ADASTAR: A Method for Adapting to Interpolation

author names withheld

Under Review for OPT 2022

Abstract

Stochastic convex optimization methods are much faster at minimizing *interpolation problems*—problems where all sample losses share a common minimizer—than non-interpolating problems. However, stochastic gradient methods need to use step sizes tailored for the interpolation setting, which are sub-optimal for non-interpolating problems, to attain these fast rates. This is problematic because verifying whether a problem is interpolating, without minimizing it, is difficult. Moreover, because interpolation is not a stable property—the addition of a single datapoint can transform an interpolating dataset into a non-interpolating one—we would like our methods to get the fast interpolation rate when it can while being robust to these perturbations. We address these problems by presenting an adaptive stochastic gradient method, termed ADASTAR, which attains the optimal, fast rate on smooth interpolation problems (up to log factors) and gracefully degrades with the minimal objective value for non-interpolating problems. We use this method as a building block to construct another stochastic gradient method, termed ADASTAR-G which adapts to interpolation and growth conditions, getting even faster rates.

1. Introduction

In this work, we study stochastic convex optimization, where we are given n samples s to solve

$$\begin{aligned} &\text{minimize } \ f(x) = \mathbb{E}[F(x;S)] = \int_\Omega F(x;s)dP(s) \\ &\text{subject to } \ x \in \mathcal{X}. \end{aligned} \tag{1}$$

We let Ω denote the sample space, and S be an associated Ω-valued random variable. For each sample $s \in \Omega$, $F(\cdot;s) : \mathbb{R}^d \to \mathbb{R}_+$ is a closed, convex, **non-negative** function which is differentiable on the closed, convex domain $\mathcal{X}$ with diameter $\text{diam}(\mathcal{X}) \leq D$. Throughout the paper, we let $\mathcal{X}^\star := \text{argmin}_{x\in\mathcal{X}} f(x)$, use $x^\star$ to denote an arbitrary element of $\mathcal{X}^\star$, and let $f^\star := f(x^\star)$.

In particular, we will study problem (1) in the interpolation setting—the setting where there exists a solution which simultaneously minimizes all of the sample losses. Interpolation problems are everywhere in modern machine learning. Noiseless least squares, over-parameterized least squares, classification problems with margin, and even the training loss of many deep learning problems are all examples of interpolation problems. In this work we will use the following formal definition for interpolation problem.

Definition 1 (Interpolation Problem) *For a non-negative F, Problem (1) is an interpolation problem if $F(x^\star;s) = 0$ for P-almost every $s \in \Omega$. i.e., $f^\star = 0$.*

© .

We know that stochastic convex optimization methods are much faster at minimizing interpolation problems than non-interpolating problems [7, 12, 13, 15–17]. Empirical risk minimization, for example, has $O(1/n + \sqrt{f^\star/n})$ convergence rate on problem (1) assuming smoothness; i.e. it has a $O(1/n)$ error on interpolation problems, and the standard $O(1/\sqrt{n})$ rate on non-interpolation problems [15]. Methods which only require a single pass over the training data, such as stochastic gradient methods, can attain the same improvements but require using step sizes tailored for the interpolation setting, which are sub-optimal for non-interpolating problems [15]. This is problematic because verifying whether a problem is interpolating, without minimizing it, is difficult. Moreover, because interpolation is not a stable property—the addition of a single datapoint can transform an interpolating dataset into a non-interpolating one—we would like our methods to get the fast interpolation rate when it can while being robust to these perturbations. In this work, we present ADASTAR to meet these desiderata.

While the majority of this paper is dedicated to studying interpolation, we also would like to get even faster rates on realistic problems. To this end, we will also study how interpolation interplays with problems which have κ-growth defined as follows:

Definition 2 (κ-growth) *For $\lambda > 0$ and $\kappa \geq 0$, f has (λ, κ)-growth if and only if*

$$f(x) - f^\star \geq \frac{\lambda}{\kappa} \operatorname{dist}(x, \mathcal{X}^\star)^\kappa.$$

Problems which satisfy growth conditions are ubiquitous; linear regression for example has quadratic ($\kappa = 2$) growth. The smaller κ is for a population function, the sharper the growth rate around the minimizer, and the faster an algorithm can minimize the population loss—$O(1/n^{\frac{\kappa}{2(\kappa-1)}})$ instead of the standard $O(1/\sqrt{n})$. Growth and interpolation are synergistic, mutually amplifying convergence rate improvements. For example, we can solve interpolating, quadratic ($\kappa = 2$) growth problems exponentially fast, while we can only provide a polynomial $O(1/n)$ convergence rate guarantee for problems which exclusively interpolate or exclusively have quadratic growth. However, as we will see in Section 3, unlike linear regression, κ-growth is not always a straightforward property to verify. For this reason, in this paper, we present ADASTAR-G: an algorithm which adapts to growth conditions and interpolation. For problems that do not satisfy any growth conditions i.e., $\kappa \to \infty$, ADASTAR-G recovers the same rates as ADASTAR.

1.1. Our contributions

Our technical contributions are as follows.

1. We present an adaptive stochastic gradient method, termed ADASTAR, which has a convergence rate of $\tilde{O}(1/n + \sqrt{f^\star/n})$, without knowledge of $f^\star$. In other words, it attains the minimax optimal fast rate on interpolation problems (up to log factors) and gracefully degrades with the minimal objective value for non-interpolation problems.

2. We present a second adaptive stochastic gradient method, termed ADASTAR-G, which adapts to interpolation and κ-growth conditions for $\kappa > 2$. In particular, with only knowledge of a lower bound on κ, our algorithm has a convergence rate of $\tilde{O}(1/n^{\frac{\kappa}{\kappa-2}})$ on interpolating κ-growth problems and a rate of $\tilde{O}(1/n^{\frac{\kappa}{2(\kappa-1)}})$, which is optimal up to log factors, on non-interpolating κ-growth problems.

We note that information theoretic lower bounds for interpolating κ-growth problems for $\kappa > 2$—which would prove or disprove the optimality (up to log factors) of ADASTAR-G on said problems—are not known. We leave this open problem for future work.

1.2. Related Work

As mentioned earlier, problems with growth are ubiquitous, and thus, have also been the focus of a lot of work. Hazan and Kale [10] present a stochastic gradient method which is optimal, including log factors, for quadratic growth problems. Chatterjee et al. [6], Juditsky and Nesterov [11], Ramdas and Singh [14] develop methods which attain a faster $O(1/n^{\frac{\kappa}{2(\kappa-1)}})$ on κ-growth problems compared to the standard $O(1/\sqrt{n})$ rate. Asi et al. [4] present information theoretic lower bounds showing these faster rates are optimal.

