|G_{[J_t]}(x^t)\right\|_{J_t,*}^2}{\left(p_{J_t}^t\right)^2}\right)^{\frac{1}{2}} + \frac{L}{p_{\min}\sqrt{b}\log 4}\log\left(4\sum_{t=1}^{T}\frac{\left\|G_{[J_t]}(x^t)\right\|_{J_t,*}^2}{\left(p_{J_t}^t\right)^2}\right)\Bigg)$$

where (a) follows from noting that $\sum_{t\in E_k}\left\|G_{[J_t]}(x^t)\right\|_{J_t,*}^2/\left(p_{J_t}^t\right)^2 \leq 4^k+\frac{L^2}{bp_{\min}^2}$, (b) from $\sqrt{a+b}\leq\sqrt{a}+\sqrt{b}$. In step (c), we used the stopping condition of the $K-1$th epoch

$$\sum_{t=1}^{T}\frac{\left\|G_{[J_t]}(x^t)\right\|_{J_t,*}^2}{\left(p_{J_t}^t\right)^2}\geq 4^{K-1}.$$

Taking expectations on both sides, we have

$$T(\mathbb{E}[f(\bar{x}_T)]-f(x))$$
$$\leq \sum_{t=1}^{T}(\mathbb{E}[f(x^t)]-f(x)) \leq \sum_{t=1}^{T}\mathbb{E}\left[\frac{1}{p_{J_t}^t}\left\langle G_{J_t}(x^t),x^t-x\right\rangle\right]$$
$$\leq \sqrt{2}R\mathbb{E}\Bigg[4\left(\sum_{t=1}^{T}\frac{\left\|G_{[J_t]}(x^t)\right\|_{J_t,*}^2}{\left(p_{J_t}^t\right)^2}\right)^{\frac{1}{2}} + \frac{L}{p_{\min}\sqrt{b}\log 4}\log\left(4\sum_{t=1}^{T}\frac{\left\|G_{[J_t]}(x^t)\right\|_{J_t,*}^2}{\left(p_{J_t}^t\right)^2}\right)\Bigg]$$
$$\overset{(a)}{\leq} 4\sqrt{2}R\left(\mathbb{E}\left[\sum_{t=1}^{T}\frac{\left\|G_{[J_t]}(x^t)\right\|_{J_t,*}^2}{\left(p_{J_t}^t\right)^2}\right]\right)^{\frac{1}{2}} + \frac{\sqrt{2}RL}{p_{\min}\sqrt{b}\log 4}\log\left(4\mathbb{E}\left[\sum_{t=1}^{T}\frac{\left\|G_{[J_t]}(x^t)\right\|_{J_t,*}^2}{\left(p_{J_t}^t\right)^2}\right]\right)$$
$$\overset{(b)}{=} 4\sqrt{2}R\left(\sum_{t=1}^{T}\mathbb{E}\left[\sum_{j=1}^{b}\frac{\left\|G_{[j]}(x^t)\right\|_{j,*}^2}{p_j^t}\right]\right)^{\frac{1}{2}} + \frac{\sqrt{2}RL}{p_{\min}\sqrt{b}\log 4}\log\left(4\sum_{t=1}^{T}\mathbb{E}\left[\sum_{j=1}^{b}\frac{\left\|G_{[j]}(x^t)\right\|_{j,*}^2}{p_j^t}\right]\right)$$

where (a) follows from Jensen's inequality and (b) from the tower law.

A.3. Proof of Theorem 3

From Proposition 2, it suffices to show that Algorithm 3 with α_p attains the regret bound

$$\sup_{p\in\mathcal{P}}\sum_{t=1}^{T}\mathbb{E}\left[\sum_{j=1}^{b}\frac{\left\|G_{[j]}(x^t)\right\|_{j,*}^2}{p_j^t}-\frac{\left\|G_{[j]}(x^t)\right\|_{j,*}^2}{p_j}\right] \leq \frac{L^2}{p_{\min}^2}\sqrt{\frac{2T\log b}{b}}. \tag{11}$$

Note that the bandit updates in Algorithm 3 correspond to mirror descent updates with $\psi_P(p)=\sum_{j=1}^{b}p_j\log p_j$ and $\psi_P^\star(u)=\sum_{j=1}^{b}\exp(u_j-1)$ (Beck and Teboulle, 2003). We wish show that the bandit mirror descent updates in Algorithm 3 achieves regret scaling as $\sqrt{T}$. To this end, we first state a standard result for mirror descent algorithms. See for example, Cesa-Bianchi and Lugosi (2006, Ch.11) or Bubeck and Cesa-Bianchi (2012, Thm 5.3). We outline the proof for completeness in Appendix A.4.

