Adaptive Importance Sampling for Finite-Sum Optimization and Sampling with Decreasing Step-Sizes

Ayoub El Hanchi
McGill University
ayoub.elhanchi@mail.mcgill.ca

David A. Stephens
McGill University
david.stephens@mcgill.ca

Abstract

Reducing the variance of the gradient estimator is known to improve the convergence rate of stochastic gradient-based optimization and sampling algorithms. One way of achieving variance reduction is to design importance sampling strategies. Recently, the problem of designing such schemes was formulated as an online learning problem with bandit feedback, and algorithms with sub-linear *static* regret were designed. In this work, we build on this framework and propose *Avare*, a simple and efficient algorithm for adaptive importance sampling for finite-sum optimization and sampling with decreasing step-sizes. Under standard technical conditions, we show that *Avare* achieves $\mathcal{O}(T^{2/3})$ and $\mathcal{O}(T^{5/6})$ *dynamic* regret for SGD and SGLD respectively when run with $\mathcal{O}(1/t)$ step sizes. We achieve this dynamic regret bound by leveraging our knowledge of the dynamics defined by the algorithm, and combining ideas from online learning and variance-reduced stochastic optimization. We validate empirically the performance of our algorithm and identify settings in which it leads to significant improvements.

1 Introduction

Functions $f : \mathbb{R}^d \to \mathbb{R}$ of the form:

$$f(x) = \sum_{i=1}^{N} f_i(x) \tag{1}$$

are prevalent in modern machine learning and statistics. Important examples include the empirical risk in the empirical risk minimization framework, [1] or the log-posterior of an exchangeable Bayesian model. When N is large, the preferred methods for solving the resulting optimization or sampling problem usually rely on stochastic estimates of the gradient of f, using variants of stochastic gradient descent (SGD) [1] or stochastic gradient Langevin dynamics (SGLD) [2]:

$$x_{t+1}^{SGD} = x_t^{SGD} - \alpha_t N \nabla f_{I_t}(x_t^{SGD}) \tag{2}$$

$$x_{t+1}^{SGLD} = x_t^{SGLD} - \alpha_t N \nabla f_{I_t}(x_t^{SGLD}) + \xi_t \qquad \xi_t \sim \mathcal{N}(0, 2\alpha_t) \tag{3}$$

where $\{\alpha_t\}_{t=1}^T$ is a sequence of step-sizes, and the index I_t is sampled uniformly from $[N]$, making $N\nabla f_{I_t}(x)$ an unbiased estimator of the gradient of f. We use $\{x_t\}_{t=1}^T$ to refer to either sequence when we do not wish to distinguish between them. It is well known that the quality of the answers given by these algorithms depends on the (trace of the) variance of the gradient estimator, and considerable efforts have been made to design methods that reduce this variance.

[1] Up to a normalizing factor of $\frac{1}{N}$ which does not affect the optimization

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

In this paper, we focus on the importance sampling approach to achieving variance reduction. At each iteration, the algorithm samples I_t according to a specified distribution p^t, and estimates the gradient using:

$$\hat{g}^t := \frac{1}{p^t_{I_t}} \nabla f_{I_t}(x_t) \tag{4}$$

It is immediate to verify that $\hat{g}^t$ is an unbiased estimator of $g^t := \nabla f(x_t)$. By cleverly choosing the distributions p^t, one can achieve significant variance reduction (up to a factor of N) compared to the estimator based on uniform sampling. Unfortunately, computing the variance-minimizing distributions at each iteration requires the knowledge of the Euclidean norm of all the individual gradients $g^t_i := \nabla f_i(x_t)$ at each iteration, making it unpractical [3, 4].

Many methods have been proposed that attempt to construct sequences of distributions $\{p^t\}_{t=1}^T$ that result in efficient estimators [3, 4, 5, 6, 7, 8, 9]. Of particular interest to us, the task of designing such sequences was recently cast as an online learning problem with bandit feedback [10, 11, 12, 13]. In this formulation, one attempts to design algorithms with sub-linear *expected* static regret, which is defined as:

$$\text{Regret}_S(T) := \sum_{t=1}^{T} c_t(p^t) - \min_{p \in \Delta} \sum_{t=1}^{T} c_t(p)$$

where Δ denotes the probability simplex in $\mathbb{R}^N$, and $c_t(p)$ is the trace of the covariance matrix of the gradient estimator (4), which is easily shown to be:

$$c_t(p) := \sum_{i=1}^{N} \frac{1}{p_i} \left\| g^t_i \right\|_2^2 - \left\| g^t \right\|_2^2 \tag{5}$$

Note that the second term cancels in the definition of regret, and we omit it in the rest of our discussion. In this formulation, and to keep the computational load manageable, one has only access to partial feedback in the form of the norm of the I^{th}_t gradient, and not to the complete cost function (5). Under the assumption of uniformly bounded gradients, the best result in this category can be found in [12] where an algorithm with $\tilde{O}(T^{2/3})$ static regret is proposed. A more difficult but more natural performance measure that makes the attempt to approximate the optimal distributions explicit is the *dynamic* regret, defined by:

$$\text{Regret}_D(T) := \sum_{t=1}^{T} c_t(p^t) - \sum_{t=1}^{T} \min_{p \in \Delta} c_t(p) \tag{6}$$

Guarantees with respect to the expected dynamic regret are more difficult to obtain, and require that the cost functions $c_t(p)$ do not change too rapidly with respect to some reasonable measure of variation. See [14, 15, 16, 17, 18] for examples of such measures and the corresponding regret bounds for general convex cost functions.

In this work, we propose *Avare*, an algorithm that achieves sub-linear dynamic regret for both SGD and SGLD when the sequence of step-sizes $\{\alpha_t\}_{t=1}^T$ is decreasing. The name *Avare* is derived from adaptive variance minimization. Specifically, our contributions are as follows:

- We show that *Avare* achieves $\mathcal{O}(T^{2/3})$ and $\mathcal{O}(T^{5/6})$ dynamic regret for SGD and SGLD respectively when α_t is $\mathcal{O}(1/t)$.
- We propose a new mini-batch estimator that combines the benefits of sampling without replacement and importance sampling while preserving unbiasedness.
- We validate empirically the performance of our algorithm and identify settings in which it leads to significant improvements.

We would like to point out that while the decreasing step size requirement might seem restrictive, we argue that for SGD and SGLD, it is the right setting to consider for variance reduction. Indeed, it is well known that under suitable technical conditions, both algorithms converge to their respective solutions exponentially fast in the early stages. Variance reduction is primarily useful at later stages when the noise from the stochastic gradient dominates. In the absence of control variates, one is forced to use decreasing step-sizes to achieve convergence. This is precisely the regime we consider.

2 Related work

It is easy to see that the cost functions (5) are convex over the probability simplex. A celebrated algorithm for convex optimization over the simplex is entropic descent [19], an instance of mirror descent [20] where the Bregman divergence is taken to be the relative entropy. A slight modification of this algorithm for online learning is the EXP3 algorithm [21] which mixes the iterates of entropic descent with a uniform distribution to avoid the assignment of null probabilities. See [22, 23, 24] for a more thorough discussion of online learning (also called online convex optimization) in general and variants of this algorithm in particular.

Since we are working over the simplex, the EXP3 algorithm is a natural choice. This is the approach taken in [10] and [11], although strictly speaking neither is able to prove sub-linear static regret bounds. The difficulty comes from the fact that the norm of the gradients of the cost functions (5) explode to infinity on the boundary of the simplex. This is amplified by the use of stochastic gradients which grow as $1/p_{min}^5$ in expectation, making it very difficult to reach regions near the boundary. Algorithms based on entropic descent for dynamic regret problems also exist, including the fixed share algorithm and projected mirror descent [25, 26, 27, 28]. Building on these algorithms, and by artificially making the cost functions strongly-convex to allow the use of decreasing step-sizes, we were only able to show $\tilde{O}(T^{7/8})$ dynamic regret using projected mirror descent for SGD with $O(1/t)$ decreasing step sizes and uniformly bounded gradients.

The approach taken in [12] is qualitatively different and is based on the follow-the-regularized-leader (FTRL) scheme. By solving the successive minimization problems stemming from the FTRL scheme analytically, the authors avoid the above-mentioned issue of exploding gradients. The rest of their analysis relies on constructing an unbiased estimate of the cost functions (5) using only the partial feedback $\left\|g_{I_t}^t\right\|_2^2$, and probabilistically bounding the deviation of the estimated cost functions from the true cost functions using a martingale concentration inequality. The final result is an algorithm that enjoys an $\tilde{O}(T^{2/3})$ static regret bound.

Our approach is similar to the one taken in [12] in that we work directly with the cost functions and not their gradients to avoid the exploding gradient issue. Beyond this point however, our analysis is substantially different. While we are still working within the online learning framework, our analysis is more closely related to the analysis of variance-reduced stochastic optimization algorithms that are based on control variates. In particular, we study a Lyapunov-like functional and show that it decreases along the trajectory of the dynamics defined by the algorithms. This yields simpler and more concise proofs, and opens the door for a unified analysis.

3 Algorithm

Most of the literature on online convex optimization with dynamic regret relies on the use of the gradients of the cost functions [14, 15, 16, 17, 18]. However, as explained in the previous section, such approaches are not viable for our problem. Unfortunately, the regret guarantees obtained from the FTRL scheme for static regret used in [12] do not directly translate into guarantees for dynamic regret.

We start by presenting the high level ideas that go into the construction of our algorithm. We then present our algorithm in explicit form, and discuss its implementation and computational complexity.

3.1 High level ideas

The simplest update rule one might consider when working with dynamic regret is a natural extension of the follow-the-leader approach:

$$p^{t+1} := \operatorname*{argmin}_{p \in \Delta} \{c_t(p)\} \tag{7}$$

Intuitively, if the cost functions do not change too quickly, then it is reasonable to expect good performance from this algorithm. In our case however, we do not have access to the full cost function. The traditional way of dealing with this problem is to build unbiased estimates of the cost functions using the bandit feedback, and then bounding the deviation of these unbiased estimates from the true cost functions. While this might work, we consider here a different approach based on constructing surrogate cost functions that are not necessarily unbiased estimates of the true costs.

For each $i \in [N]$, denote by h_i^t the last observed gradient of f_i at time t, with h_i^1 initialized arbitrarily (to 0 for example). We consider the following surrogate cost for all $t \in [T]$:

$$\widetilde{c}_t(p) := \sum_{i=1}^{N} \frac{1}{p_i} \left\| h_i^t \right\|_2^2 \tag{8}$$

As successive iterates of the algorithm become closer and closer to each other, the squared norm of the h_i^ts become better and better approximations to the squared norm of the g_i^ts, thereby making the surrogate costs (8) accurate approximations of the true costs (5). The idea of using previously seen gradients to approximate current gradients is inspired by the celebrated variance-reduced stochastic optimization algorithm SAGA [29].

We are now almost ready to formulate our algorithm. If we try to directly minimize the surrogate cost functions over the entire simplex at each iteration as in (7), we might end up assigning null or close to null probabilities to certain indices. Depending on how far we are in the algorithm, this might or might not be a problem. In the initial stages, this is clearly an issue since this might only be an artifact of the initialization (notably when we initialize the h_i^1 to 0), or an accurate representation of the current norm of the gradient, but which might not be representative later on as the algorithm progresses. On the other hand, in the later stages of the algorithm, the cost functions are nearly identical, so that an assignment of near zero probabilities is a reflection of the true optimal probabilities, and is not necessarily problematic.