Interpolation problems with growth have also been extensively studied. Vaswani et al. [16] showed that SGD has a linear convergence rate (exponentially fast) on interpolation problems with growth. Asi and Duchi [2], Asi et al. [3] develop new methods that are more stable than SGD, which also get these fast rates on interpolation problems with growth. Chadha et al. [5] develop lower bounds showing the optimality of these methods.

Notation Throughout this work, F is β-smooth; i.e., $\|F'(x;s) - F'(y;s)\|_2 \leq \beta \|x-y\|_2$. We will let $h'(x)$ denote the gradient of a differentiable function h at x. We may use g_k in place of $F'(x_k, s_k)$ for notational simplicity. We let $[n]$ denote the sequence of integers $1, \ldots, n$. For a point $u \in \mathbb{R}^d$ and a set $V \subset \mathbb{R}^d$, we let $\operatorname{dist}(u, V) := \inf_{v \in V} \|u - v\|_2$.

2. Adapting to Interpolation Problems

In this section, we present our algorithm ADASTAR, which adapts to minimal function value $f^\star$, attaining the optimal fast $1/n$ rate, up to log factors, on interpolation problems. ADASTAR executes $T = \log(n)$ epochs of SGD—each call of SGD is executed for n iterations—which totals to using $O(n \log n)$ samples. We run SGD with a special adaptive step size choice that depends on the parameter value outputted by the SGD algorithm from the previous epoch. In particular, the step size of the ith iteration of the tth call of SGD is chosen proportional to $1/\sqrt{\sum_{k=1}^{i} F(\hat{x}_t; s_k)}$, where $\hat{x}_t$ is the parameter returned by $t-1$ epoch's call of SGD. Intuitively, as t gets larger, the parameter returned by t^{th} call of SGD should get closer to $x^\star$, implying that $F(\hat{x}_t; s_k)$ should be closer to $f^\star$. This in turn means each subsequent call of SGD has a more aggressive step size choice. The fact that $\mathbb{E}[F(\hat{x}_t; s_k)] \geq f^\star$ intuitively lower bounds how large this step size choice can be, preventing the algorithm from being overly aggressive and diverging. The fact that this lower bound depends on the optimal function value $f^\star$ is precisely why this algorithm is adaptive to $f^\star$. For problems where $f^\star$ is smaller, ADASTAR will be able to use more aggressive step sizes. The full description of ADASTAR can be found in Algorithm 1; the adaptive-step-size SGD algorithm that ADASTAR calls is described in Algorithm 2.

Our convergence results for ADASTAR rely on relating the error guarantees epoch t with the error guarantee of epoch $t-1$; this relation is presented in Lemma 3. The proof of Lemma 3 can be found in Section A.1.1.

Algorithm 1: ADASTAR: Adapting to $f^\star$ algorithm

Data: F: sample function being optimized; T: epochs; n: number of iterations per epoch; $\mathcal{X}$: parameter domain; x_1: initial parameter; β: smoothness parameter
Result: x_T: estimate of optimal parameter
$D \leftarrow$ diameter of $\mathcal{X}$
for $t \leftarrow 1$ **to** T **do**
  $\hat{x}_{t+1} \leftarrow$ SGD-HELPER$(F, n, \mathcal{X}, \hat{x}_t, \beta)$
end
return $\hat{x}_T$

Algorithm 2: SGD-HELPER

Data: F: sample function being optimized; n: number of iterations; $\mathcal{X}$: parameter domain; x_1: initial parameter; β: smoothness parameter
Result: $\hat{x}$: estimate of optimal parameter
for $i \leftarrow 1$ **to** n **do**
  $\alpha_i \leftarrow \min\left(\frac{1}{4\beta}, \frac{D}{2\sqrt{2\beta\sum_{k=1}^{i} F(x_1;s_k)}}\right)$
  $x_{i+1} \leftarrow \Pi_{\mathcal{X}}(x_i - \alpha_i F'(x_i, s_i))$
end
return $\hat{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

Lemma 3 *Let F be β-smooth and non-negative. Let $\{\hat{x}_t\}_{t=1}^T$ be the iterates generated by running Algorithm 1 on Problem (1). We have for all $t \in [T]$,*

$$\mathbb{E}[f(\hat{x}_t)] - f^\star \leq \frac{4\beta D^2}{n} + \frac{3D\sqrt{2\beta\mathbb{E}[f(\hat{x}_{t-1})]}}{\sqrt{n}}.$$

Once we have this relation, we can analyze the recurrence to obtain a non-recursive guarantee on the final epoch of ADASTAR. This result is presented in Theorem 4. The proof of Theorem 4 can be found in Appendix A.

Theorem 4 *Let F be β-smooth and non-negative. Let $\{\hat{x}_t\}_{t=1}^T$ be the iterates generated by running Algorithm 1 on Problem (1) with $T = \log_4(n) + 1$. We have, with $c = 18$, the error guarantee*

$$\mathbb{E}[f(\hat{x}_T)] - f^\star \leq \frac{(4c + 1/2)\beta D^2}{n} + \sqrt{\frac{4cD^2\beta f^\star}{n}}.$$

We also provide a high probability result for ADASTAR in Corollary 5. The high probability result will be useful for proving guarantees about ADASTAR-G. The proof of Corollary 5 is very similar to the proof of Theorem 4, except that it uses a high probability version of Lemma 3 instead (i.e., Lemma 12). The full proof details can be found in Appendix B.

Corollary 5 *Let F be β-smooth, non-negative, and bounded above by $B < \infty$. Let $\{\hat{x}_t\}_{t=1}^T$ be the iterates generated by running Algorithm 1 on Problem (1) with $T = \log_4(n) + 1$. With probability*

at least $1-\varepsilon$*, we have*

$$f(\hat{x}_T) - f^\star \lesssim \frac{\log^2\left(\frac{\log n}{\varepsilon}\right)}{n} + \log\left(\frac{\log n}{\varepsilon}\right)\sqrt{\frac{1}{n}}.$$

ADASTAR uses a total of $O(n \log n)$ samples to get an $O(1/n + \sqrt{f^\star/n})$ error guarantee. In other words, relative to the optimal, non-adaptive first order algorithm (constant step size SGD) which leverages knowledge of $f^\star$, ADASTAR has to use $\log n$ times more samples to adapt to $f^\star$.