Lemma 2 (Bubeck and Cesa-Bianchi (2012), Thm 5.3)**.** *The following bound holds for Algorithm 3 for any $p\in\mathcal{P}$.*

$$\sum_{t=1}^{T}\left\langle\widehat{\ell}_t(x^t),p^t-p\right\rangle \leq \frac{B_{\psi_P}(p,p_1)}{\alpha_p}+\frac{1}{\alpha_p}\sum_{t=1}^{T}B_{\psi_P^\star}\left(\nabla\psi_P(p^t)-\alpha_p\widehat{\ell}_t(x^t),\nabla\psi_P(p^t)\right) \tag{12}$$

A straightforward calculation gives that

$$B_{\psi_P^\star}\left(\nabla\psi_P(p^t)-\alpha_p\widehat{\ell}_t(x^t),\nabla\psi_P(p^t)\right)$$
$$=\sum_{j=1}^{b}p_j^t\left(\exp(-\alpha_p\widehat{\ell}_{t,j}(x^t))+\alpha_p\widehat{\ell}_{t,j}(x^t)-1\right)$$
$$\leq \frac{\alpha_p^2}{2}p_{J_t}^t\widehat{\ell}_{t,J_t}(x^t)^2 \leq \frac{L^4\alpha_p^2}{2p_{\min}^4b^2p_{J_t}^t} \tag{13}$$

since $e^z-z-1\leq z^2$ for $z\leq 0$ where we used the fact that $\widehat{\ell}\geq 0$. From convexity, we have for $p\in\mathcal{P}$

$$\sum_{t=1}^{T}\mathbb{E}\left[\sum_{j=1}^{b}\frac{\left\|G_{[j]}(x^t)\right\|_{j,*}^2}{p_j^t}-\frac{\left\|G_{[j]}(x^t)\right\|_{j,*}^2}{p_j}\right]$$
$$\overset{(a)}{=}\sum_{t=1}^{T}\mathbb{E}\left[\sum_{j=1}^{b}\frac{\left\|G_{[j]}(x^t)\right\|_{j,*}^2}{p_j^t}-\frac{\left\|G_{[j]}(x^t)\right\|_{j,*}^2}{p_j}\right] + \sum_{t=1}^{T}\mathbb{E}\left[\left\langle p^t-p,\frac{L^2}{bp_{\min}^2}\mathbb{1}\right\rangle\right]$$
$$\overset{(b)}{\leq}\sum_{t=1}^{T}\mathbb{E}\left[\sum_{j=1}^{b}\left(-\frac{\left\|G_{[j]}(x^t)\right\|_{j,*}^2}{(p_j^t)^2}+\frac{L^2}{bp_{\min}^2}\right)(p_j^t-p_j)\right]$$

$$\overset{(c)}{=} \sum_{t=1}^{T} \mathbb{E}\left[\left\langle \widehat{\ell}_t(x^t), p^t - p \right\rangle\right]$$
$$\overset{(d)}{\leq} \frac{\log b}{\alpha_p} + \frac{\alpha_p}{2} \sum_{t=1}^{T} \frac{L^4}{p_{\min}^4 b^2} \mathbb{E}\left[\frac{1}{p^t_{J_t}}\right]$$
$$\overset{(e)}{\leq} \frac{\log b}{\alpha_p} + \frac{\alpha_p}{2} \frac{L^4}{p_{\min}^4 b} T$$

where in (a) we used the fact that p, p^t are probabilities and in (b) we used convexity of $g(p) = \sum_{j=1}^{b} \frac{1}{p_j} \left(\left\|G_{[j]}(x^t)\right\|_{j,*}\right)^2 + p^\top \mathbb{1}$. To obtain (d), we used $D_{\mathrm{kl}}(p \| p_1) \leq \log b$, Lemma 2 and (13). Finally, step (e) follows from tower law. Setting $\alpha_p = \frac{p_{\min}^2}{L^2}\sqrt{\frac{2b\log b}{T}}$, we obtain

$$\max_{p \in \mathcal{P}} \sum_{t=1}^{T} \mathbb{E}\left[\sum_{j=1}^{b} \frac{\|G_j(x^t)\|_*^2}{p_j^t} - \frac{\|G_j(x^t)\|_*^2}{p_j}\right] \leq \frac{L^2}{p_{\min}^2}\sqrt{\frac{2T\log b}{b}}. \tag{14}$$

Using this in the bound of Proposition 2, we obtain the desired result.