The above discussion suggests the following algorithm. Define the ε-restricted probability simplex to be:

$$\Delta(\varepsilon) := \left\{ p \in \mathbb{R}^N \mid p_i \geq \varepsilon,\ \sum_{i=1}^{N} p_i = 1 \right\} \tag{9}$$

And let $\{\varepsilon_t\}_{t=1}^T$ be a decreasing sequence of positive numbers with $\varepsilon_1 \leq \frac{1}{N}$. Then we propose the following algorithm:

$$p^t := \operatorname*{argmin}_{p \in \Delta(\varepsilon_t)} \{\widetilde{c}_t(p)\} \tag{10}$$

Our theoretical analysis in Section 4 suggests a specific decay rate for the sequence $\{\varepsilon_t\}_{t=1}^T$ that depends on the sequence of step-sizes $\{\alpha_t\}_{t=1}^T$ and whether SGD or SGLD is run.

3.2 Explicit form

Equation (10) defines our sequence of distribution $\{p^t\}_{t=1}^T$, but the question remains whether we can solve the implied optimization problems efficiently. In this section we answer this question in the affirmative, and provide an explicit algorithm for the computation of the sequence $\{p^t\}_{t=1}^T$.

We state our main result of this section in the following lemma. The proof can be found in appendix A.

Lemma 1. *Let $\{a_i\}_{i=1}^N$ be a non-negative set of numbers where at least one of the a_is is strictly positive, and let $\varepsilon \in [0, 1/N]$. Let $\pi : [N] \to [N]$ be a permutation that orders $\{a_i\}_{i=1}^N$ in a decreasing order ($a_{\pi(1)} \geq a_{\pi(2)} \geq \cdots \geq a_{\pi(N)}$). Define:*

$$\rho := \max \left\{ i \in [N] \;\middle|\; a_{\pi(i)} \geq \varepsilon \frac{\sum_{j=1}^{i} a_{\pi(j)}}{1 - (N-i)\varepsilon} \right\} \tag{11}$$

and:

$$\lambda := \frac{\sum_{j=1}^{\rho} a_{\pi(j)}}{1 - (N-\rho)\varepsilon} \tag{12}$$

Then a solution of the optimization problem:

$$\min_{p \in \Delta(\varepsilon)} \sum_{i=1}^{N} \frac{1}{p_i} a_i^2 \tag{13}$$

is given by:

$$p_i^* = \begin{cases} a_i/\lambda & \text{if } i \in \{\pi(1), \ldots, \pi(\rho)\} \\ \varepsilon & \text{otherwise} \end{cases} \tag{14}$$

In the case all a_i are zero, any $p \in \Delta(\varepsilon)$ is a solution.

In light of Lemma 1, to compute p^t as defined in (10), it is enough to know the value of ρ_t as defined in (11), replacing a_i with $\|h_i^t\|_2$ and ε with ε_t. Using ρ_t we can then compute λ_t using (12), and obtain p^t from (14). It remains to specify an efficient way to perform this computation.

3.3 Implementation details

The naive way to perform the above computation is to do the following at each iteration:

- Sort $\{\|h_i^t\|_2\}_{i=1}^N$ in decreasing order.
- Find ρ_t by traversing $(\pi(i))_{i=1}^N$ in increasing order and finding the first $i \in [N]$ for which the inequality in (11) does not hold.
- Explicitly compute the probabilities using (12) and (14).
- Sample from p^t using inverse transform sampling.

This has complexity $O(N \log N)$. In appendix A, we present an algorithm that requires only $O(N)$ vectorized operations, $O(\log^2 N)$ sequential (non-vectorized) operations, and cN memory for small c. The algorithm uses three data structures:

- An array storing $\{\|h_i^t\|_2\}_{i=1}^N$ unsorted.
- An order statistic tree storing the pairs $(\|h_i^t\|_2, i)_{i=1}^N$ sorted according to $\|h_i^t\|_2$.
- An array storing $\{\sum_{j=1}^i \left\|h_{\pi(j)}^t\right\|_2\}_{i=1}^N$ where π is the permutation that sorts $\{\|h_i^t\|_2\}_{i=1}^N$ in the order statistic tree.

The order statistic tree along with the array storing the cumulative sums allows the retrieval of ρ_t in $O(\log^2 N)$ time. The cumulative sums allow to sample from p^t in $O(\log N)$ time using binary search, and maintaining them is the only operation that requires a vectorized $O(N)$ operation. All other operations run in $O(\log N)$ time. See appendix A for the full version of the algorithm and a more complete discussion of its computational complexity.

4 Theory

In this section, we prove a sub-linear dynamic regret guarantee for our proposed algorithm when used with SGD and SGLD with decreasing step-sizes. We present our results in a more general setting, and show that they apply to our cases of interest. We start by stating our assumptions:

Assumption 1. *(Bounded gradients) There exists a $G > 0$ such that $\|\nabla f_i(x)\|_2 \leq G$ for all $x \in \mathbb{R}^d$ and for all $i \in [N]$.*

Assumption 2. *(Smoothness) There exists an $L > 0$ such that $\|\nabla f_i(x) - \nabla f_i(y)\|_2 \leq L \|x - y\|_2$ for all $x, y \in \mathbb{R}^d$ and for all $i \in [N]$.*

Assumption 3. *(Contraction of the iterates in expectation) There exists constants $A \geq 0$, $B \geq 1$ and $\delta \in (0, 1]$ such that $\mathbb{E}\left[\|x_{t+1} - x_t\|_2 \mid I_1, \ldots, I_{t-1}\right] \leq A/(B + t - 1)^\delta$ for all $t \in [T]$.*

The bounded gradients assumption has been used in all previous related work [10, 11, 12, 13], although it is admittedly quite strong. The smoothness assumption is standard in the study of optimization and sampling algorithms. Note that we chose to state our assumptions using index-independent constants to make the presentation clearer and since this does not affect our derivation of the sequence $\{\varepsilon_t\}_{t=1}^T$.

Finally, Assumption 3 should really be derived from more basic assumptions, and is only stated to allow for a unified analysis. Note that in the optimization case this is a very mild assumption since we are generally only interested in convergent sequences, and for any such sequence with reasonably fast convergence this assumption holds. The following proposition shows that Assumption 3 holds for our cases of interest. All the proofs for this section can be found in appendix B.

Proposition 1. *For any choice of $\{p^t\}_{t=1}^T$, the iterates of SGD (2) with the gradient estimator (4) and decreasing step-sizes $\alpha_t := E/(F + t - 1)^\beta$ with $E \geq 0$, $F \geq 1$ and $\beta \in (0, 1]$ satisfy Assumption 3*

with $A := NGE$, $B := F$, and $\delta := \beta$. Under the same conditions, the iterates of SGLD (3) satisfy Assumption 3 with $A := \sqrt{E}\left(NG\sqrt{\alpha_1} + \sqrt{2d}\right)$, $B := F$, and $\delta := \beta/2$.

We now state a proposition that relates the optimal function value for the problem (13) over the restricted simplex, with the optimal function value over the whole simplex. Its proof is taken from ([12], Lemma 6):

Proposition 2. *Let $\{a_i\}_{i=1}^N$ be a non-negative set of numbers, and let $\varepsilon \in [0, 1/2N]$. Then:*

$$\min_{p\in\Delta(\varepsilon)} \sum_{i=1}^N \frac{1}{p_i} a_i^2 - \min_{p\in\Delta} \sum_{i=1}^N \frac{1}{p_i} a_i^2 \leq 6\varepsilon N \left(\sum_{i=1}^N a_i\right)^2$$

The following lemma gives our first bound on the regret per time step:

Lemma 2. *Let $q^t := \operatorname{argmin}_{p\in\Delta}\{c_t(p)\}$. Under Assumption 1, and when using the sequence of distributions defined by (10), we have the following bound for $t \in \{t_0, \dots, T\}$:*

$$\mathbb{E}\left[c_t(p^t) - c_t(q^t)\right] \leq \frac{4G}{\varepsilon_t}\,\mathbb{E}\left[\sum_{i=1}^N \left\|g_i^t - h_i^t\right\|_2\right] + 6\varepsilon_t G^2 N^3$$

where $t_0 := \min\{t \in [T] \mid \varepsilon_t \leq \frac{1}{2N}\}$.

Proof outline. Let $\widetilde{p}^t := \operatorname{argmin}_{p\in\Delta}\{\widetilde{c}_t(p)\}$. Then we have the following decomposition:

$$\mathbb{E}\left[c_t(p^t) - c_t(q^t)\right] = \underbrace{\mathbb{E}\left[c_t(p^t) - \widetilde{c}_t(p^t)\right]}_{\text{(A)}} + \underbrace{\mathbb{E}\left[\widetilde{c}_t(p^t) - \widetilde{c}_t(\widetilde{p}^t)\right]}_{\text{(B)}} + \underbrace{\mathbb{E}\left[\widetilde{c}_t(\widetilde{p}^t) - c_t(q^t)\right]}_{\text{(C)}}$$

The terms (A) and (C) are the penalties we pay for using a surrogate cost function, while (B) is the price we pay for restricting the simplex. Using Assumption 1, proposition 2, and the fact that p^t is contained in the ε_t-restricted simplex, each of these terms can be bound to give the result stated. □

The expectation in the first term of the above lemma is highly reminiscent of the first term of the Lyapunov function used to study the convergence of SAGA first proposed in [30] and subsequently refined in [31] (with g_i^t replaced by g_i^*, the gradient at the minimum). Inspired by this similarity, we prove the following recursion:

Lemma 3. *Under Assumptions 2 and 3, we have:*

$$\mathbb{E}\left[\sum_{i=1}^N \left\|g_i^{t+1} - h_i^{t+1}\right\|_2\right] \leq \frac{NLA}{(B+t-1)^\delta} + (1-\varepsilon_t)\,\mathbb{E}\left[\sum_{i=1}^N \left\|g_i^t - h_i^t\right\|_2\right]$$

The natural thing to do at this stage is to unroll the above recursion, replace in Lemma 2, sum over all time steps, and minimize the obtained regret bound over the choice of the sequence $\{\varepsilon_t\}_{t=1}^T$. However, even if we can solve the resulting optimization problem efficiently, the solution will still depend on the constants G and L and on the initial error due to the initialization of the h_is, both of which are usually unknown. Here instead, we make an asymptotic argument to find the optimal decay rate of $\{\varepsilon_t\}_{t=1}^T$, propose a sequence that satisfies this decay rate, and show that it gives rise to the dynamic regret bounds stated.

Denote by $\varphi(t)$ the expectation in the first term of the upper bound in Lemma 2. Suppose we take ε_t to be of order $t^{-\beta}$. Looking at the recursion in Lemma 3, we see that to control the positive term, we need the negative term to be of order at least $t^{-\delta}$, so that $\varphi(t)$ cannot be smaller than $t^{\beta-\delta}$. The bound of Lemma 2 is therefore of order $t^{2\beta-\delta} + t^{-\beta}$. The minimum is attained when the exponents are equal so we have: $2\beta - \delta = -\beta \implies \beta = \frac{\delta}{3}$

We are now ready to guess the form of ε_t. Matching the positive term in Lemma 3, we consider the following sequence:

$$\varepsilon_t := \frac{1}{C^{1-\delta/3}(C+t-1)^{\delta/3}} \tag{15}$$

For a free parameter C satisfying $C \geq N$ to ensure $\varepsilon_1 \leq 1/N$. With this choice of the sequence $\{\varepsilon_t\}_{t=1}^T$, we are now finally ready to state our main result of the paper:

Theorem 1. *Under Assumptions 1 and 2 on the functions f_i and Assumption 3 on the sequence $\{x_t\}_{t=1}^T$, algorithm (10) with the sequence $\{\varepsilon_t\}_{t=1}^T$ given in (15) satisfies the following dynamic regret bound for all $T \geq t_0$:*

$$\mathbb{E}\left[\textit{Regret}_D(T)\right] \leq \mathcal{O}(T^{1-\delta/3}) \tag{16}$$

where $t_0 := \min\{t \in [T] \mid \varepsilon_t \leq \frac{1}{2N}\}$ as in Lemma 2.