3. Adaptating to Growth and Interpolation

Problems in practice often also satisfy some type of local growth condition like κ-growth. As mentioned in the introduction, these problems enjoy faster convergence guarantees: $O(1/n^{\frac{\kappa}{2(\kappa-1)}})$, for $\kappa > 1$, compared to the standard $O(1/\sqrt{n})$ rate. Sometimes, growth conditions are dictated by the loss function and thus are straightforward to verify—e.g., linear regression has quadratic growth. In other circumstances, κ depends heavily on the sampling distribution, making them difficult or impossible to verify a priori. Consider a simple one dimensional classification example.

Example 1 (1D seperable classification) *Suppose we are interested in classifying inputs* $a \in \mathbb{R}$ *into its correct binary label* $b \in \{\pm 1\}$*. Let the sample loss function be* $F(x;(a,b)) = \phi(xab)$*, where* $\phi(t) = (1-t)_+^\gamma$ *for some* $\gamma \geq 1$*. Let* $x^\star > 0$ *be the smallest real number such that* $f(x^\star) = \mathbb{E}[\phi(x^\star ab)] = 0$*; i.e., it is the perfect classifier of the data with the smallest margin. Suppose there exists* $B \in \mathbb{R}$ *such that* $x^\star \leq B$*. Let* $x < x^\star$ *(a classifier that incurs error). Let* $z := ab$ *has a density* p*, and suppose that* x *is large enough such that* p *has some growth locally on the boundary of the its support:* $p(z) \geq (z - 1/x^\star)^\eta$ *for* $z \in [1/x^\star, 1/x]$*. Then, the error of* x *can be lower bounded as*

$$\begin{aligned}
\mathbb{E}[\phi(xz)] &= \int_{1/x^\star}^{1/x} (1-xz)^\gamma p(z) dz \geq \int_{1/x^\star}^{1/x} x^\gamma \left(\frac{1}{x} - z\right)^\gamma \left(z - \frac{1}{x^\star}\right)^\eta dz \\
&= \left(\frac{x\gamma}{\eta+1}\right)^\gamma \int_{1/x^\star}^{1/x} \left(z - \frac{1}{x^\star}\right)^{\eta+\gamma} dz \\
&= \left(\frac{1}{x}\right)^{\eta+1} \left(\frac{1}{x^\star}\right)^{\eta+\gamma+1} \left(\frac{\gamma}{\eta+1}\right)^\gamma \left(\frac{1}{\eta+\gamma+1}\right) (x^\star - x)^{\eta+\gamma+1} \\
&\gtrsim |x^\star - x|^{\eta+\gamma+1}.
\end{aligned}$$

The second equality comes from repeatedly reapplying integration by parts. The last equality uses the bound B*. Thus, this problem has* $\kappa = \eta + \gamma + 1$ *growth.*

The practitioner could get faster rates if they knew κ, but κ is difficult to obtain because η is a data dependent quantity. We acknowledge the example is unrealistic—the optimal strategy for the practitioner reduces to examining the sign of a. However, this strategy would not generalize to higher dimensional settings, or settings where data that has been shifted. The main purpose of the one-dimensional example is to highlight how these more complex classification settings, may also have growth conditions which are hard to verify a priori. This hope that we can leverage

unverifiable growth conditions motivates developing methods which are ignorant of and can adapt to κ and interpolation.

To address this issue, we present ADASTAR-G, which adapts to κ-growth and interpolation, and only requires knowledge of a lower bound on κ, denoted by $\underline{\kappa}$. ADASTAR-G executes $K = O(\log n)$ epochs of ADASTAR—recall, ADASTAR in turn runs $T = O(\log n)$ epochs of adaptive SGD for n iterations—which totals to using $O(n \log^2 n)$ samples. This epoch algorithm leverages the fact that growth, by definition, gives us a way of converting bounds on the function error into bounds on parameter error. Ideally we would shrink the domain in proportion to this guarantee (which depends on κ) with every epoch. The idea being because the diameter of the space is now smaller, we can choose a more refined step size as well, leading to better convergence guarantees. However, since κ is unknown, ADASTAR-G instead shrinks the diameter of the set by half with every epoch. Up to some epoch, which we term k_0, this halving will work perfectly and will give us the fast rates we wanted. After epoch k_0, the shrinking will be too aggressive, but because geometric sums converge very quickly, the extra error that this geometric halving induces will not matter. Thus, this strategy naturally adapts to the growth parameter κ, so long as the practitioner runs ADASTAR-G for enough epochs, necessitating the lower bound on κ. Finally, because ADASTAR is the subroutine used in each epoch of ADASTAR-G, ADASTAR-G also naturally adapts to interpolation as well. The pseudocode for ADASTAR-G can be found in Algorithm 3, and the convergence guarantee for the algorithm can be found below in Theorem 6. The proof of Theorem 6 can be found in Appendix C.

Algorithm 3: ADASTAR-G: Adapting to $f^\star$ and growth algorithm

Data: F: sample function being optimized; K: epochs; T: number of epochs of AdaVal; n: number of iterations per epoch of AdaVal; $\mathcal{X}_1$: parameter domain; x_1: initial parameter; β: smoothness parameter

Result: x_K: estimate of optimal parameter

$D_0 \leftarrow$ diameter of $\mathcal{X}$

for $k \leftarrow 1$ **to** K **do**

  $x_{k+1} \leftarrow \text{ADASTAR}(F, T, n, \mathcal{X}_k, x_k, \beta)$

  $D_{k+1} = D_k/2$

  $\mathcal{X}_{k+1} = \{x \in \mathcal{X}_1 : \|x - x_{k+1}\|_2 \leq D_{k+1}\}$

end

return x_K

Theorem 6 *Let F be β-smooth and non-negative. Let f have κ-growth with $\kappa \geq \underline{\kappa} > 2$. Let c_0 be some absolute constant, and $\varepsilon \in (0, 1)$. Let $\{x_k\}_{k=1}^K$ be generated from Algorithm 3 with $K = \left\lfloor \log_2 \left(\left(\frac{n\lambda}{c_0 \underline{\kappa} \beta \log(K \log n/\varepsilon)} \right)^{\frac{1}{\underline{\kappa}-2}} D_0 \right) \right\rfloor + 1$ and $T = \log_4(n) + 1$. With probability at least $1 - \varepsilon$, treating $f^\star$ as a constant*

$$f(x_K) - f^\star \lesssim \left(\frac{\beta \log(K \log n/\varepsilon)}{n} \right)^{\frac{\kappa}{\kappa-2}} + \left(\sqrt{\frac{\beta \log(K \log n/\varepsilon)}{n}} \sqrt{f^\star} \right)^{\frac{\kappa}{\kappa-1}}.$$

To summarize, ADASTAR-G uses $n \log^2 n$ samples to get an error $\tilde{O}(1/n^{\frac{\kappa}{\kappa-2}} + (\sqrt{f^\star/n})^{\frac{\kappa}{\kappa-1}})$. The log factors represent the price ADASTAR-G pays to adapt to growth and interpolation.