A.4. Proof of Lemma 2

From Algorithm 3, we have

$$\begin{aligned}
&\alpha_p \widehat{\ell}_t(x^t)^\top (p^t - p) \\
&= \left(\nabla\psi_P(w^{t+1}) - \nabla\psi_P(p^t)\right)^\top (p^t - p) \\
&= B_{\psi_P}(p, p^t) + B_{\psi_P}(p^t, w^{t+1}) - B_{\psi_P}(p, w^{t+1}).
\end{aligned} \tag{15}$$

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

$$\begin{aligned}
&B_{\psi_P}(p, w^{t+1}) \geq B_{\psi_P}(p, p_{t+1}) + B_{\psi_P}(p_{t+1}, w^{t+1}) \\
&\equiv \left(\nabla\psi_P(p) - \nabla\psi_P(w^{t+1})\right)^\top (p - p_{t+1}) \geq 0.
\end{aligned}$$

The latter inequality is just the optimality condition for $p_{t+1} = \operatorname{argmin}_{p \in \mathcal{P}} B_{\psi_P}(p, w^{t+1})$. Applying the first equality in (15) and summing for $t = 1, \ldots, T$, we obtain

$$\begin{aligned}
&\alpha_p \sum_{t=1}^{T} \widehat{\ell}_t(x^t)^\top (p^t - p) \\
&\leq B_{\psi_P}(p, p_1) - B_{\psi_P}(p, p_{T+1}) \\
&\quad + \sum_{t=1}^{T} \left(B_{\psi_P}(p^t, w^{t+1}) - B_{\psi_P}(p_{t+1}, w^{t+1})\right) \\
&\leq B_{\psi_P}(p, p_1) + \sum_{t=1}^{T} B_{\psi_P}(p^t, w^{t+1}) \\
&= B_{\psi_P}(p, p_1) + \sum_{t=1}^{T} B_{\psi_P^\star}(\nabla\psi_P(w^{t+1}), \nabla\psi_P(p^t)).
\end{aligned}$$

Now, noting that $\nabla\psi_P(w^{t+1}) = \nabla\psi_P(p^t) + \alpha_p \widehat{\ell}_t(x^t)$, we obtain the result.

A.5. Proof of Proposition 5

Let us first solve for the KKT conditions of the following minimization problem

$$\begin{aligned}
&\underset{p \in \mathbb{R}^n}{\text{minimize}} \sum_{j=1}^{b} \frac{L_j^2}{p_j} \\
&\text{subject to } \; p^\top \mathbb{1} = 1, p \geq p_{\min}.
\end{aligned}$$

Taking the first order conditions for the Lagrangian

$$\mathcal{L}(p, \eta, \theta) = \sum_{j=1}^{b} \frac{L_j^2}{p_j} - \eta(p^\top - \mathbb{1}) - \theta^\top (p - p_{\min}\mathbb{1}),$$

we have

$$p_j = \frac{L_j}{\sqrt{|\eta + \theta_j|}} = \begin{cases} \frac{L_j}{\sqrt{|\eta|}} & \text{if } \; L_j \geq \sqrt{|\eta|} p_{\min} \\ p_{\min} & \text{otherwise} \end{cases}$$

where the last equality follows from complementary slackness. Let $I := \left\{j : L_j \geq \sqrt{|\eta|} p_{\min}\right\}$. Plugging p_j's into the equality constraint, we get

$$\sum_{j=1}^{b} p_j = \frac{1}{\sqrt{|\eta|}} \sum_{j \in I} L_j + (b - |I|) p_{\min} = 1$$

so that

$$\sqrt{|\eta|} = \frac{1}{1 - (b - |I|)p_{\min}} \sum_{j \in I} L_j. \tag{16}$$

Then, by plugging in p_j into the objective and using the above identity for $|\eta|$ yields

$$\begin{aligned}
\sum_{j=1}^{b} \frac{L_j^2}{p_j} &= \sqrt{|\eta|} \sum_{j \in I} L_j + \frac{1}{p_{\min}} \sum_{j \in I^c} L_j^2 \\
&= \left((1 - (b - |I|)p_{\min}\right)|\eta| + \frac{1}{p_{\min}} \sum_{j \in I^c} L_j^2.
\end{aligned} \tag{17}$$