Proof outline. Using the sequence given by (15) and the recursion in Lemma 3, we show by induction that $\varphi(t) \leq \mathcal{O}(t^{-2\delta/3})$. Replacing in Lemma 2, summing over all time steps, and bounding the resulting sums by the appropriate integrals we get the result. □

Note that since $C \geq N$, we have $t_0 \leq (2^{3/\delta} - 1)N + 1$, so t_0 is bounded by a constant independent of T. Furthermore, setting $C = 2N$ makes the theorem hold for all $T \in \mathbb{N}$. In practice however, it might be beneficial to set $C = N$ to overcome a bad initialization of the h_is.

Combining Theorem 1 with proposition 1 we obtain the following corollary:

Corollary 1. *Under Assumptions 1 and 2 on the functions f_i, if SGD (2) is run with step-sizes $\mathcal{O}(1/t)$ using the estimator (4) and probabilities (10) with the sequence $\{\varepsilon_t\}_{t=1}^T$ given by (15), then for all $T \geq t_0$:*

$$\mathbb{E}\left[\textit{Regret}_D(T)\right] \leq \mathcal{O}(T^{2/3}) \tag{17}$$

and for SGLD (3):

$$\mathbb{E}\left[\textit{Regret}_D(T)\right] \leq \mathcal{O}(T^{5/6}) \tag{18}$$

under the same conditions.

5 A new mini-batch estimator

In most practical applications, one uses a mini-batch of samples to construct the gradient estimator instead of just a single sample. The most basic such estimator is the one formed by sampling a mini-batch of indices $S_t = \{I_t^1, \dots, I_t^m\}$ uniformly and independently, and taking the sample mean. This gives an unbiased estimator, whose variance decreases as $1/m$. A simple way to make it more efficient is by sampling the indices uniformly but without replacement. In that case, the variance decreases by an additional factor of $(1-(m-1)/(N-1))$. For $m \ll N$, the difference is negligible, and the additional cost of sampling without replacement is not justified. However, when using unequal probabilities, this argument no longer holds, and the additional variance reduction obtained from sampling without replacement can be significant even for small m.

For our problem, besides the additional variance reduction, sampling without replacement allows a higher rate of replacement of the h_is, which is directly related to our regret through the factor in front of the second term of Lemma 3, whose proper generalization for mini-batch sampling is $(1-\min_{i\in[N]} \pi_i^t)$, where $\pi_i^t = P(i \in S_t)$ is the inclusion probability of index i. This makes sampling without replacement desirable for our purposes. Unfortunately, unlike in the uniform case, the sample mean generalization of (4) is no longer unbiased. We propose instead the following estimator:

$$\hat{g}_b^t = \frac{1}{m}\sum_{j=1}^m \hat{g}_j^t \qquad \hat{g}_j^t := \left[\frac{1}{q_{I_t^j}^{t,j}} g_{I_t^j}^t + \sum_{k=1}^{j-1} g_{I_t^k}^t\right] \qquad q_i^{t,j} := \frac{p_i^t}{1-\sum_{k=1}^{j-1} p_{I_t^k}^t} \tag{19}$$

We summarize some of its properties in the following proposition:

Proposition 3. *Let $S_t^j := \{I_t^1, \dots, I_t^j\}$ for $j \in [m]$ and $S_t^0 := \emptyset$. We have:*

(a) $\mathbb{E}\left[\hat{g}_b^t\right] = g^t$

(b) $\mathbb{E}\left[\left\|\hat{g}_b^t - g^t\right\|_2^2\right] = (1/m^2)\sum_{j=1}^m \mathbb{E}\left[\left\|\hat{g}_j^t - g^t\right\|_2^2\right]$

(c) $\operatorname{argmin}_{p\in\Delta}\{\mathbb{E}\left[\left\|\hat{g}_b^t - g^t\right\|_2^2\right]\} = \operatorname{argmin}_{p\in\Delta}\{c_t(p)\}$

(d) $\mathbb{E}\left[\left\|\hat{g}_{j+1}^t - g^t\right\|_2^2\right] = \left(1 - \mathbb{E}\left[q_{I_t^j}^{t,j}\right]\right)\mathbb{E}\left[\left\|\hat{g}_j^t - g^t\right\|_2^2\right] - \mathbb{E}\left[q_{I_t^j}^{t,j}\left\|\hat{g}_j^t - g^t\right\|_2^2\right]$

where all the expectations in (d) are conditional on S_t^{j-1}.

The proposed estimator is therefore unbiased and its variance decreases super-linearly in m (by (b) and (d)). Although we were unable to prove a regret bound for this estimator, (c) suggests that it is still reasonable to use our algorithm. To take into account the higher rate of replacement of the h_is, we propose using the following ε_t sequence, which is based on the mini-batch equivalent of Lemma 3 and the inequality $\min_{i\in[N]} \pi_i^t \geq m\varepsilon_t$ [32, 33, 34]:

$$\varepsilon_t := \frac{1}{C^{1-\delta/3}(C + m(t-1))^{\delta/3}} \tag{20}$$

6 Experiments

In this section, we present results of experiments with our algorithm. We start by validating our theoretical results through a synthetic experiment. We then show that our proposed method outperforms existing methods on real world datasets. Finally, we identify settings in which adaptive importance sampling can lead to significant performance gains for the final optimization algorithm.

In all experiments, we added l_2-regularization to the model's loss and set the regularization parameter $\mu = 1$. We ran SGD (2) with decreasing step sizes $\alpha_t = \frac{m}{2NL+m\mu t}$ where m is the batch size following [35]. We experimented with 3 different samplers in addition to our proposed sampler: Uniform, Multi-armed bandit Sampler (*Mabs*) [11], and Variance Reducer Bandit (*Vrb*) [12]. The hyperparameters of both *Mabs* and *Vrb* are set to the ones prescribed by the theory in the original papers. For our propsoed sampler *Avare*, we use the epsilon sequence given by (20) with $C = N$, $\delta = 1$, and initialized $h_i = 0$ for all $i \in [N]$. For each sampler, we ran SGD 10 times and averaged the results. The shaded areas represent a one standard deviation confidence interval.

To validate our theoretical findings, we randomly generated a dataset for binary classification with $N = 100$ and $d = 10$. We then trained a logistic regression model on the generated data, and used a batch size of 1 to match the setting of our regret bound. The results of this experiment are depicted in figure 1, and show that *Avare* outperforms the other methods, achieving significantly lower dynamic regret and faster convergence.

For our second experiment, we tested our algorithm on three real world datasets: MNIST, IJCNN1 [36], and CIFAR10. We used a softmax regression model, and a batch size of 128 sampled with replacement. The results of this experiment can be seen in figure 2. The third column shows the relative error which we define as $[c_t(p^t) - \min_{p\in\Delta} c_t(p)] \,/\, \min_{p\in\Delta} c_t(p)$. In all three cases, *Avare* achieves significantly smaller dynamic regret, with a relative error quickly decaying to zero. The performance gains in the final optimization algorithm are clear in both MNIST and IJCNN1, but are not noticeable in the case of CIFAR10.

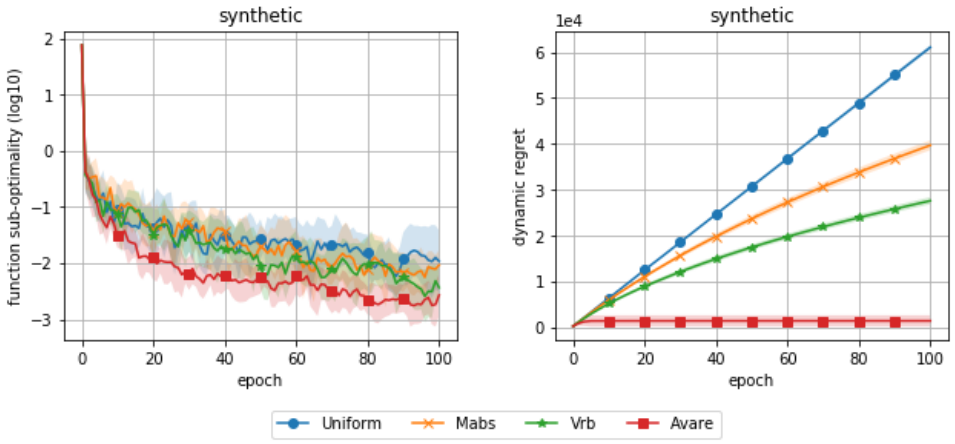

Figure 1: Evolution of function sub-optimality (left) and dynamic regret (right) as a function of data passes on a synthetic dataset with an l_2 regularized logistic regression model and different samplers.

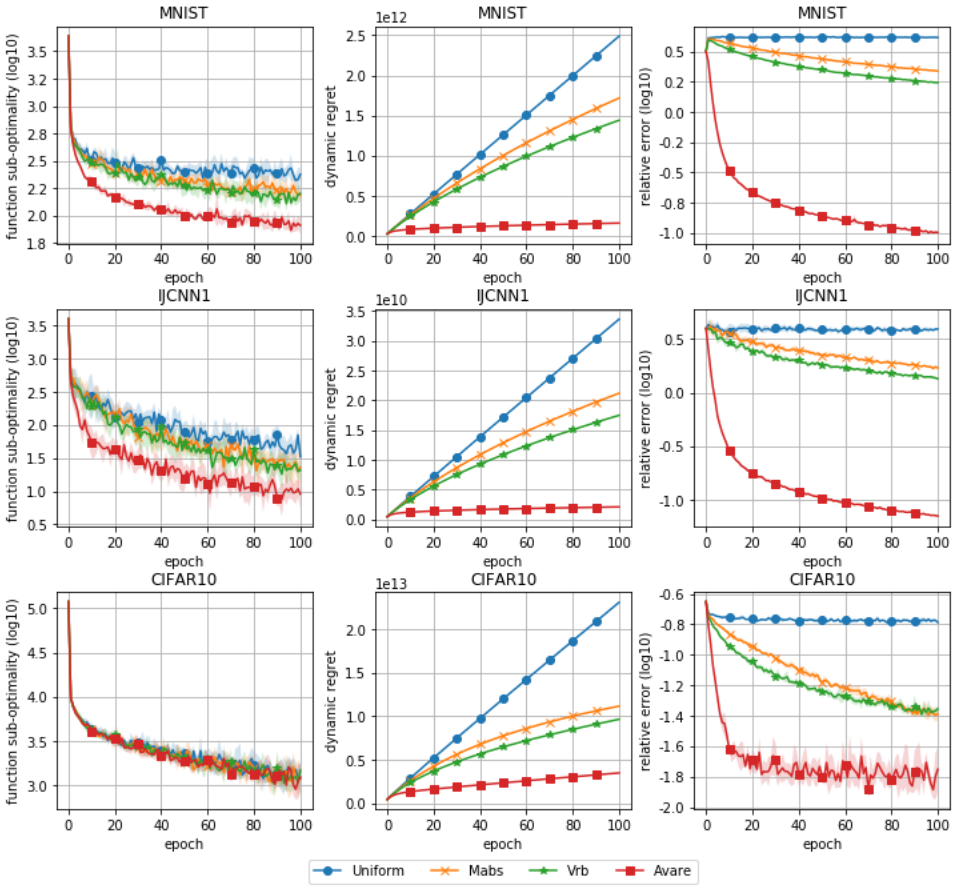

Figure 2: Comparison of the performance of importance samplers on an l2-regularized softmax regression model on three real world datasets: MNIST (top), IJCNN1 (middle), CIFAR10 (bottom).