References

[1] Alekh Agarwal and John C. Duchi. Distributed delayed stochastic optimization. In *Advances in Neural Information Processing Systems 24*, 2011.

[2] Hilal Asi and John C. Duchi. Stochastic (approximate) proximal point methods: Convergence, optimality, and adaptivity. *SIAM Journal on Optimization*, 29(3):2257–2290, 2019. URL https://arXiv.org/abs/1810.05633.

[3] Hilal Asi, Karan Chadha, Gary Cheng, and John C. Duchi. Minibatch stochastic approximate proximal point methods. In *Advances in Neural Information Processing Systems 33*, 2020.

[4] Hilal Asi, Daniel Lévy, and John C. Duchi. Adapting to function difficulty and growth conditions in private optimization. In *NeurIPS*, 2021.

[5] Karan N. Chadha, Gary Cheng, and John C. Duchi. Accelerated, optimal, and parallel: Some results on model-based stochastic optimization. In *ICML*, 2022.

[6] Sabyasachi Chatterjee, John Duchi, John Lafferty, and Yuancheng Zhu. Local minimax complexity of stochastic convex optimization. In *Advances in Neural Information Processing Systems 29*, 2016.

[7] Andrew Cotter, Ohad Shamir, Nati Srebro, and Karthik Sridharan. Better mini-batch algorithms via accelerated gradient methods. In *Advances in Neural Information Processing Systems 24*, 2011.

[8] John C. Duchi. Introductory lectures on stochastic convex optimization. In *The Mathematics of Data*, IAS/Park City Mathematics Series. American Mathematical Society, 2018.

[9] John C. Duchi. Information theory and statistics. Lecture Notes for Statistics 311/EE 377, Stanford University, 2019. URL http://web.stanford.edu/class/stats311/lecture-notes.pdf. Accessed May 2019.

[10] Elad Hazan and Satyen Kale. An optimal algorithm for stochastic strongly convex optimization. In *Proceedings of the Twenty Fourth Annual Conference on Computational Learning Theory*, 2011. URL http://arxiv.org/abs/1006.2425.

[11] Anatoli Juditsky and Yuri Nesterov. Deterministic and stochastic primal-dual subgradient algorithms for uniformly convex minimization. *Stochastic Systems*, 4(1):44—-80, 2014.

[12] Chaoyue Liu and Mikhail Belkin. Accelerating sgd with momentum for over-parameterized learning. In *International Conference on Learning Representations*, 2020.

[13] Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[14] Aaditya Ramdas and Aarti Singh. Optimal rates for stochastic convex optimization under tsybakov noise condition. In *Proceedings of the 30th International Conference on Machine Learning*, pages 365–373, 2013.

[15] Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. Smoothness, low noise and fast rates. In *nips2010*, pages 2199–2207, 2010.

[16] Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, 2019.

[17] Blake E. Woodworth and Nathan Srebro. An even more optimal stochastic optimization algorithm: Minibatching and interpolation learning. In *Advances in Neural Information Processing Systems 34*, 2021.

Appendix A. Proof of Theorem 4

A.1. Supporting Lemmas

Lemma 7 *For a β-smooth and non-negative function $h : \mathcal{X} \to \mathbb{R}$, we have $\|h'(x)\|_2 \leq \sqrt{4\beta h(x)}$ for all $x \in \mathcal{X}$.*

Proof See the proof of Lemma 3.1 in [15]. □

Lemma 8 *One step of* SGM with step size α_i on a function $F(\cdot; s)$ has the following guarantee for any $y \in \mathcal{X}$

$$F(x_i; s) \leq F(y; s) + \frac{1}{2\alpha_i}(\|x_i - y\|_2^2 - \|x_{i+1} - y\|_2^2) + \frac{\alpha_i}{2}\|g_i\|_2^2 .$$

Proof Using convexity, we have that

$$F(x_i; s) - F(y; s) \leq \langle g_i, x_i - y\rangle = \langle g_i, x_i - x_{i+1}\rangle + \langle g_i, x_{i+1} - y\rangle.$$

Since x_{i+1} solves a convex optimization problem $x_{i+1} = \operatorname{argmin}_{x\in\mathcal{X}} \langle g_i, x - x_i\rangle + \frac{1}{2\alpha_i}\|x - x_i\|_2^2$, for all $y \in \mathcal{X}$, we have the following optimality condition holds

$$\langle g_i + \frac{1}{\alpha_i}(x_{i+1} - x_i), y - x_{i+1}\rangle \geq 0,$$

which means that

$$\begin{aligned} F(x_i; s) - F(y; s) &\leq \langle g_i, x_i - x_{i+1}\rangle + \frac{1}{\alpha_i}\langle x_{i+1} - x_i, y - x_{i+1}\rangle \\ &= \langle g_i, x_i - x_{i+1}\rangle + \frac{1}{2\alpha_i}\left(\|x_i - x^\star\|_2^2 - \|x_{i+1} - x^\star\|_2^2 - \|x_{i+1} - x_i\|_2^2\right), \end{aligned}$$

where the equality is the result of some algebra. By Fenchel-Young, we have that

$$\langle g_i, x_i - x_{i+1}\rangle \leq \frac{1}{2\alpha_i}\|x_i - x_{i+1}\|_2^2 + \frac{\alpha_i}{2}\|g_i\|_2^2 .$$

Combining the two inequalities, gives us the desired result.
□

Lemma 9 *Let $\{x_i\}_{i=1}^n$ be the iterates generated by running projected stochastic subgradient descent with a non-increasing sequence of step sizes $\{\alpha_i\}_{i=1}^n$ on Problem (1). We have that*

$$\frac{1}{n}\sum_{i=1}^n f(x_i) - f^\star \leq \frac{D^2}{2\alpha_n n} + \frac{1}{2n}\sum_{i=1}^n \alpha_i \|g_i\|_2^2 + \frac{1}{n}\sum_{i=1}^n \langle f'(x_i) - g_i, x_i - x^\star\rangle$$

Proof See Equation (3.4.8) from Theorem 3.4.7 from [8]. □

A.1.1. PROOF OF SUPPORTING LEMMA 3

Let $\mathcal{F}_t := \sigma(\{S_{j,i}\}_{j\in[t],i\in[n]})$ be the σ-field generated by the first t epochs of Algorithm 1, so $\hat{x}_t \in \mathcal{F}_{t-1}$. For this proof, let $x_{t,i}$ (or x_i) denote the output of the ith iteration of the subalgorithm Algorithm 2 in epoch t of Algorithm 1. $s_{t,i}$ denotes the sample drawn in the ith iteration of the subalgorithm Algorithm 2 in the t epoch of Algorithm 1.