Now, let $L_j^2 = cj^{-\alpha}$ so that $I = \left\{j : j \leq \left(\frac{c}{|\eta| p_{\min}^2}\right)^{\frac{1}{\alpha}}\right\}$. When $\alpha \in [2, \infty)$, we have

$$\sum_{j=1}^{|I|} L_j = c \sum_{j=1}^{|I|} j^{-\alpha/2} = O(\log |I|) = O\left(\log\left(\frac{b}{|\eta|}\right)\right).$$

From (16), we have $|\eta| = O(\log^2 b)$ and $|I| = O\left(\left(\frac{b^2}{\log^2 b}\right)^{\frac{1}{\alpha}}\right) = o(b)$. Using these in the bound (17), we obtain

$$\sum_{j=1}^{b} \frac{L_j^2}{p_j} \leq O(\log^2 b) + \frac{b}{\delta} c (b - |I|)^{1-\alpha} = O(\log^2 b)$$

which was the result for $\alpha \in [2, \infty)$.

When $\alpha \in (1, 2)$, we have

$$\sum_{j=1}^{|I|} L_j = c \sum_{j=1}^{|I|} j^{-\alpha/2} = \Theta\left(|I|^{1-\alpha/2}\right) = \Theta\left(\left(\frac{b^2}{|\eta|}\right)^{\frac{1}{\alpha}-\frac{1}{2}}\right)$$

so that from (16), $|\eta| = \Theta(b^{2-\alpha})$ and $|I| = \Theta(b)$. Using these in the bound (17) for the objective, we obtain

$$\sum_{j=1}^{b} \frac{L_j^2}{p_j} = \Theta(b^{2-\alpha})$$

which gives the second result.

B. Procedures for Efficient Updates

B.1. Proof of Proposition 4

We are interested in finding the solution to the projection problem

$$\underset{q}{\text{minimize}} \left\{ D_{\text{kl}}\left(q \| w\right) : q^\top \mathbb{1} = 1, q \geq p_{\min} \right\}$$

where $w \in \mathbb{R}_+^b$ is a probability vector with its value at J-th element shrunken down. Let the Lagrangian be

$$\mathcal{L}(q, \eta, \theta) = \sum_{j=1}^{b} q_j \log \frac{q_j}{w_j} - \eta(q^\top \mathbb{1} - 1) - \theta^\top (q - p_{\min}\mathbb{1})$$

where $\theta \in \mathbb{R}_+^b$. Writing down the first order conditions for q, we have

$$\log \frac{q}{w} + \mathbb{1} - \eta \mathbb{1} - \theta = 0$$

where $\eta \in \mathbb{R}$ is the dual variable for $q^\top \mathbb{1} = 1 - bp_{\min}$ and $\theta \in \mathbb{R}_+^b$ is the dual variable for $q \geq 0$. From strict complementarity, we have that

$$q_j = \left(w_j \exp(\eta - 1) - p_{\min}\right)_+ + p_{\min}$$

Then, it suffices to solve for η^* such that

$$\sum_{j \in I(\eta)} \left(w_j \exp(\eta - 1) - p_{\min}\right) = 1 - bp_{\min} \tag{18}$$

where $I(\eta) := \{1 \leq j \leq b : w_j \exp(\eta - 1) \geq p_{\min}\}$.

Now, assume that w is sorted and stored in a binary tree up to a constant s. At each node, we also store the sum of the values in the right/left subtree. Denote by $w_{(1)} \geq \ldots \geq w_{(b)}$ the sorted values of w. Let

$$f(j) := w_{(j)}(1 - (b - j)p_{\min}) - p_{\min} \sum_{i=1}^{j} w_{(j)}$$

and $J^* := \max\{1 \leq j \leq b : f(j) \geq 0\}$. We first show that the optimal dual variable η^* is given by

$$e^{\eta^* - 1} = \frac{1 - (b - J^*)p_{\min}}{\sum_{i=1}^{J^*} w_{(j)}}.$$

To this end, let $J(\eta) := |I(\eta)|$. For the optimal dual variable η^*, we have that $J(\eta^*)$ satisfies

$$w_{(J(\eta^*))} \exp(\eta^* - 1) \geq p_{\min} \quad \text{and}$$
$$w_{(J(\eta^*)+1)} \exp(\eta^* - 1) < p_{\min}. \tag{19}$$