Table 1: Useful ratios in determining the effectiveness of variance reduction through importance sampling. L_i is the smoothness constant of f_i, $L_{max} = \max_{i \in [N]} L_i$, and g_i^* and is the gradient of f_i at the loss minimizer x^*.

Dataset	$\frac{NL_{max}}{\sum_{i=1}^N L_i}$	$\frac{N \sum_{i=1}^N \|g_i^*\|_2^2}{(\sum_{i=1}^N \|g_i^*\|_2)^2}$
Synthetic	1.69	4.46
MNIST	3.28	5.08
IJCNN1	1.12	4.83
CIFAR10	3.40	1.14

To determine the effectiveness of non-uniform sampling in accelerating the optimization process, previous work [4, 11] has suggested to look at the ratio of the maximum smoothness constant and the average smoothness constant. We show here that this is the wrong measure to look at when using adaptive probabilities. In particular, we argue that the ratio of the variance with uniform sampling at the loss minimizer to the optimal variance at the loss minimizer is much more informative of the performance gains achievable through adaptive importance sampling. For each dataset, these ratios are displayed in Table 1, supporting our claim. We suspect that for large models capable of fitting the data nearly perfectly, our proposed ratio will be large since many of the per-example gradients at the optimum will be zero. We therefore expect our method to be particularly effective in the training of models of this type. We leave such experiments for future work. Finally, in appendix D, we propose an extension of our method to constant step-size SGD, and show that it preserves the performance gains observed when using decreasing step-sizes.

Broader Impact

Our work develops a new method for variance reduction for stochastic optimization and sampling algorithms. On the optimization side, we expect our method to be very useful in accelerating the training of large scale neural networks, particularly since our method is expected to provide significant performance gains when the model is able to fit the data nearly perfectly. On the sampling side, we expect our method to be useful in accelerating the convergence of MCMC algorithms, opening the door for the use of accurate Bayesian methods at a large scale.

Acknowledgments and Disclosure of Funding

This research was supported by an NSERC discovery grant. We would like to thank the anonymous reviewers for their useful comments and suggestions.

References

[1] Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *Ann. Math. Statist.*, 22(3):400–407, 1951.

[2] Max Welling and Yee Whye Teh. Bayesian Learning via Stochastic Gradient Langevin Dynamics. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 681–688. Omnipress, 2011.

[3] Deanna Needell, Rachel Ward, and Nathan Srebro. Stochastic Gradient Descent, Weighted Sampling, and the Randomized Kaczmarz algorithm. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D Lawrence, and Kilian Q Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 1017–1025, 2014.

[4] Peilin Zhao and Tong Zhang. Stochastic Optimization with Importance Sampling for Regularized Loss Minimization. In Francis R Bach and David M Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1–9. JMLR.org, 2015.

[5] Guillaume Bouchard, Théo Trouillon, Julien Perez, and Adrien Gaidon. Accelerating Stochastic Gradient Descent via Online Learning to Sample. *CoRR*, abs/1506.0, 2015.

[6] Sebastian U Stich, Anant Raj, and Martin Jaggi. Safe Adaptive Importance Sampling. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M Wallach, Rob Fergus, S V N Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 4381–4391, 2017.

[7] Angelos Katharopoulos and François Fleuret. Not All Samples Are Created Equal: Deep Learning with Importance Sampling. In Jennifer G Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2530–2539. PMLR, 2018.

[8] Tyler B Johnson and Carlos Guestrin. Training Deep Models Faster with Robust, Approximate Importance Sampling. In Samy Bengio, Hanna M Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 7276–7286, 2018.

[9] Dominik Csiba and Peter Richtárik. Importance Sampling for Minibatches. *J. Mach. Learn. Res.*, 19:27:1—-27:21, 2018.

[10] Hongseok Namkoong, Aman Sinha, Steve Yadlowsky, and John C Duchi. Adaptive Sampling Probabilities for Non-Smooth Optimization. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2574–2583. PMLR, 2017.

[11] Farnood Salehi, L Elisa Celis, and Patrick Thiran. Stochastic Optimization with Bandit Sampling. *CoRR*, abs/1708.0, 2017.

[12] Zalan Borsos, Andreas Krause, and Kfir Y Levy. Online Variance Reduction for Stochastic Optimization. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018*, volume 75 of *Proceedings of Machine Learning Research*, pages 324–357. PMLR, 2018.

[13] Zalán Borsos, Sebastian Curi, Kfir Yehuda Levy, and Andreas Krause. Online Variance Reduction with Mixtures. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 705–714. PMLR, 2019.

[14] Omar Besbes, Yonatan Gur, and Assaf J Zeevi. Non-Stationary Stochastic Optimization. *Oper. Res.*, 63(5):1227–1244, 2015.

[15] Xi Chen, Yining Wang, and Yu-Xiang Wang. Non-stationary Stochastic Optimization with Local Spatial and Temporal Changes. *CoRR*, abs/1708.0, 2017.

[16] Aryan Mokhtari, Shahin Shahrampour, Ali Jadbabaie, and Alejandro Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *55th IEEE Conference on Decision and Control, CDC 2016, Las Vegas, NV, USA, December 12-14, 2016*, pages 7195–7201. IEEE, 2016.

[17] Tianbao Yang, Lijun Zhang, Rong Jin, and Jinfeng Yi. Tracking Slowly Moving Clairvoyant: Optimal Dynamic Regret of Online Learning with True and Noisy Gradient. In Maria-Florina Balcan and Kilian Q Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 449–457. JMLR.org, 2016.

[18] Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. Adaptive Online Learning in Dynamic Environments. In Samy Bengio, Hanna M Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 1330–1340, 2018.

[19] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 2003.

[20] A. S. Nemirovsky and D. B. Yudin. Problem Complexity and Method Efficiency in Optimization. *The Journal of the Operational Research Society*, 1984.

[21] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The Nonstochastic Multiarmed Bandit Problem. *SIAM J. Comput.*, 32(1):48–77, 2002.

[22] Martin Zinkevich. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In Tom Fawcett and Nina Mishra, editors, *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 928–936. AAAI Press, 2003.

[23] Shai Shalev-Shwartz. Online Learning and Online Convex Optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[24] Elad Hazan. Introduction to Online Convex Optimization. *Found. Trends Optim.*, 2(3-4):157–325, 2016.

[25] Mark Herbster and Manfred Warmuth. Tracking the Best Expert. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 286–294. Morgan Kaufmann, San Francisco (CA), 1995.

[26] Mark Herbster and Manfred K Warmuth. Tracking the Best Linear Predictor. *J. Mach. Learn. Res.*, 1:281–309, sep 2001.

[27] Nicolò Cesa-Bianchi, Pierre Gaillard, Gábor Lugosi, and Gilles Stoltz. Mirror Descent Meets Fixed Share (and feels no regret). In Peter L Bartlett, Fernando C N Pereira, Christopher J C Burges, Léon Bottou, and Kilian Q Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 989–997, 2012.

[28] András György and Csaba Szepesvári. Shifting Regret, Mirror Descent, and Matrices. In Maria-Florina Balcan and Kilian Q Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2943–2951. JMLR.org, 2016.

[29] Aaron Defazio, Francis R. Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *CoRR*, abs/1407.0202, 2014.

[30] Thomas Hofmann, Aurélien Lucchi, Simon Lacoste-Julien, and Brian McWilliams. Variance Reduced Stochastic Gradient Descent with Neighbors. In Corinna Cortes, Neil D Lawrence, Daniel D Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2305–2313, 2015.

[31] Aaron Defazio. A Simple Practical Accelerated Method for Finite Sums. In Daniel D Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 676–684, 2016.

[32] Yaming Yu. On the inclusion probabilities in some unequal probability sampling plans without replacement. *Bernoulli*, 18(1):279–289, 2012.

[33] Hartmut Milbrodt. Comparing inclusion probabilities and drawing probabilities for rejective sampling and successive sampling. *Statistics Probability Letters*, 14(3):243–246, 1992.

[34] Subhash Kochar and Ramesh Korwar. On Random Sampling Without Replacement from a Finite Population. *Annals of the Institute of Statistical Mathematics*, 53:631–646, 2001.

[35] Sebastian U. Stich. Unified optimal analysis of the (stochastic) gradient method, 2019.

[36] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3), May 2011.

Appendix A Algorithm

A.1 Proof of Lemma 1

Lemma 1. *Let $\{a_i\}_{i=1}^N$ be a non-negative set of numbers where at least one of the a_is is strictly positive, and let $\varepsilon \in [0, 1/N]$. Let $\pi : [N] \to [N]$ be a permutation that orders $\{a_i\}_{i=1}^N$ in a decreasing order ($a_{\pi(1)} \geq a_{\pi(2)} \geq \cdots \geq a_{\pi(N)}$). Define:*

$$\rho := \max \left\{ i \in [N] \;\middle|\; a_{\pi(i)} \geq \varepsilon \frac{\sum_{j=1}^{i} a_{\pi(j)}}{1 - (N-i)\varepsilon} \right\} \tag{11}$$

and:

$$\lambda := \frac{\sum_{j=1}^{\rho} a_{\pi(j)}}{1 - (N-\rho)\varepsilon} \tag{12}$$

Then a solution of the optimization problem:

$$\min_{p \in \Delta(\varepsilon)} \sum_{i=1}^{N} \frac{1}{p_i} a_i^2 \tag{13}$$

is given by:

$$p_i^* = \begin{cases} a_i/\lambda & \text{if } i \in \{\pi(1), \ldots, \pi(\rho)\} \\ \varepsilon & \text{otherwise} \end{cases} \tag{14}$$

In the case all a_i are zero, any $p \in \Delta(\varepsilon)$ is a solution.

Proof.

Edge case and well-definedness. If $a_i = 0$ for all $i \in [N]$, then the objective function is identically 0 over $\Delta(\varepsilon)$ for all $\varepsilon \in [0, 1/N]$, so that any $p \in \Delta(\varepsilon)$ is a solution (we set $(1/0)0 := 0$ in the objective). Else there exists an $i \in [N]$ such that $a_i > 0$, and therefore $a_{\pi(1)} > 0$. Now:

$$\begin{aligned} &\frac{\varepsilon}{1 - (N-1)\varepsilon} \leq 1 \\ &\Leftrightarrow \varepsilon \leq 1 - (N-1)\varepsilon \\ &\Leftrightarrow 0 \leq 1 - N\varepsilon \\ &\Leftrightarrow N\varepsilon \leq 1 \\ &\Leftrightarrow \varepsilon \leq \frac{1}{N} \end{aligned}$$

The last inequality is true, so it implies the first, and we have:

$$a_{\pi(1)} \geq \varepsilon \frac{a_{\pi(1)}}{1 - (N-1)\varepsilon}$$

Therefore ρ is well defined and is ≥ 1. As a consequence, λ is also well defined and is > 0, making p^* in turn well-defined.