Lemma 8 gives us

$$F(x_{t,i}; s_{t,i}) \leq F(\hat{x}_{t-1}; s) + \frac{1}{2\alpha_i}(\|x_{t,i} - \hat{x}_{t-1}\|_2^2 - \|x_{t,i+1} - \hat{x}_{t-1}\|_2^2) + \frac{\alpha_i}{2}\|g_{t,i}\|_2^2 .$$

Combining Lemma 8 and Lemma 7 gives

$$\begin{aligned}\frac{1}{4\beta}\sum_{i=1}^n \alpha_i \|g_{t,i}\|_2^2 &\leq \sum_{i=1}^n \alpha_i F(x_{t,i}; s_{t,i}) \\ &\leq \sum_{i=1}^n \alpha_i F(\hat{x}_{t-1}; s_{t,i}) + \frac{1}{2}(\|x_{t,1} - \hat{x}_{t-1}\|_2^2 - \|x_{t,n+1} - \hat{x}_{t-1}\|_2^2) + \frac{1}{2}\sum_{i=1}^n \alpha_i^2 \|g_{t,i}\|_2^2 .\end{aligned}$$

Using the fact that $\alpha_i \leq \frac{1}{4\beta}$ and rearranging, we get

$$\sum_{i=1}^n \alpha_i \|g_{t,i}\|_2^2 \leq 4\beta D^2 + 8\beta \sum_{i=1}^n \alpha_i F(\hat{x}_{t-1}; s_{t,i})$$

Combining this result with Lemma 9 and using the fact that for all $i \in [n]$, $\mathbb{E}[\langle f'(x_{t,i}) - g_{t,i}, x_{t,i} - x^\star\rangle \mid \mathcal{F}_{t-1}] = 0$,

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^n \mathbb{E}[f(x_{t,i})|\mathcal{F}_{t-1}] - f^\star &\leq \mathbb{E}\left[\frac{D^2}{2\alpha_n n} + \frac{1}{2n}\sum_{i=1}^n \alpha_i \|g_{t,i}\|_2^2 \middle| \mathcal{F}_{t-1}\right] \\ &\leq \mathbb{E}\left[\frac{2\beta D^2}{n} + \frac{D^2}{2\alpha_n n} + \frac{4\beta}{n}\sum_{i=1}^n \alpha_i F(\hat{x}_{t-1}, s_{t,i}) \middle| \mathcal{F}_{t-1}\right].\end{aligned}$$

We now plug in the step size $\alpha_i = \min\left(\frac{1}{4\beta}, \frac{D}{2\sqrt{2\beta\sum_{k=1}^i F(\hat{x}_{t-1};s_{t,k})}}\right)$. We proceed by cases. If $\alpha_n = \frac{D}{2\sqrt{2\beta\sum_{k=1}^n F(\hat{x}_{t-1};s_{t,k})}}$, we have that

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^n \mathbb{E}[f(x_i)|\mathcal{F}_{t-1}] - f^\star &\leq \mathbb{E}\left[\frac{2\beta D^2}{n} + \frac{D\sqrt{2\beta}}{n}\sqrt{\sum_{i=1}^n F(\hat{x}_{t-1}; s_{t,i})} + \frac{D\sqrt{2\beta}}{n}\sum_{i=1}^n \frac{F(\hat{x}_{t-1}; s_{t,i})}{\sqrt{\sum_{k=1}^i F(\hat{x}_{t-1}; s_{t,k})}} \middle| \mathcal{F}_{t-1}\right] \\ &\leq \frac{2\beta D^2}{n} + \mathbb{E}\left[\frac{3D\sqrt{2\beta}}{n}\sqrt{\sum_{k=1}^n F(\hat{x}_{t-1}; s_{t,i})} \middle| \mathcal{F}_{t-1}\right] \\ &\leq \frac{2\beta D^2}{n} + \frac{3D\sqrt{2\beta f(\hat{x}_{t-1})}}{\sqrt{n}}.\end{aligned} \tag{2}$$

The second inequality comes from the fact that for any non-negative sequence $a_1, \ldots, a_n$, $\sum_{i=1}^n \frac{a_i}{\sqrt{\sum_{k=1}^i a_k}} \leq 2\sqrt{\sum_{i=1}^n a_i}$. The third inequality comes from Jensen's inequality. In the other case where $\alpha_n = \frac{1}{4\beta}$, we have that

$$\frac{1}{n}\sum_{i=1}^n \mathbb{E}[f(x_i)|\mathcal{F}_{t-1}] - f^\star \leq \frac{4\beta D^2}{n} + \frac{D\sqrt{2\beta f(\hat{x}_{t-1})}}{\sqrt{n}}. \tag{3}$$

Taking the max of these two bounds, taking expectations, using online-to-batch conversion (i.e., Jensen's inequality), and concavity of square root (Jensen's inequality again), we get the desired result.