Now, note that $e^{\eta^* - 1}$ satisfies

$$e^{\eta^* - 1} = \frac{1 - (b - J(\eta^*))p_{\min}}{\sum_{i=1}^{J(\eta^*)} w_{(j)}}$$

from $p^\top \mathbb{1} = 1$. Plugging this back into (19), we have that $f(J(\eta^*)) \geq 0$ and $f(J(\eta^*)) < 0$. Since $f(j)$ is nonincreasing in j, it follows that $J(\eta^*) = J^*$ which show the claim.

Next, we show that $J^* = b - 1$ or b are the only possibilities.

1. Case $w_J < p_{\min}$: Here, $w_J = w_{(b)}$ since $p \geq p_{\min}$. Noting that $w_j = p_j$ for $j \neq J$, if

   $$w_{(b)} \frac{1}{\sum_{j=1}^{b} w_{(j)}} = w_J \frac{1}{1 - p_J + w_J} \geq p_{\min},$$

   then $J^* = b$ and $e^{\eta^* - 1} = \frac{1}{\sum_{j=1}^{b} w_j}$. Otherwise, we surely have that

   $$w_{(b-1)} \frac{1 - p_{\min}}{\sum_{j=1}^{b-1} w_{(j)}} = w_{(b-1)} \frac{1 - p_{\min}}{1 - p_J} \geq w_{(b-1)} \geq p_{\min}$$

   since $p_j = w_j$ for $j \neq J$ and $p_J \geq p_{\min}$. It follows that $J^* = b - 1$ and $e^{\eta^* - 1} = \frac{1 - p_{\min}}{1 - p_J}$.

2. Case $w_J \geq p_{\min}$: since

   $$w_{(b)} \frac{1}{\sum_{j=1}^{b} w_{(j)}} \geq w_{(b)} \geq p_{\min},$$

   we have $J^* = b$ and $e^{\eta^* - 1} = \frac{1}{\sum_{j=1}^{b} w_j}$.

Combining the two cases and noting that

$$q = \left(w e^{\eta^* - 1} - p_{\min}\right)_+ + p_{\min},$$

we have

$$q = \begin{cases} \frac{1}{1 - p_J + w_J} w & \text{if } w_J \geq \frac{p_{\min}(1 - p_J)}{1 - p_{\min}} \\ \left(\frac{1 - p_{\min}}{1 - p_J} w - p_{\min}\right)_+ + p_{\min} & \text{otherwise} \end{cases}$$

Since $w_j \geq p_{\min}$ for $j \neq J$, result follows.

B.2. Update

We recapitulate that a key characteristic of our tree implementation is that at each node, in addition to the value v, we store $\text{sum}_l, \text{sum}_r$, the sum of elements in the left/right subtree. Below, we give an algorithm that modifies an element of the tree and updates $\text{sum}_l, \text{sum}_r$ of its parents accordingly. All node comparisons are based on their corresponding index values.

Algorithm 6 Update

w_{new}, J
Set value at index J to w_{new}
Initialize node with that of index J
while node.parent ! = NULL **do**
  if node is a left child **then**
    node.parent.sum$_l$ $\leftarrow$ node.parent.sum$_r$
      $+w_{new} - w_{old}$
  else
    node.parent.sum$_r$ $\leftarrow$ node.parent.sum$_r$
      $+w_{new} - w_{old}$

node $\leftarrow$ node.parent

B.3. Sampling

For completeness, we give the pseudo-code for sampling from the BST in $O(\log b)$.