Optimality proof. It is easily verified that problem (13) is convex. Its Lagrangian is given by:

$$\mathcal{L}(p, \mu, \nu) = \sum_{i=1}^{N} \frac{1}{p_i} a_i^2 - \sum_{i=1}^{N} \mu_i (p_i - \varepsilon) + \nu \left(\sum_{i=1}^{N} p_i - 1 \right)$$

and the KKT conditions are:

$$\begin{aligned} \text{(Stationarity)} \quad & p_i = \frac{a_i}{\sqrt{\nu - \mu_i}} \\ \text{(Complementary slackness)} \quad & \mu_i = 0 \vee p_i = \varepsilon \\ \text{(Primal feasibility)} \quad & p_i \geq \varepsilon \wedge \sum_{j=1}^{N} p_j = 1 \\ \text{(Dual feasibility)} \quad & \mu_i \geq 0 \end{aligned}$$

By convexity of the problem, the KKT conditions are sufficient conditions for global optimality. To show that our proposed solution is optimal, it therefore suffices to exhibit constants $(\mu_i^*)_{i=1}^N$ and ν^* that together with p^* satisfy these conditions. Let:

$$\nu^* := \lambda^2$$
$$\mu_i^* := \begin{cases} 0 & \text{if } i \in \{\pi(1), \dots, \pi(\rho)\} \\ \nu^* - a_i^2/\varepsilon^2 & \text{otherwise} \end{cases}$$

Note that the μ_i^*s are well defined since when $\varepsilon = 0$, $\rho = N$, and $\mu_i^* = 0$ for all $i \in [N]$. We claim that the triplet $(p^*, (\mu_i^*)_{i=1}^N, \nu^*)$ satisfies the KKT conditions.

Stationarity and complementary slackness are immediate from the definitions. The first clause of primal feasibility holds by definition of p_i^* for $i \in \{\pi(\rho+1), \dots, \pi(N)\}$. In the other case we have:

$$\begin{aligned} p_i^* &= \frac{a_i}{\lambda} \\ &= \frac{a_i}{\sum_{j=1}^{\rho} a_{\pi(j)}} (1 - (N-\rho)\varepsilon) \\ &\geq \frac{a_{\pi(\rho)}}{\sum_{j=1}^{\rho} a_{\pi(j)}} (1 - (N-\rho)\varepsilon) \\ &\geq \varepsilon \end{aligned}$$

where in the third line we used the fact that π orders $\{a_i\}_{i=1}^N$ in decreasing order, and in the last line we used the inequality in the definition of ρ. For the second clause of primal feasibility:

$$\sum_{j=1}^{N} p_j^* = \sum_{j=1}^{N} p_{\pi(j)}^* = \sum_{j=1}^{\rho} \frac{a_{\pi(i)}}{\lambda} + \sum_{j=\rho+1}^{N} \varepsilon = (1 - (N-\rho)\varepsilon) + (N-\rho)\varepsilon = 1$$

Finally, dual feasibility holds by definition of μ_i^* for $i \in \{\pi(1), \dots, \pi(\rho)\}$. In the other case we have:

$$\begin{aligned} \mu_i^* &= \nu^* - \frac{a_i^2}{\varepsilon^2} \\ &= \lambda^2 - \frac{a_i^2}{\varepsilon^2} \\ &= \left(\lambda + \frac{a_i}{\varepsilon}\right)\left(\lambda - \frac{a_i}{\varepsilon}\right) \\ &\geq \left(\lambda + \frac{a_i}{\varepsilon}\right)\left(\lambda - \frac{a_{\pi(\rho+1)}}{\varepsilon}\right) \end{aligned}$$

The first factor is positive by positivity of λ and non-negativity of the a_is. For the second factor, we have by the maximality of ρ:

$$\begin{aligned} & a_{\pi(p+1)} < \varepsilon \frac{\sum_{j=1}^{\rho+1} a_{\pi(j)}}{1 - (N-\rho-1)\varepsilon} \\ & \Rightarrow a_{\pi(\rho+1)}(1 - (N-\rho)\varepsilon + \varepsilon) < \varepsilon \sum_{i=1}^{\rho} a_{\pi(j)} + \varepsilon a_{\pi(\rho+1)} \\ & \Rightarrow a_{\pi(\rho+1)}(1 - (N-\rho)\varepsilon) < \varepsilon \sum_{i=1}^{\rho} a_{\pi(j)} \\ & \Rightarrow \frac{a_{\pi(\rho+1)}}{\varepsilon} < \lambda \end{aligned} \tag{21}$$

Therefore the second factor is also positive, and dual feasibility holds. □

A.2 Implementation and complexity

In this section of the appendix, we provide pseudocode for the implementation of the algorithm and discuss its computational complexity.

Algorithm 1 Implementation of the proposed sampler

class SAMPLER:

  procedure INITIALIZE($\{\|h_i\|_2\}_{i=1}^N$)

    $self.H \leftarrow Array(\{\|h_i\|_2\}_{i=1}^N)$

    $self.T \leftarrow OST(keys = \{\|h_i\|_2\}_{i=1}^N, values = [N])$

    $self.CS \leftarrow Array(\{\sum_{j=1}^i \left\|h_{\pi(j)}\right\|_2\}_{i=1}^N)$

  procedure DELETE(x)

    $r \leftarrow self.T.rank(x)$

    $self.T.delete(x)$

    $self.CS[r:N] \leftarrow self.CS[r:N] - x$

  procedure INSERT(x, i)

    $self.T.insert(key = x, value = i)$

    $r \leftarrow self.T.rank(x)$

    $self.CS[r:N] \leftarrow self.CS[r:N] + x$

  procedure UPDATE($\|h_I\|_2$, I)

    $self.delete(self.H[I])$

    $self.H[I] \leftarrow \|h_I\|_2$

    $self.insert(\|h_I\|_2, I)$

  function SEARCH(ε, node)

    $r \leftarrow self.T.rank(node)$

    if $r == N$ **then return** r

    $c \leftarrow 1 - (N - r)\varepsilon$

    if $c \cdot node.key < \varepsilon \cdot self.CS[r]$ **then**

      return $self.search(\varepsilon, node.left)$

    else

      $d = 1 - (N - r - 1)\varepsilon$

      if $d \cdot node.successor.key < \varepsilon \cdot self.CS[r+1]$ **then**

        return r

      else

        return $self.search(\varepsilon, node.right)$

  function SAMPLE(ε)

    $\rho \leftarrow self.search(\varepsilon, self.T.root)$

    $\lambda \leftarrow self.CS[\rho] / (1 - (N - \rho)\varepsilon)$

    $b \sim Bernoulli((N - \rho)\varepsilon)$

    if b == 1 **then**

      Sample an index $\tilde{I}$ uniformly from $\{\rho + 1, \ldots, N\}$.

    else

      $u \sim Uniform([0, 1])$

      Find the first index $\tilde{I}$ for which $\lambda u \leq self.CS[\tilde{I}]$ using binary search.

    $I \leftarrow self.T.select(\tilde{I})$

    return I

Algorithm 2 AVARE

Input: $x_1, T, \{\alpha_t\}_{t=1}^T, \{\left\|h_i^1\right\|_2\}_{i=1}^N, C \geq N$

$sampler \leftarrow \textsc{Sampler}(\{\left\|h_i^1\right\|_2\}_{i=1}^N)$

for $t = 1, 2, \ldots, T$ **do**

  Set ε_t according to (15)

  $I_t \leftarrow sampler.sample(\varepsilon_t)$

  Obtain x_{t+1} using (2) or (3) and the estimator (4).

  $sampler.update(\|g_{I_t}\|_2, I_t)$

return x_{T+1}

A.2.1 Implementation

We assume in algorithm 1 that an order statistic tree (OST) (see, for example, [38], chapter 14) can be instantiated and that the ordering in the tree is such that the key of the left child of a node is greater than or equal to that of the node itself. Furthermore, we assume that the $rank(x)$ method returns the position of x in the order determined by an inorder traversal of the tree. Finally, the $select(i)$ method returns the value of the i^{th}-largest key in the tree.

The algorithm works as follows. At initialization, three data structures are initialized: an arrray H holding the gradient norms according to the original indices, an order statistic tree T holding the gradient norms as keys and the original indices as values, and an array CS holding the cumulative sums of the gradient norms, where the sums are accumulated in the (decreasing) order that sorts the gradients norms in T.

The $sample(\varepsilon)$ method allows to sample from the optimal distribution on $\Delta(\varepsilon)$. It uses the $search(\varepsilon, node)$ method to find ρ by searching the tree T and using the maximality property of ρ. Once ρ is determined, λ can be calculated. Using the fact that the cumulative sums are proportional to the CDF of the distribution, the algorithm then samples an index using inverse-transform sampling. The sampled index is then transformed back to an index in the original order using the $select$ method of the tree T.

Finally, the $update(\|h_I\|_2, I)$ method replaces the gradient norm of a given index by a new one. It calls the methods $delete(x)$ and $insert(x, i)$ which perform the deletion and insertion while maintaining the tree T and array CS.

A.2.2 Complexity

First, let us analyze the cost of running the $update(\|h_I\|_2, I)$ method. For the array H, we only use random access and assignment, which are both $O(1)$. For the tree T, we use the methods $insert$, $delete$, and $rank$, all of which are $O(\log N)$. Finally, for the array CS we add and subtract from a sub-array, which takes $O(N)$ time, although this operation is vectorized and very fast in practice.

Let us now look at the cost of running $sample(\varepsilon)$. The $search$ method is recursive, but will only be called at most as many times as the height of the tree, which is $O(\log N)$. Now for each call of $search$, both the $rank$ and $successor$ methods of the tree T require $O(\log N)$ time. The rest of the $search$ method only requires $O(1)$ operations. The total cost of the $search$ method is therefore $O(\log^2 N)$. For the rest of the $sample$ method, the operations that dominate the cost are the $select$ method of the tree T, which takes $O(\log N)$ time, and the binary search in the else branch, which also runs in $O(\log N)$ time. Consequently, the total cost of the sample method is $O(\log^2 N)$.

The total per iteration cost of using the proposed sampler is therefore $O(N)$ vectorized operations, and $O(\log^2 N)$ sequential (non-vectorized) operations. The total memory cost is $O(N)$.

Appendix B Theory

We restate our assumptions here for ease of reference.