A.2. Proof of result

Let $Q \coloneqq \sqrt{\frac{cD^2\beta}{n}}$. Further let $u_t = \mathbb{E}[f(\hat{x}_t)] - f^\star$. From Lemma 3 , we have that

$$u_{t+1} \leq h(u_t) \coloneqq Q\sqrt{u_t + f^\star} + Q^2.$$

Let $r(u) = u$ be the identity function. We note that our analysis centers around analyzing the convergence of repeatedly h. h is a squareroot function, such that $r(0) = 0 \leq Q\sqrt{f^\star} + Q^2 = h(0)$. Since linear functions grow faster than square root functions, we there must exist a point $u^\star$ such that $r(u^\star) = h(u^\star)$ and $r(u) > h(u)$ for $u \geq u^\star$. With some algebra, we can show that

$$u^\star = \frac{3Q^2 + Q\sqrt{5Q^2 + 4f^\star}}{2} \leq \frac{3Q^2 + \sqrt{5}Q^2 + 2Q\sqrt{f^\star}}{2} \leq \tilde{u} \coloneqq 4\frac{c\beta D^2}{n} + \sqrt{\frac{cD^2\beta f^\star}{n}}.$$

Suppose our initial choice of $\hat{x}_1$ is such that $u_1 \leq u^\star$. In this case, we have $u_2 = h(u_1) > r(u_1) = u_1$; further, since h and r are monotonically increasing we have $u_2 = h(u_1) < h(u^\star) = u^\star$. By induction, we have that in this setting, $u_t \leq u^\star \leq \tilde{u}$ for all $t \geq 1$. Thus, the bound of our theorem covers this setting.

Thus, it suffices to consider the alternative setting where $\hat{x}_1$ is selected such that $u_1 > u^\star$. In this setting we know our error is monotonically decreasing because $u_t = h(u_{t-1}) \leq r(u_{t-1}) = u_{t-1}$. In order to get a convergence rate, we need to do more analysis. h is difficult to directly analyze because of the square root, but since $u^\star$ is the fixed point of h, it suffices to analyze a linear upper bound of h which shares (nearly) the same fixed point as h. In particular, by concavity, we will upper bound h by the linear approximation around $\tilde{u}$ to get a recurrence relation. For reference, we have that $h'(u) = \frac{Q}{2\sqrt{u+f^\star}}$ where $h'(\tilde{u}) = \frac{1}{2\sqrt{4+\sqrt{f^\star}/Q}+f^\star/Q^2} \leq 1/4$

We proceed by cases. If $u_t \leq \tilde{u}$, we are done. Otherwise, we use the fact that $u_t > \tilde{u}$ and $h(\tilde{u}) \leq \tilde{u}$ because $\tilde{u} \geq u^\star$ to obtain:

$$u_{t+1} = h(u_t) \leq (1 - h'(\tilde{u}))h(\tilde{u}) + h'(\tilde{u})u_t \leq \frac{3}{4}\tilde{u} + \frac{1}{4}u_t \leq \frac{3\tilde{u}}{4}\sum_{j=0}^{t-2}\frac{1}{4^j} + \frac{u_1}{4^{t-1}}$$

$$= \tilde{u}(1 - 1/4^{t-1}) + u_1(1/4^{t-1}) \leq 4\frac{c\beta D^2}{n} + \sqrt{\frac{cD^2\beta f^\star}{n}} + u_1(1/4^{t-1}).$$

We choose $T = \log_4(n) + 1$. This gives the error guarantee

$$\mathbb{E}[f(\hat{x}_T)] - f^\star \leq \frac{4c\beta D^2}{n} + \sqrt{\frac{cD^2\beta f^\star}{n}} + \frac{u_1}{n}.$$

We conclude by using the following fact

$$u_1 = \mathbb{E}f(\hat{x}_1) - f^\star \leq \langle \nabla f(x^\star), \hat{x}_1 - x^\star \rangle + \frac{\beta}{2}D^2 \leq D\sqrt{4\beta f^\star} + \frac{\beta}{2}D^2$$

to give the desired error rate.

Appendix B. Proof of Corollary 5

B.1. Supporting Lemmas

Lemma 10 *Let $X_1, \ldots, X_n$ be a martingale difference sequence adapted to the filtration $\mathcal{F}_t$ with $|X_t| \leq b$. Define $V := \sum_{t=1}^n \mathbb{E}[X_t^2|\mathcal{F}_{t-1}]$. For any $\delta < 1/e$ and $n \geq 3$*

$$\mathbb{P}\left[\sum_{t=1}^n X_t \geq \max\{2\sqrt{V}, 3b\sqrt{\log 1/\delta}\}\sqrt{\log 1/\delta}\right] \leq 4\delta \log n$$

Proof See Lemma 7 in [1]. □

Lemma 11 *Let $X_1, \ldots, X_n$ be independent, mean 0, random variables such that $|X_t| \leq b$ and $\sigma_t^2 := \mathbb{E}[X_t^2]$. For all any $0 < \delta < 1$, we have*

$$\mathbb{P}\left[\sum_{t=1}^n X_t \geq \max\left\{\sqrt{\frac{12}{5}\sum_{t=1}^n \sigma_t^2}, 4b\sqrt{\log 1/\delta}\right\}\sqrt{\log 1/\delta}\right] \leq \delta$$

Proof We know that X_t is $(\frac{6\sigma_t^2}{5}, 2b)$ sub-exponential. Using Corollary 3.17 from [9], we have that

$$\mathbb{P}\left[\sum_{t=1}^n X_t \geq t\right] \leq \exp\left(-\frac{1}{2}\min\left\{\frac{5t^2}{6\sum_{t=1}^n \sigma_t^2}, \frac{t}{2b}\right\}\right)$$

Setting the right hand side to δ and rearranging gives the desired result. □

Lemma 12 *Let F be β-smooth, non-negative, and bounded above by $B < \infty$. Let $\{\hat{x}_t\}_{t=1}^T$ be the iterates generated by running Algorithm 1 on Problem (1). For all $t \in [T]$, with probability at least $1 - \varepsilon$, we have*

$$f(\hat{x}_t) - f^\star \lesssim \log\left(\frac{\log n}{\varepsilon}\right)\left[\frac{1}{n} + \sqrt{\frac{f(\hat{x}_{t-1})}{n}}\right]$$

Proof For this proof, let $x_{t,i}$ (or x_i) denote the output of the ith iteration of the subalgorithm Algorithm 2 in epoch t of Algorithm 1. We begin with the result from Lemma 9 and plug in the step size schedule used by Algorithm 2. After applying the same arguments from (2) and (3), we obtain the following bound

$$\frac{1}{n}\sum_{i=1}^n f(x_i) - f^\star \leq \frac{4\beta D^2}{n} + \frac{3D\sqrt{2\beta}}{n}\sqrt{\sum_{k=1}^n F(\hat{x}_{t-1}; s_{t,i})} + \frac{1}{n}\sum_{i=1}^n \langle f'(x_i) - g_i, x_i - x^\star \rangle.$$