Algorithm 7 Sample Tree

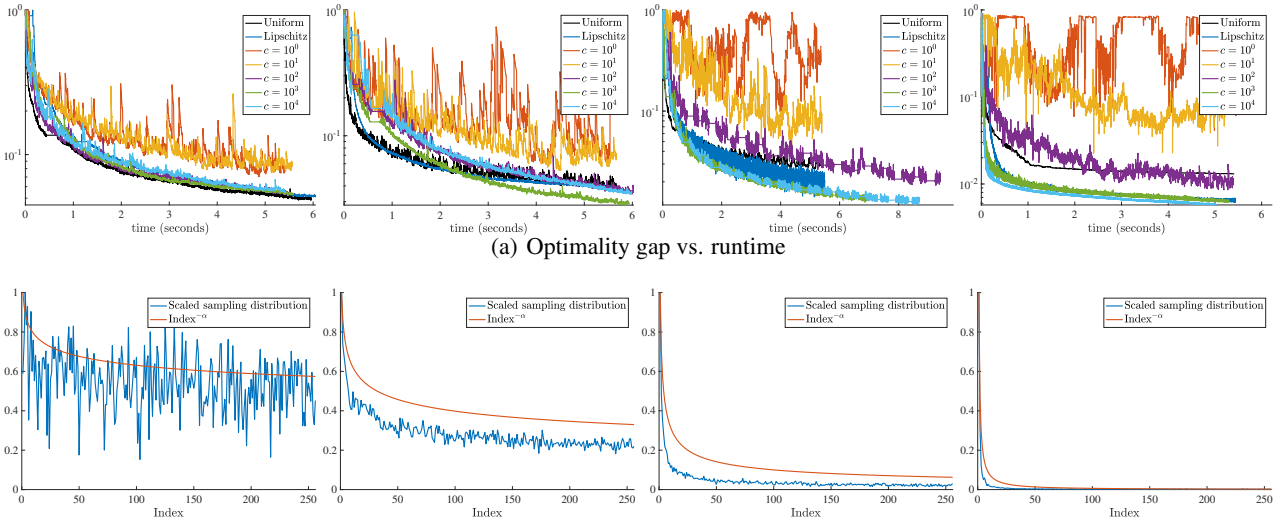
$coin \leftarrow$ Uniform(0,1), node $\leftarrow$ root
$coin \leftarrow scale \cdot coin$
while node is not a leaf **do**
  if $coin <$ node.v **then**
    Return node
  elseif $coin <$ node.v + node.sum$_l$ **then**
    $coin \leftarrow coin -$ node.v
    node $\leftarrow$ node.left
  else
    $coin \leftarrow coin -$ node.$v -$ node.sum$_l$
    node $\leftarrow$ node.right
return node

C. Detailed synthetic experiments

Here we present further experiments for Sections 4.1 and 4.2. Namely, Figures 5 and 6 shows experiments over more α and more values of c for each α. This more detailed setup illustrates the phase change in behavior: from emulating uniform sampling at small α to learning and taking advantage of structure at large α. Interestingly, we see that even though we are able to learn structure in sampling over examples (Figure 6), the magnitude of improvement over uniform sampling is smaller than in the coordinate sampling case (Figure 5). We postulate that this is due to the following effect: for an ϵ-optimality gap, SGD requires $O(R^2L^2/\epsilon^2)$ iterations, whereas $O(R^2L^2b/\epsilon^2)$ iterations. Since our bandit algorithm requires roughly $O(b \log b)$ iterations to learn structure, it has more time to take advantage of this structure in coordinate sampling before reaching a given optimality gap. For SGD, our adaptive algorithm does better than leverage scores when $\alpha = 2.2$ which is a result of learning a more aggressive sampling distribution.

Figure 7 analyzes the effects of stepsize on the performance of our algorithm. Specifically, we consider the same synthetic dataset as in Section 4.1, and we fix the number of iterations to 10^5. Varying the stepsize parameter β, we track the optimality gap at the end of the procedure for our method as well as uniform sampling. Although the sensitivity to stepsize (the width of the bowl) appears the same in both approaches, our method is able to handle larger stepsizes (our "bowl" is shifted to the right) and learns more efficiently for a given stepsize (our "bowl" is deeper) at larger α. Importantly, we achieve these advantages solely by tilting the sampling distribution to match the underlying data's structure.