Assumption 1. *(Bounded gradients) There exists a $G > 0$ such that $\|\nabla f_i(x)\|_2 \leq G$ for all $x \in \mathbb{R}^d$ and for all $i \in [N]$.*

Assumption 2. *(Smoothness) There exists an $L > 0$ such that $\|\nabla f_i(x) - \nabla f_i(y)\|_2 \leq L \|x - y\|_2$ for all $x, y \in \mathbb{R}^d$ and for all $i \in [N]$.*

Assumption 3. *(Contraction of the iterates in expectation) There exists constants $A \geq 0$, $B \geq 1$ and $\delta \in (0, 1]$ such that $\mathbb{E}\left[\|x_{t+1} - x_t\|_2 \mid I_1, \ldots, I_{t-1}\right] \leq A/(B + t - 1)^\delta$ for all $t \in [T]$.*

B.1 Proof of proposition 1

Proposition 1. *For any choice of $\{p^t\}_{t=1}^T$, the iterates of SGD (2) with the gradient estimator (4) and decreasing step-sizes $\alpha_t := E/(F + t - 1)^\beta$ with $E \geq 0$, $F \geq 1$ and $\beta \in (0, 1]$ satisfy Assumption 3 with $A := NGE$, $B := F$, and $\delta := \beta$. Under the same conditions, the iterates of SGLD (3) satisfy Assumption 3 with $A := \sqrt{E}\left(NG\sqrt{\alpha_1} + \sqrt{2d}\right)$, $B := F$, and $\delta := \beta/2$.*

Proof. Conditioning on the knowledge of $\{I_1, \ldots, I_{t-1}\}$ we have for SGD:

$$\begin{aligned}
\mathbb{E}\left[\left\|x_{t+1}^{SGD} - x_t^{SGD}\right\|_2\right] &= \mathbb{E}\left[\alpha_t \frac{1}{p_{I_t}^t}\left\|g_{I_t}^t\right\|_2\right] \\
&= \alpha_t \sum_{i=1}^N \left\|g_i^t\right\|_2 \\
&\leq \alpha_t NG
\end{aligned}$$

and for SGLD we have:

$$\begin{aligned}
\mathbb{E}\left[\left\|x_{t+1}^{SGLD} - x_t^{SGLD}\right\|_2\right] &\leq \mathbb{E}\left[\alpha_t \frac{1}{p_{I_t}^t}\left\|g_{I_t}^t\right\|_2\right] + \mathbb{E}\left[\|\xi_t\|_2\right] \\
&\leq \alpha_t \sum_{i=1}^N \left\|g_i^t\right\|_2 + \sqrt{\mathbb{E}\left[\|\xi_t\|_2^2\right]} \\
&\leq \alpha_t NG + \sqrt{\alpha_t}\sqrt{2d} \\
&\leq \sqrt{\alpha_t}\left(NG\sqrt{\alpha_1} + \sqrt{2d}\right)
\end{aligned}$$

where in the first line we used the triangle inequality, in the second we used Jensen's inequality, and in the last we used the fact that $\{\alpha_t\}_{t=1}^T$ is decreasing. Replacing with the value of α_t we obtain the result. □

B.2 Proof of proposition 2

The following proof is taken from ([12], Lemma 6).

Proposition 2. *Let $\{a_i\}_{i=1}^N$ be a non-negative set of numbers, and let $\varepsilon \in [0, 1/2N]$. Then:*

$$\min_{p \in \Delta(\varepsilon)} \sum_{i=1}^N \frac{1}{p_i} a_i^2 - \min_{p \in \Delta} \sum_{i=1}^N \frac{1}{p_i} a_i^2 \leq 6\varepsilon N \left(\sum_{i=1}^N a_i\right)^2$$

Proof. By Lemma 1 we have:

$$\begin{aligned}
\min_{p \in \Delta(\varepsilon)} \sum_{i=1}^N \frac{1}{p_i} a_i^2 &= \lambda \sum_{i=1}^\rho a_{\pi(i)} + \sum_{i=\rho+1}^N \frac{a_{\pi(i)}^2}{\varepsilon} \\
&\leq \lambda^2 \left(1 - (N - \rho)\varepsilon\right) + \varepsilon \sum_{i=\rho+1}^N \frac{a_{\pi(\rho+1)}^2}{\varepsilon^2} \\
&\leq \lambda^2 \left(1 - (N - \rho)\varepsilon\right) + (N - \rho)\varepsilon\lambda^2 \\
&= \lambda^2
\end{aligned}$$

where in the third line we used inequality (21) from the proof of Lemma 1. Now for the case $\varepsilon = 0$ we have $\rho = N$, so the second term in the first line is zero and the inequality becomes an equality:

$$\min_{p\in\Delta} \sum_{i=1}^{N} \frac{1}{p_i} a_i^2 = \left(\sum_{i=1}^{N} a_i \right)^2 \tag{22}$$

The difference is therefore bounded by:

$$\begin{aligned}
\min_{p\in\Delta(\varepsilon)} \sum_{i=1}^{N} \frac{1}{p_i} a_i^2 - \min_{p\in\Delta} \sum_{i=1}^{N} \frac{1}{p_i} a_i^2 &\leq \frac{\left(\sum_{i=1}^{\rho} a_{\pi(i)}\right)^2}{(1-(N-\rho)\varepsilon)^2} - \left(\sum_{i=1}^{N} a_i \right)^2 \\
&\leq \left(\frac{1}{(1-N\varepsilon)^2} - 1 \right) \left(\sum_{i=1}^{N} a_i \right)^2 \\
&\leq 6\varepsilon N \left(\sum_{i=1}^{N} a_i \right)^2
\end{aligned}$$

where in the last line we used the inequality $\frac{1}{(1-x)^2} - 1 \leq 6x$ for $x \in [0, 1/2]$ which gives the restriction $\varepsilon \in [0, 1/2N]$. □

B.3 Proof of Lemma 2

Lemma 2. *Let $q^t := \operatorname{argmin}_{p\in\Delta}\{c_t(p)\}$. Under Assumption 1, and when using the sequence of distributions defined by (10), we have the following bound for $t \in \{t_0, \dots, T\}$:*

$$\mathbb{E}\left[c_t(p^t) - c_t(q^t)\right] \leq \frac{4G}{\varepsilon_t} \mathbb{E}\left[\sum_{i=1}^{N} \left\| g_i^t - h_i^t \right\|_2 \right] + 6\varepsilon_t G^2 N^3$$

where $t_0 := \min\{t \in [T] \mid \varepsilon_t \leq \frac{1}{2N}\}$.

Proof. Let $t \geq t_0$ and $\widetilde{p}^t := \operatorname{argmin}_{p\in\Delta}\{\widetilde{c}_t(p)\}$. We have the following decomposition:

$$\mathbb{E}\left[c_t(p^t) - c_t(q^t)\right] = \underbrace{\mathbb{E}\left[c_t(p^t) - \widetilde{c}_t(p^t)\right]}_{\text{(A)}} + \underbrace{\mathbb{E}\left[\widetilde{c}_t(p^t) - \widetilde{c}_t(\widetilde{p}^t)\right]}_{\text{(B)}} + \underbrace{\mathbb{E}\left[\widetilde{c}_t(\widetilde{p}^t) - c_t(q^t)\right]}_{\text{(C)}}$$

We bound each term separately:

$$\begin{aligned}
\text{(A)} &= \mathbb{E}\left[\sum_{i=1}^{N} \frac{1}{p_i^t} \left(\left\| g_i^t \right\|_2^2 - \left\| h_i^t \right\|_2^2 \right) \right] \\
&= \mathbb{E}\left[\sum_{i=1}^{N} \frac{1}{p_i^t} \left(\left\| g_i^t \right\|_2 - \left\| h_i^t \right\|_2 \right) \left(\left\| g_i^t \right\|_2 + \left\| h_i^t \right\|_2 \right) \right] \\
&\leq \mathbb{E}\left[\sum_{i=1}^{N} \frac{2G}{p_i^t} \left\| g_i^t - h_i^t \right\|_2 \right] \\
&\leq \frac{2G}{\varepsilon_t} \mathbb{E}\left[\sum_{i=1}^{N} \left\| g_i^t - h_i^t \right\|_2 \right]
\end{aligned}$$

where in the third line we used Assumption 1 and the reverse triangle inequality, and in the last line we used the fact that $p^t \in \Delta(\varepsilon_t)$. Since $t \geq t_0$, we can apply proposition 2 on (B) to obtain:

$$\text{(B)} \leq \mathbb{E}\left[6\varepsilon_t N \left(\sum_{i=1}^{N} \left\| h_i^t \right\|_2 \right)^2 \right] \leq 6\varepsilon_t G^2 N^3$$

where the second inequality uses Assumption 1. Finally, using the optimal function values (22) we have for (C):

$$\begin{aligned}
\text{(C)} &= \mathbb{E}\left[\left(\sum_{i=1}^{N}\left\|h_i^t\right\|_2\right)^2 - \left(\sum_{i=1}^{N}\left\|g_i^t\right\|_2\right)^2\right] \\
&= \mathbb{E}\left[\left(\sum_{i=1}^{N}\left(\left\|h_i^t\right\|_2 - \left\|g_i^t\right\|_2\right)\right)\left(\sum_{i=1}^{N}(\left\|g_i^t\right\|_2 + \left\|h_i^t\right\|_2)\right)\right] \\
&\leq 2GN\,\mathbb{E}\left[\sum_{i=1}^{N}\left\|g_i^t - h_i^t\right\|_2\right] \\
&\leq \frac{2G}{\varepsilon_t}\,\mathbb{E}\left[\sum_{i=1}^{N}\left\|g_i^t - h_i^t\right\|_2\right]
\end{aligned}$$

where we again used Assumption 1 and the reverse triangle inequality in the third line. the last inequality follows from the fact that $\varepsilon_t \leq \frac{1}{N}$. Combining the three bounds gives the result. □

B.4 Proof of Lemma 3

Lemma 3. *Under Assumptions 2 and 3, we have:*

$$\mathbb{E}\left[\sum_{i=1}^{N}\left\|g_i^{t+1} - h_i^{t+1}\right\|_2\right] \leq \frac{NLA}{(B+t-1)^{\delta}} + (1-\varepsilon_t)\,\mathbb{E}\left[\sum_{i=1}^{N}\left\|g_i^t - h_i^t\right\|_2\right]$$

Proof. Conditioning on the knowledge of $\{I_1, \ldots, I_{t-1}\}$ we have:

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{N}\left\|g_i^{t+1} - h_i^{t+1}\right\|_2\right] &= \sum_{j=1}^{N} p_j^t\left[\left\|g_j^{t+1} - g_j^t\right\|_2 + \sum_{\substack{i=1\\ i\neq j}}^{N}\left\|g_i^{t+1} - h_i^t\right\|_2\right] \\
&\leq \sum_{j=1}^{N} p_j^t\left[\left\|g_j^{t+1} - g_j^t\right\|_2 + \sum_{\substack{i=1\\ i\neq j}}^{N}\left(\left\|g_i^{t+1} - g_i^t\right\|_2 + \left\|g_i^t - h_i^t\right\|_2\right)\right] \\
&= \sum_{j=1}^{N} p_j^t\left[\sum_{i=1}^{N}\left\|g_i^{t+1} - g_i^t\right\|_2 + \sum_{\substack{i=1\\ i\neq j}}^{N}\left\|g_i^t - h_i^t\right\|_2\right] \\
&\leq NL\sum_{j=1}^{N} p_j^t\left\|x_{t+1} - x_t\right\|_2 + \sum_{i=1}^{N}\left(\sum_{\substack{j=1\\ j\neq i}}^{N} p_j^t\right)\left\|g_i^t - h_i^t\right\|_2 \\
&= NL\,\mathbb{E}\left[\left\|x_{t+1} - x_t\right\|_2\right] + \sum_{i=1}^{N}(1-p_i^t)\left\|g_i^t - h_i^t\right\|_2 \\
&\leq \frac{NLA}{(B+t-1)^{\delta}} + (1-\varepsilon_t)\sum_{i=1}^{N}\left\|g_i^t - h_i^t\right\|_2
\end{aligned}$$

where in the fourth line we used Assumption 2, and in the last line we used Assumption 3 and the fact that $p^t \in \Delta(\varepsilon_t)$. Taking expectation with respect to the choice of $\{I_1, \ldots, I_{t-1}\}$ on both sides we get the result. □

B.5 Lemma 4

Before proving Theorem 1, we first state and prove the following solution of the recursion of Lemma 3 assuming the sequence $\{\varepsilon_t\}_{t=1}^{T}$ is given by (15).