Bounding the second term We bound the $\sqrt{\sum_{k=1}^n F(\hat{x}_{t-1}; s_{t,i})}$ term first. Let $X_i := F(\hat{x}_{t-1}; s_{t,i}) - f(\hat{x}_{t-1})$. We have that $|X_i| \le B$ and $\sigma^2 = \mathbb{E}[X_i^2] \le 2Bf(\hat{x}_{t-1})$. Using Lemma 11, we have that with probability at least $1 - \varepsilon/2$

$$\sum_{i=1}^n X_i \le \sqrt{\frac{24}{5} Bnf(\hat{x}_{t-1}) \log(2/\varepsilon)} + 4B\log(2/\varepsilon).$$

In particular, we have that

$$\begin{aligned}
\frac{3D\sqrt{2\beta}}{n}\sqrt{\sum_{k=1}^n F(\hat{x}_{t-1}; s_{t,i})} &\le \frac{3D\sqrt{2\beta}}{\sqrt{n}}\sqrt{f(\hat{x}_{t-1})} + \frac{3D\sqrt{2\beta}}{n}\sqrt{\left[\sum_{k=1}^n F(\hat{x}_{t-1}; s_{t,i}) - f(\hat{x}_{t-1})\right]_+} \\
&\le \frac{3D\sqrt{2\beta}}{\sqrt{n}}\sqrt{f(\hat{x}_{t-1})} + \frac{3D\sqrt{2\beta}}{n^{3/4}}\left(\frac{24}{5}Bf(\hat{x}_{t-1})\log(2/\varepsilon)\right)^{1/4} \\
&\qquad + \frac{3D\sqrt{2\beta}}{n}\left(4B\log(2/\varepsilon)\right)^{1/2} \\
&\le 3D\sqrt{2\beta}\left[\left(1 + \frac{24B\log(2/\varepsilon)}{5}\right)\sqrt{\frac{f(\hat{x}_{t-1})}{n}} + \left(\sqrt{4B\log(2/\varepsilon)} + \frac{24B\log(2/\varepsilon)}{5}\right)\frac{1}{n}\right].
\end{aligned}$$

The first inequality comes from the fact that $\sqrt{x+y} \le \sqrt{x} + \sqrt{y}$ for non-negative x and y. The second inequality comes from our bound on $\sum_{i=1}^n X_i$. The third inequality comes from the fact that $\frac{f(\hat{x}_{t-1})^{1/4}}{n^{3/4}} \le \max(1/n, \sqrt{f(\hat{x}_{t-1})/n})$; we prove this fact by analyzing by cases on whether $f(\hat{x}_{t-1}) \le 1/n$ is true.

Bounding the third term Now we bound the last term. Let $Y_i := \langle f'(x_i) - g_i, x_i - x^\star\rangle$. We have that $|Y_i| \le \tilde{B}$. Using Lemma 7 and Cauchy-Schwartz we see that $\tilde{B} = 4D\sqrt{\beta B}$ suffices. Let $\mathcal{F}_i' := \sigma(\{S_{t,j}\}_{j\in[i]})$ be the σ-field generated by the first i iterations of Algorithm 2 of the tth epoch Algorithm 1; in particular, $x_i \in \mathcal{F}_{i-1}'$. We have $\sigma_i^2 = \mathbb{E}[Y_i^2|\mathcal{F}_{i-1}'] \le 16\beta D^2 f(x_t)$ using Lemma 7. Applying Lemma 10, we get that with probability at least $1 - \varepsilon/2$ (where $\varepsilon < 3\log(n)/e$),

$$\sum_{i=1}^n Y_i \le 2\sqrt{\log(8\log n/\varepsilon)}\sqrt{16\beta D^2 \sum_{i=1}^n f(x_i)} + 3\tilde{B}\log(8\log n/\varepsilon).$$

In particular we have

$$\frac{1}{n}\sum_{i=1}^n \langle f'(x_i) - g_i, x_i - x^\star\rangle \le \frac{2\sqrt{16\beta D^2\log(8\log n/\varepsilon)}}{\sqrt{n}}\sqrt{\frac{1}{n}\sum_{i=1}^n f(x_i)} + \frac{3\tilde{B}\log(8\log n/\varepsilon)}{n}$$

Combining terms Combining everything and using a union bound, we get that with probability at least $1 - \varepsilon$,

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} f(x_i) - f^\star &\le \left(4\beta D^2 + 3\tilde{B}\log(8\log n/\varepsilon) + \sqrt{18\beta D^2}\left(\sqrt{4B\log(2/\varepsilon)} + \frac{24B\log(2/\varepsilon)}{5}\right)\right)\frac{1}{n} \\ &+ \sqrt{18\beta D^2}\left(1 + \frac{24B\log(2/\varepsilon)}{5}\right)\sqrt{\frac{f(\hat{x}_{t-1})}{n}} \\ &+ \frac{2\sqrt{16\beta D^2\log(8\log n/\varepsilon)}}{\sqrt{n}}\sqrt{\frac{1}{n}\sum_{i=1}^{n} f(x_i) - f^\star + f^\star} \\ &=: \frac{A_1}{n} + A_2\sqrt{\frac{f(\hat{x}_{t-1})}{n}} + \frac{A_3}{\sqrt{n}}\sqrt{\frac{1}{n}\sum_{i=1}^{n} f(x_i) - f^\star + f^\star}\end{aligned}$$

for A_1, A_2, A_3 defined accordingly. We proceed by cases: either the left hand side is less than $\frac{A_1}{n} + A_2\sqrt{\frac{f(\hat{x}_{t-1})}{n}}$. Otherwise, if we let $Q := A_3/\sqrt{n}$, $C := A_1/n + A_2\sqrt{f(\hat{x}_{t-1})/n}$, and $H := \frac{1}{n}nf(x_i)$, rearranging, we get that the above inequality implies that $H - C \le Q\sqrt{H + f^\star}$. Because $H \ge C$ in this case, we can square both sides and use the quadratic formula, to get the following must hold

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} f(x_i) - f^\star &\le C + \frac{Q^2}{2} + \frac{1}{2}\sqrt{4CQ^2 + Q^4 + 4Q^2 f^\star} \\ &\le (A_1 + A_3^2 + A_3\sqrt{A_1} + A_3\sqrt{A_2})\frac{1}{n} + (A_2 + A_3 + A_3\sqrt{A_2})\sqrt{\frac{f(\hat{x}_{t-1})}{n}} \\ &\lesssim \frac{\log\left(\frac{\log n}{\varepsilon}\right)}{n} + \left(\log(1/\varepsilon) + \sqrt{\log\left(\frac{\log n}{\varepsilon}\right)\log(1/\varepsilon)}\right)\sqrt{\frac{f(\hat{x}_{t-1})}{n}}.\end{aligned}$$