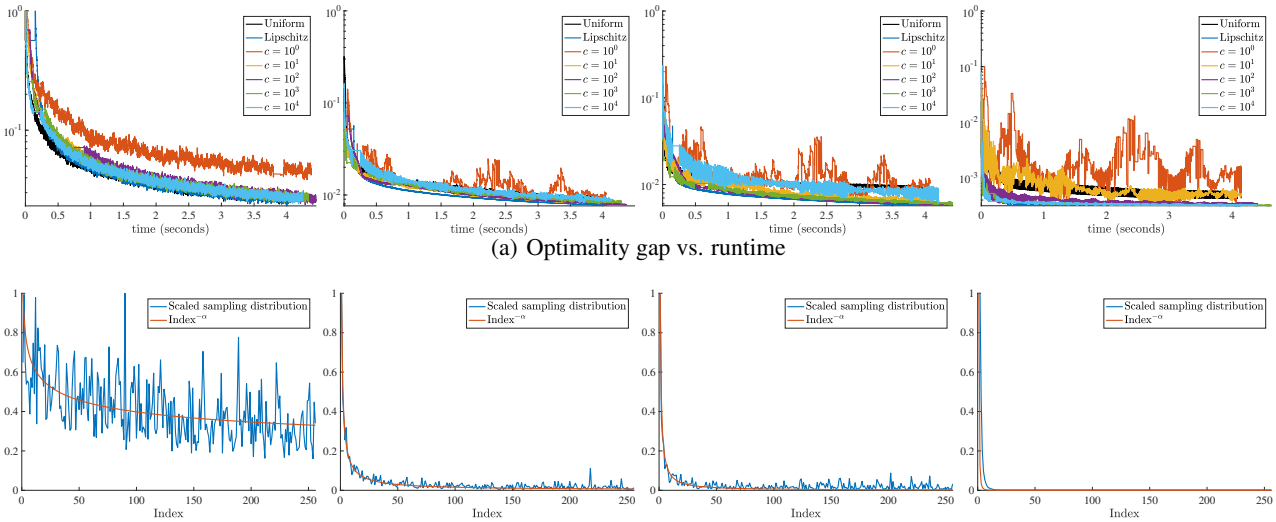
Finally, we consider using the Gauss-Southwell rule for the experiments in Section 4.1; similar to using blockwise Lipschitz constants, the Gauss-Southwell rule descends on the coordinate with the largest gradient magnitude at every iteration. This method is inefficient except for specialized loss functions, as seen in Figure 8.



(a) Optimality gap vs. runtime

(b) Learned probability distribution compared to $j^{-\alpha}$. We use c that resulted in the best performance.

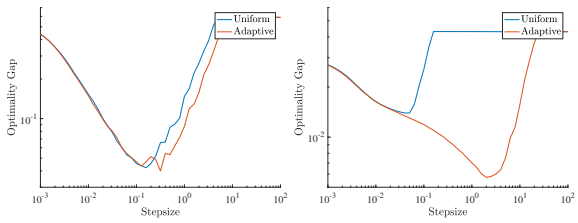
Figure 5. Adaptive coordinate descent (left to right: $\alpha = 0.2, 0.4, 1.0, 2.2$)



(a) Optimality gap vs. runtime

(b) Learned probability distribution compared to $j^{-\alpha}$. We use c that resulted in the best performance.

Figure 6. Adaptive SGD (left to right: $\alpha = 0.4, 1.8, 2.2, 6.0$)



(a) $\alpha = 0.4$ (b) $\alpha = 2.2$

Figure 7. Optimality gap vs. stepsize after 10^5 iterations

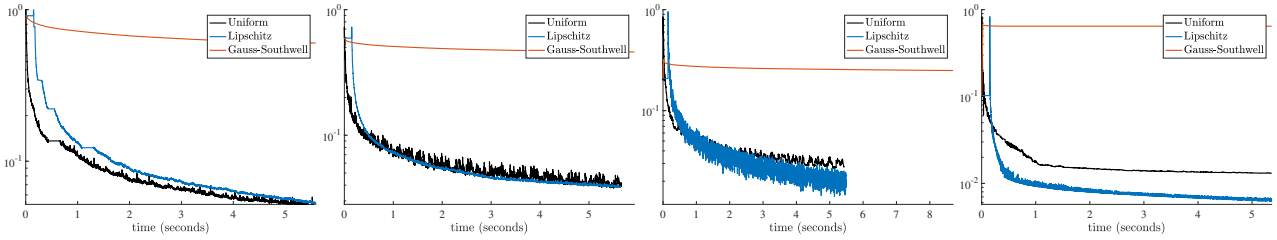


Figure 8. Optimality gap vs. runtime for the same experiments as in Figure 5 with the Gauss-Southwell rule (left to right: $\alpha = 0.2, 0.4, 1.0, 2.2$)