Lemma 4. *Assuming the use of the sequence $\{\varepsilon_t\}_{t=1}^T$ given by (15) we have:*

$$\mathbb{E}\left[\sum_{i=1}^N \left\|g_i^t - h_i^t\right\|_2\right] \leq \frac{K}{(C+t-1)^{2\delta/3}}$$

where:

$$K := \max\left\{\frac{3C^{1-\delta/3}D}{3-2\delta}, C^{2\delta/3}\sum_{i=1}^N \left\|g_i^1 - h_i^1\right\|_2\right\}$$

and:

$$D := \begin{cases} NLA & \text{if } B \geq C \\ (\frac{C}{B})^\delta NLA & \text{if } B < C \end{cases}$$

where A, B, and δ are as in Assumption 3.

Proof.

A simple inequality. Suppose $B \geq C$, then:

$$\frac{NLA}{(B+t-1)^\delta} \leq \frac{NLA}{(C+t-1)^\delta}$$

otherwise, we have:

$$\frac{NLA}{(B+t-1)^\delta} = \frac{(\frac{C}{B})^\delta NLA}{(C+(\frac{C}{B})(t-1))^\delta} \leq \frac{(\frac{C}{B})^\delta NLA}{(C+t-1)^\delta}$$

where the last inequality follows from the fact that $C > B$ and $t \geq 1$. We conclude that:

$$\frac{NLA}{(B+t-1)^\delta} \leq \frac{D}{(C+t-1)^\delta}$$

Induction proof. Let $\varphi(t) := \mathbb{E}\left[\sum_{i=1}^N \left\|g_i^t - h_i^t\right\|_2\right]$, and let $K' := C^{2-2\delta/3}K$. For $t = 1$ the statement holds since:

$$\varphi(1) = \frac{C^{2\delta/3}\varphi(1)}{(C+t-1)^{2\delta/3}} \leq \frac{K}{(C+t-1)^{2\delta/3}}$$

For $t \geq 1$ we have by Lemma 3 and the above inequality:

$$\begin{aligned}
\varphi(t+1) &\leq \frac{D}{(C+t-1)^\delta} + \left(1 - \frac{1}{C^{1-\delta/3}(C+t-1)^{\delta/3}}\right)\varphi(t) \\
&= \frac{aC^{3-\delta}D}{aC^{3-\delta}(C+t-1)^\delta} + \left(1 - \frac{1}{C^{1-\delta/3}(C+t-1)^{\delta/3}}\right)\varphi(t) \\
&\leq \frac{K'}{aC^{3-\delta}(C+t-1)^\delta} + \left(1 - \frac{1}{C^{1-\delta/3}(C+t-1)^{\delta/3}}\right)\frac{K'}{C^{2-2\delta/3}(C+t-1)^{2\delta/3}}
\end{aligned}$$

where $a = \frac{3}{3-2\delta}$ and where the last line follows by the induction hypothesis. To simplify notation let $x := (C+t-1)$, $E := C^{1-\delta/3}$, $\gamma := (1-\frac{1}{a}) = \frac{2\delta}{3}$. Then the above inequality can be rewritten as:

$$\begin{aligned}
\varphi(t+1) &\leq K'\left(\frac{1}{E^2x^{2\delta/3}} - \frac{\gamma}{E^3x^\delta}\right) \\
&= K'\frac{Ex^{\delta/3} - \gamma}{E^3x^\delta} \\
&= K'\frac{E^3x^\delta - \gamma^3}{E^3x^\delta(E^2x^{2\delta/3} + E\gamma x^{\delta/3} + \gamma^2)} \\
&\leq K'\frac{1}{E^2x^{2\delta/3} + E\gamma x^{\delta/3}}
\end{aligned}$$

Now by concavity of $x^{2\delta/3}$ we have:

$$E^2\left[(x+1)^{2\delta/3} - x^{2\delta/3}\right] \leq E^2\frac{2\delta}{3}x^{2\delta/3-1}$$

so that:

$$\begin{aligned}
&E^2x^{2\delta/3} + E\gamma x^{\delta/3} \geq E^2(x+1)^{2\delta/3} \\
&\Leftrightarrow E\gamma x^{\delta/3} \geq E^2\left[(x+1)^{2\delta/3} - x^{2\delta/3}\right] \\
&\Leftarrow \gamma E x^{\delta/3} \geq \frac{2\delta}{3}E^2x^{2\delta/3-1} \\
&\Leftrightarrow x^{1-\delta/3} \geq C^{1-\delta/3} \\
&\Leftrightarrow x \geq C \\
&\Leftrightarrow (C+t-1) \geq C \\
&\Leftrightarrow t \geq 1
\end{aligned}$$

The last statement is true, and therefore so is the first. Replacing in the bound on $\varphi(t+1)$ we get:

$$\varphi(t+1) \leq \frac{K'}{C^{2-2\delta/3}(C+(t+1)-1)^{2\delta/3}} = \frac{K}{(C+(t+1)-1)^{2\delta/3}}$$

which finishes the proof. □

B.6 Proof of Theorem 1

Theorem 1. *Under Assumptions 1 and 2 on the functions f_i and Assumption 3 on the sequence $\{x_t\}_{t=1}^T$, algorithm (10) with the sequence $\{\varepsilon_t\}_{t=1}^T$ given in (15) satisfies the following dynamic regret bound for all $T \geq t_0$:*

$$\mathbb{E}\left[\textit{Regret}_D(T)\right] \leq \mathcal{O}(T^{1-\delta/3}) \tag{16}$$

where $t_0 := \min\{t \in [T] \mid \varepsilon_t \leq \frac{1}{2N}\}$ as in Lemma 2.

Proof. Combining Lemma 4 with Lemma 2 we have the following per-step bound for $t \geq t_0$:

$$\mathbb{E}\left[c_t(p^t) - c_t(q^t)\right] \leq \frac{4GKC^{1-\delta/3} + \frac{6G^2N^3}{C^{1-\delta/3}}}{(C+t-1)^{\delta/3}} =: \frac{K'}{(C+t-1)^{\delta/3}}$$

Summing over the time steps $\{t_0, \ldots, T\}$ we get:

$$\sum_{t=t_0}^{T}\mathbb{E}\left[c_t(p^t) - c_t(q^t)\right] \leq \sum_{t=t_0}^{T}\frac{K'}{(C+t-1)^{\delta/3}}$$

Therefore:

$$\begin{aligned}
\mathbb{E}\left[\text{Regret}_D(T)\right] &= \sum_{t=1}^{t_0-1}\mathbb{E}\left[c_t(p^t) - c_t(q^t)\right] + \sum_{t=t_0}^{T}\mathbb{E}\left[c_t(p^t) - c_t(q^t)\right] \\
&\leq \sum_{t=1}^{t_0-1}\mathbb{E}\left[c_t(p^t)\right] + \sum_{t=t_0}^{T}\frac{K'}{(C+t-1)^{\delta/3}} \\
&\leq \sum_{t=1}^{t_0-1}\sum_{i=1}^{N}\mathbb{E}\left[\frac{1}{p_i^t}\left\|g_i^t\right\|_2^2\right] + K'\int_{t=t_0-1}^{T}\frac{1}{(C+t-1)^{\delta/3}}dt \\
&\leq (t_0-1)\frac{NG^2}{\varepsilon_{t_0}} + K'\frac{(C+T-1)^{1-\delta/3} - (C+t_0-2)^{1-\delta/3}}{1-\delta/3} \\
&\leq (2^{3/\delta}-1)\frac{N^2G^2}{\varepsilon_{t_0}} + K'\frac{(C+T-1)^{1-\delta/3} - (C-1)^{1-\delta/3}}{1-\delta/3} \\
&= \mathcal{O}(T^{1-\delta/3})
\end{aligned}$$

Where the line before the last follows from the fact that $1 \leq t_0 \leq (2^{3/\delta}-1)N+1$ since $C \geq N$. □

Appendix C A new mini-batch estimator

C.1 A class of unbiased estimators

It will be useful for our discussion to consider the following class $\mathcal{C}(p^{t,1}, \dots, p^{t,m})$ of estimators:

$$\hat{g}_b^t(p^{t,1}, \dots, p^{t,m}) := \frac{1}{m}\sum_{j=1}^{m} \hat{g}_j^t(p^{t,j}) \qquad \hat{g}_j^t(p^{t,j}) := \left[\frac{1}{p_{I_t^j}^{t,j}} g_{I_t^j}^t + \sum_{k=1}^{j-1} g_{I_t^k}^t\right]$$

where each $p^{t,j}$ is a distribution on $[N] \setminus \{I_t^1, \dots, I_t^{j-1}\}$. The estimator we proposed in Section 5 is:

$$\hat{g}_b^t = \frac{1}{m}\sum_{j=1}^{m} \hat{g}_j^t \qquad \hat{g}_j^t := \left[\frac{1}{q_{I_t^j}^{t,j}} g_{I_t^j}^t + \sum_{k=1}^{j-1} g_{I_t^k}^t\right] \qquad q_i^{t,j} := \frac{p_i^t}{1 - \sum_{k=1}^{j-1} p_{I_t^k}^t} \tag{19}$$

where the indices $S_t = \{I_t^1, \dots, I_t^m\}$ are sampled without replacement according to p^t. Setting:

$$p_i^{t,j} := \begin{cases} 0 & \text{if } i \in \{I_t^1, \dots, I_t^{j-1}\} \\ q_i^{t,j} & \text{otherwise} \end{cases}$$

we see that our proposed estimator belongs to the class of estimators $\mathcal{C}$ introduced above. The proofs of (a) and (b) of proposition 3 below apply with almost no modification to any estimator in the class $\mathcal{C}$. A natural question then is which estimator in the above-defined class achieves minimum variance. We answer this in the proof of part (c) below, and show that our proposed estimator (19) with $p^t := \operatorname{argmin}_{p\in\Delta}\{c_t(p)\}$ achieves minimum variance.

C.2 Proof of proposition 3

Proposition 3. *Let $S_t^j := \{I_t^1, \dots, I_t^j\}$ for $j \in [m]$ and $S_t^0 := \emptyset$. We have:*

(a) $\mathbb{E}\left[\hat{g}_b^t\right] = g^t$

(b) $\mathbb{E}\left[\left\|\hat{g}_b^t - g^t\right\|_2^2\right] = (1/m^2)\sum_{j=1}^m \mathbb{E}\left[\left\|\hat{g}_j^t - g^t\right\|_2^2\right]$

(c) $\operatorname{argmin}_{p\in\Delta}\{\mathbb{E}\left[\left\|\hat{g}_b^t - g^t\right\|_2^2\right]\} = \operatorname{argmin}_{p\in\Delta}\{c_t(p)\}$

(d) $\mathbb{E}\left[\left\|\hat{g}_{j+1}^t - g^t\right\|_2^2\right] = \left(1 - \mathbb{E}\left[q_{I_t^j}^{t,j}\right]\right)\mathbb{E}\left[\left\|\hat{g}_j^t - g^t\right\|_2^2\right] - \mathbb{E}\left[q_{I_t^j}^{t,j}\left\|\hat{g}_j^t - g^t\right\|_2^2\right]$

where all the expectations in (d) are conditional on S_t^{j-1}.

Proof.