The second inequality comes from an application of the fact $\sqrt{a+b} \le \sqrt{a} + \sqrt{b}$ for $a, b \ge 0$ and a reapplication of the fact that $\frac{f(\hat{x}_{t-1})^{1/4}}{n^{3/4}} \le \max(1/n, \sqrt{f(\hat{x}_{t-1})/n})$. Online-to-batch conversion (i.e., Jensen's inequality) gives the desired result. □

B.2. Proof of result

The proof structure mirrors Theorem 4 except we use Lemma 12 with ε/T as ε in place of Lemma 3. The use of ε/T ensures that the probability of failure any epoch is at most ε by union bound. If we let $u_t = f(\hat{x}_t) - f^\star$ and further set $T = \log_4(n) + 1$, we have with probability at least $1 - \varepsilon/T$.

$$f(\hat{x}_t) - f^\star \lesssim \log\left(\frac{\log n}{\varepsilon}\right)\left[\frac{1}{n} + \sqrt{\frac{f(\hat{x}_{t-1})}{n}}\right].$$

Using the previous notation, we have

$$u_{t+1} \lesssim h(u_t) := Q\sqrt{u_t + f^\star} + B$$

where $Q \coloneqq \log\left(\frac{\log n}{\varepsilon}\right)/\sqrt{n}$ and $B \coloneqq \log\left(\frac{\log n}{\varepsilon}\right)/n$. This corresponds to the following value of $u^\star$ and $\tilde{u}$

$$u^\star = \frac{2B + Q^2 + Q\sqrt{Q^2 + 4BQ^2 + 4f^\star}}{2} \lesssim \tilde{u} \coloneqq \frac{\log^2\left(\frac{\log n}{\varepsilon}\right)}{n} + \log\left(\frac{\log n}{\varepsilon}\right)\sqrt{\frac{1}{n}}$$

The remaining steps proceed in the same way to give the desired result.

Appendix C. Proof of Theorem 6

We let $\rho = \sqrt{\frac{c_0\beta\log(K\log n/\varepsilon)}{n}}$. We have by union bound that all K calls of ADASTAR succeed with probability at least $1-\varepsilon$. Throughout this proof, we will do our analysis on this event that all K calls of ADASTAR succeeds. From Theorem 5, we have that

$$f(x_k) - f^\star \leq \max\left(D_k^2\rho^2, D_k\rho\sqrt{f^\star}\right)$$

Without loss of generality, assume that f has κ-growth for some $\kappa > 2$. Due to growth, we have the following guarantee:

$$\begin{aligned}\frac{\lambda}{\kappa}\operatorname{dist}(x_k, \mathcal{X}^\star)^\kappa \leq f(x_k) - f^\star &\leq \max(D_k^2\rho^2, D_k\rho\sqrt{f^\star}) \\ \implies \operatorname{dist}(x_k, \mathcal{X}^\star) &\leq \left[\frac{\kappa}{\lambda}\max(D_k^2\rho^2, D_k\rho\sqrt{f^\star})\right]^{1/\kappa}.\end{aligned}$$

Colloquially, let k_0 be the last iterate where we can guarantee that there exists $x^\star \in \mathcal{X}^\star$ such that $x^\star \in \mathcal{X}_{k_0}$. Formally, k_0 is largest integer such that

$$D_{k_0+1} \geq \left[\frac{\kappa}{\lambda}\max(D_{k_0}^2\rho^2, D_{k_0}\rho\sqrt{f^\star})\right]^{1/\kappa}.$$

Phrased differently, if $\kappa > 2$, then it is the largest integer such that

$$D_{k_0} \geq \max\left(\left(2^\kappa\rho^2\kappa/\lambda\right)^{\frac{1}{\kappa-2}}, \left(2^\kappa\rho\sqrt{f^\star}\kappa/\lambda\right)^{\frac{1}{\kappa-1}}\right).$$

in particular, we have that

$$k_0 = \left\lfloor \log_2\left(\frac{D_0}{\max\left(\left(2^\kappa\rho^2\kappa/\lambda\right)^{\frac{1}{\kappa-2}}, \left(2^\kappa\rho\sqrt{f^\star}\kappa/\lambda\right)^{\frac{1}{\kappa-1}}\right)}\right)\right\rfloor + 1.$$

Because $\kappa \geq \underline{\kappa}$, we have that $K \geq k_0$. Thus, we have

$$f(x_K) - f^\star = f(x_{k_0+1}) - f^\star + \sum_{j=k_0+2}^{K}\left[f(x_j) - f(x_{j-1})\right]$$

The first term we bound as

$$f(x_{k_0+1}) - f^\star \leq \max\left(D_{k_0}^2\rho^2, D_{k_0}\rho\sqrt{f^\star}\right)$$

The second term we can bound as

$$\begin{aligned}\sum_{j=k_0+2}^{K}[f(x_j)-f(x_{j-1})] &\leq \sum_{j=k_0+2}^{K}[f(x_j)-\min_{x\in\mathcal{X}_j} f(x)] \leq \sum_{j=k_0+2}^{K}\max\left(D_j^2\rho^2, D_j\rho\sqrt{f^\star}\right)\\ &\leq \sum_{j=k_0+2}^{K}\max\left(\left(\frac{D_{k_0}}{2^{j-k_0}}\right)^2\rho^2, \left(\frac{D_{k_0}}{2^{j-k_0}}\right)\rho\sqrt{f^\star}\right)\\ &\leq \max\left(D_{k_0}^2\rho^2, D_{k_0}\rho\sqrt{f^\star}\right)\end{aligned}$$

where the last inequality comes from the fact that $\sum_{j=k_0+2} r^j \leq 1/(1-r)-1$ for $r<1$. Thus, we have that

$$f(x_K)-f^\star \leq 2\max\left(D_{k_0}^2\rho^2, D_{k_0}\rho\sqrt{f^\star}\right).$$

Note that by the definition of D_{k_0}, we have that

$$\frac{D_{k_0}}{2} = D_{k_0+1} \leq \max\left(\left(2^\kappa\rho^2\kappa/\lambda\right)^{\frac{1}{\kappa-2}}, \left(2^\kappa\rho\sqrt{f^\star}\kappa/\lambda\right)^{\frac{1}{\kappa-1}}\right).$$

Evaluating by using the fact that max is upper bounded by addition, and examining dominating terms (assuming that $f^\star$ is a constant gives the result.