(a) For $j \in [m]$ and conditional on S_t^{j-1} we have:

$$\begin{aligned}
\mathbb{E}\left[\hat{g}_j^t\right] &= \sum_{\substack{i=1 \\ i\notin S_t^{j-1}}}^{N} q_i^{t,j}\left[\frac{1}{q_i^{t,j}} g_i^t + \sum_{k\in S_t^{j-1}} g_k^t\right] \\
&= \sum_{\substack{i=1 \\ i\notin S_t^{j-1}}}^{N} g_i^t + \left(\sum_{k\in S_t^{j-1}} g_k^t\right)\underbrace{\left(\sum_{\substack{i=1 \\ i\notin S_t^{j-1}}}^{N} q_i^{t,j}\right)}_{=1} \\
&= \sum_{i=1}^{N} g_i^t \\
&= g^t
\end{aligned}$$

Taking expectation with respect to the choice of S_t^{j-1}, and taking the average over $j \in [m]$ we get the result.

(b) We have:

$$\mathbb{E}\left[\left\|\hat{g}_b^t - g^t\right\|_2^2\right] = \mathbb{E}\left[\left\|\frac{1}{m}\sum_{j=1}^m \hat{g}_j^t - g^t\right\|_2^2\right]$$
$$= \frac{1}{m^2}\sum_{j=1}^m \mathbb{E}\left[\left\|\hat{g}_j^t - g^t\right\|_2^2\right] + \frac{2}{m^2}\sum_{\substack{j,i \\ j<i}}^m \mathbb{E}\left[\langle \hat{g}_j^t - g^t, \hat{g}_i^t - g^t\rangle\right]$$

To show the claim, it is therefore enough to show that second term is zero. Let $j \in [m-1]$. Conditional on S_t^{i-1} we have:

$$\mathbb{E}\left[\langle \hat{g}_j^t - g^t, \hat{g}_i^t - g^t\rangle\right] = \langle \hat{g}_j^t - g^t, \mathbb{E}\left[\hat{g}_i^t\right] - g^t\rangle = 0$$

where we used the fact that the conditional expectation is zero by part (a). Taking expectation with respect to S_t^{i-1} on both sides yields the result.

(c) As discussed in the previous section, (b) applies to all estimators in the class $\mathcal{C}$, so we have for all such estimators:

$$\mathbb{E}\left[\left\|\hat{g}_b^t(p^{t,1}, \ldots, p^{t,m}) - g^t\right\|_2^2\right] = \frac{1}{m^2}\sum_{j=1}^m \mathbb{E}\left[\left\|\hat{g}_j^t(p^{t,j}) - g^t\right\|_2^2\right]$$

minimizing over $(p^{t,1}, \ldots, p^{t,m})$ by minimizing each term with respect to its variable we get:

$$\operatorname*{argmin}_{(p^{t,1},\ldots,p^{t,m})} \{\mathbb{E}\left[\left\|\hat{g}_b^t(p^{t,1}, \ldots, p^{t,m}) - g^t\right\|_2^2\right]\} = \left(\frac{p^{t,*}}{1 - \sum_{k=1}^{j-1} p^{t,*}_{I_t^k}}\right)_{j=1}^m$$

where:

$$p^{t,*} = \operatorname*{argmin}_{p \in \Delta}\{c_t(p)\} = \frac{\|g_i^t\|_2}{\sum_{j=1}^N \left\|g_j^t\right\|_2}$$

Recalling that our estimator is in $\mathcal{C}$, and noticing that the optimal probabilities over $\mathcal{C}$ are feasible for our estimator we get the result.

(d) Fix $t \in [T]$. We drop the superscript t from p^t, $q_i^{t,j}$, and g_i^t. We also drop the subscript t from I_t^j and S_t^j to simplify notation. Define for $j \in [m]$:

$$x_j := \frac{1}{q_{I^j}^j} g_{I^j} \qquad \mu_j := g^t - \sum_{k=1}^{j-1} g_{I^k}$$

We have from part (a):

$$\mathbb{E}\left[x_j\right] = \mu_j$$

Before proceeding with the proof, we first derive an identity relating q_i^{j+1} and q_i^j:

$$\frac{1}{q_i^{j+1}} = \frac{1 - \sum_{k \in S^j} p_k}{p_i}$$
$$= \frac{1 - \sum_{k \in S^{j-1}} p_k - p_{I^j}}{p_i}$$
$$= \left(1 - \frac{p_{I^j}}{1 - \sum_{k \in S^{j-1}} p_k}\right)\left(\frac{1 - \sum_{k \in S^{j-1}} p_k}{p_i}\right)$$
$$= \left(1 - q_{I^j}^j\right)\frac{1}{q_i^j}$$

Now, conditional on S_t^j, we have:

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\hat{g}_{j+1}^t - g^t\right\|_2^2\right] \\
&= \mathbb{E}\left[\left\|x_{j+1} - \mu_{j+1}\right\|_2^2\right] \\
&= \mathbb{E}\left[\left\|x_{j+1}\right\|_2^2\right] - \left\|\mu_{j+1}\right\|_2^2 \\
&= \left(\sum_{\substack{i=1 \\ i\notin S^j}}^N \frac{1}{q_i^{j+1}} \left\|g_i\right\|_2^2\right) - \left\|\mu_{j+1}\right\|_2^2 \\
&= \left(\sum_{\substack{i=1 \\ i\notin S^{j-1}}}^N \frac{1}{q_i^{j+1}} \left\|g_i\right\|_2^2\right) - \frac{1}{q_{I^j}^{j+1}} \left\|g_{I^j}\right\|_2^2 - \left\|\mu_j - g_{I^j}\right\|_2^2 \\
&= \left(1 - q_{I^j}^j\right)\left(\sum_{\substack{i=1 \\ i\notin S^{j-1}}}^N \frac{1}{q_i^j} \left\|g_i\right\|_2^2\right) - \left(1 - q_{I^j}^j\right) \frac{1}{q_{I^j}^j} \left\|g_{I^j}\right\|_2^2 - \left(\left\|\mu_j\right\|_2^2 - 2\langle \mu_j, g_{I^j}\rangle + \left\|g_{I^j}\right\|_2^2\right) \\
&= \left(1 - q_{I^j}^j\right)\left(\sum_{\substack{i=1 \\ i\notin S^{j-1}}}^N \frac{1}{q_i^j} \left\|g_i\right\|_2^2 - \left\|\mu_j\right\|_2^2\right) - \left(\frac{1}{q_{I^j}^j} \left\|g_{I^j}\right\|_2^2 - 2\langle g_{I^j}, \mu_j\rangle + q_{I^j}^j \left\|\mu_j\right\|_2^2\right) \\
&= \left(1 - q_{I^j}^j\right) \mathbb{E}\left[\left\|x_j - \mu_j\right\|_2^2\right] - q_{I^j}^j \left\|x_j - \mu_j\right\|_2^2 \\
&= \left(1 - q_{I^j}^j\right) \mathbb{E}\left[\left\|\hat{g}_j^t - g^t\right\|_2^2\right] - q_{I^j}^j \left\|\hat{g}_j^t - g^t\right\|_2^2
\end{aligned}
$$

where the expectation in the last two lines is conditional on S^{j-1}. Taking expectation with respect to S^{j-1} on both sides yields the result.

□

Appendix D Extension to constant step-size SGD

While our analysis heavily relies on the assumption of decreasing step-sizes, we have found empirically that a slight modification of our method works just as well when a constant step-size is used. We propose the following epsilon sequence to account for the use of a constant step-size:

$$\varepsilon_t = \frac{1}{C^{1-\delta/3}(C + m(t-1))^{\delta/3}} + p_{min} \tag{23}$$

for a constant $p_{min} \in [0, 1/N]$ and the following condition on C:

$$C \leq \frac{1}{\frac{1}{N} - p_{min}}$$

which ensures that $\varepsilon_1 \leq 1/N$. We ran the same experiment on MNIST, IJCNN1, and CIFAR10 as in Section 6, but with a constant step-size $\alpha_t = \alpha = \frac{m}{2NL}$, and the epsilon sequence (23) with $p_{min} = \frac{1}{5N}$ and $C = \frac{1}{\frac{1}{N} - p_{min}}$, and $\delta = 1$. The results are displayed in figure 3, showing a similar performance compared to the decreasing step-sizes case. Note that choosing a too small p_{min} can start to deteriorate the performance of the algorithm. It is still unclear how to set p_{min} so as to guarantee good performance, but our experiments suggest that setting $\frac{1}{5N}$ is a safe choice.

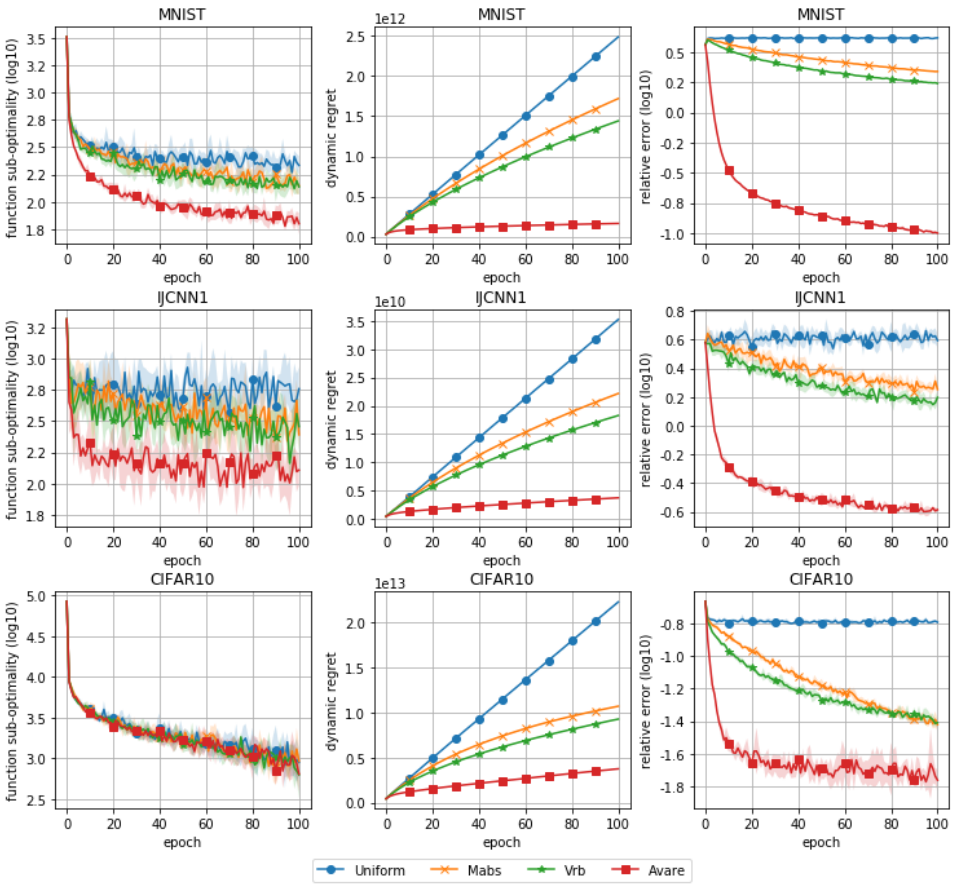

Figure 3: Comparison of the performance of importance samplers on an l2-regularized softmax regression model on three real world datasets: MNIST (top), IJCNN1 (middle), CIFAR10 (bottom). For this set of experiments, SGD was run using a constant step size.

Supplementary references

[37] Zalan Borsos, Andreas Krause, and Kfir Y Levy. Online Variance Reduction for Stochastic Optimization. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018*, volume 75 of *Proceedings of Machine Learning Research*, pages 324–357. PMLR, 2018.

[38] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009.