Retrospective Approximation for Smooth Stochastic Optimization

author names withheld

Under Review for OPT 2020

Abstract

We consider stochastic optimization problems where a smooth (and potentially nonconvex) objective is to be minimized using a stochastic first-order oracle. These type of problems arise in many settings from simulation optimization to deep learning. We present Retrospective Approximation (RA) as a universal *sequential sample-average approximation* (SAA) paradigm where during each iteration k, a sample-path approximation problem is implicitly generated using an adapted sample size M_k, and solved (with prior solutions as "warm start") to an adapted error tolerance ϵ_k, using a "deterministic method" such as the line search quasi-Newton method. The principal advantage of RA is that decouples optimization from stochastic approximation, allowing the direct adoption of existing deterministic algorithms without modification, thus mitigating the need to redesign algorithms for the stochastic context. A second advantage is the obvious manner in which RA lends itself to parallelization. We identify conditions on $\{M_k, k \geq 1\}$ and $\{\epsilon_k, k \geq 1\}$ that ensure almost sure convergence and convergence in L_1-norm, along with optimal iteration and work complexity rates. We illustrate the performance of RA with line-search quasi-Newton on an ill-conditioned least squares problem, as well as an image classification problem using a deep convolutional neural net.

1. Introduction

We consider unconstrained smooth stochastic optimization problems of the form:

$$\text{min:} \;\; f(x) := \mathbb{E}[F(x, Y)] = \int F(x, Y)\, dP \qquad (Q)$$

where $F(\cdot, Y) : \mathbb{R}^d \times \mathcal{Y} \to \mathbb{R}$ and Y is a random variable with distribution P. In the standard context of parameter estimation, x represents the vector of model parameters to be fitted, Y represents the "random input-output data pairs," and $F(\cdot, \cdot)$ is a composition of the model and loss functions. The expected function f and its gradient, ∇f cannot be observed but can be estimated by making observations of the random function $F(\cdot, Y)$ and its derivative $\nabla F(\cdot, Y)$ at any given x. The random variable Y is "realized" either by drawing an observation from an existing dataset or by using Monte Carlo.

Stochastic Approximation (SA) [13], also known as Stochastic Gradient Descent (SGD), and its variants [5, 7, 8, 11, 12, 14], form the popular class of methods that are used for solving problems of the type (Q). Over the past few years, there has been an increased interest in stochastic quasi-Newton methods [1, 4, 6, 9, 10, 15–17] that incorporate curvature information within stochastic gradient based methods. While it has been observed that these methods are competitive [1, 3, 4, 6], they pose continuing challenges in terms of building and updating quasi-Newton matrices using observed stochastic gradients computed. In general, such methods entail a careful redesign of various components, e.g., line search and curvature updates, of classical deterministic nonlinear optimization algorithms, to make them suitable for stochastic settings. In this paper, we deviate from such attempts and instead consider an alternate approach called Retrospective Approximation (RA) that aims to strategically incorporate existing deterministic algorithms *as is*, that is, without any modifications, for use within a stochastic setting.

RA is a general purpose iterative "stochastic framework" that repeats the following three steps: (i) during *outer iteration* k, implicitly construct a sample-path approximation to the true problem using a sample size M_k that is adapted to past data represented by a filtration $\mathcal{F}_{k-1}$; (ii) using a weighted average of past solutions as "warm start," employ any *deterministic iterative solver* on the sample-path approximation to obtain a solution X_k to within a specified error tolerance ϵ_k, again adapted to past data represented by a filtration $\mathcal{F}_{k-1}$; (iii) update the weighted average of solutions obtained.

The main advantage of RA is that the framework's structure naturally decouples optimization from stochastic approximation. Such decoupling allows the use of any deterministic method for solving the "inner" approximate problems, thereby avoiding the need to redesign algorithms for the stochastic context and retaining the performance of established deterministic algorithms. Second, the decoupling forms the basis for a clear trade-off between computational effort and statistical accuracy, leading to theoretical guidance on the choice of sample sizes and error tolerances. Third, RA's structure lends itself to trivial parallelization where the computations required on samples M_k for solving the inner problem can be performed in parallel.

The difference in computation effort exerted by RA versus traditional stochastic gradient methods is rooted in what constitutes *an iteration*. Each iteration in a typical stochastic gradient method is computationally cheap in that it involves one gradient call after which the iterate is updated. By contrast, each (outer) iteration of RA can be more computationally expensive since it invokes the deterministic solver to identify an iterate satisfying the stipulated error tolerance. (The iterations performed by the deterministic solver will be called *inner iterations*.) Each outer iteration of RA is thus likely to be more productive than each iteration in stochastic gradient methods, but such gains come at an increased computational cost incurred during many inner iterations. A fair comparison of RA and SGD should thus involve a comparison of the total computational work as opposed to a measure such as iteration complexity.

RA's efficiency crucially depends on correctly balancing the sampling effort during each iteration with the stipulated error tolerance for the deterministic algorithm. Our analysis in this context leads to sufficient conditions that guarantee the consistency of RA's iterates, and the identification of the relationship between the adapted sample sizes and the error tolerances that result in optimal iteration and work complexity rates within RA. We are also able to establish central limit theorems, strong invariance principles, and stopping theorems, but we do not detail them here.

2. Retrospective Approximation

We now outline RA more precisely. Define the sample-path problem (S_m) having sample size m.

$$\text{minimize: } f_m(x) := m^{-1} \sum_{j=1}^{m} F(x, Y_j) = \int F(x, Y)\, dP_m, \quad x \in \mathbb{R}^d. \qquad (S_m)$$

The random function $f_m(\cdot)$ is said to be a sample-path approximation to $f(\cdot)$ constructed with sample size m, and P_m is the empirical measure associated with iid copies of Y. We define the corresponding sample-path derivative function $\nabla f_m(\cdot)$ in an analogous manner.

We assume that we have at our disposal a method to *globally solve* (S_m) to any specified accuracy $\epsilon > 0$. In the smooth (potentially nonconvex) case, this means we have a deterministic solver capable of identifying a point, say $X_m(\epsilon) \in \mathbb{R}^d$, that satisfies $\|\nabla f_m(X_m(\epsilon))\| \leq \epsilon$.

The RA algorithm framework, presented in Algorithm 1 is an iterative framework that is organized into outer and inner iterations. Assuming the existence of a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_k\}_{k\geq 1}, P)$, during the k-th outer iteration, RA uses a specified deterministic solver to solve the sample-path problem $(S_{M_k}), M_k \in \mathcal{F}_{k-1}$ to within accuracy $\epsilon_k \in \mathcal{F}_{k-1}$. The solution X_k obtained at the end of the k-th outer iteration is appropriately averaged with past solutions, and the resulting average $\bar{X}_k$ is used as a "warm start" (or "initial guess") when solving the $(k+1)$-th sample-path problem $(S_{M_{k+1}})$.

```
Algorithm 1: Retrospective Approximation
input : (i) initial guess $X_0$; (ii) procedure to update sample sizes $\{M_k, k \geq 1\}$; (iii) procedure
        to update error tolerances $\{\epsilon_k, k \geq 1\}$; (v) procedure to update weights $\{W_k, k \geq 1\}$;
        (iv) solver-S, e.g., line search L-BFGS.
for $k = 1, 2, ...$ do
    1. Set sample size and error tolerance: Choose $M_k \in \mathcal{F}_{k-1}$ and $\epsilon_k \in \mathcal{F}_{k-1}$.
    2. Solve $k$-th sample-path problem to accuracy $\epsilon_k$: with Solver-S and $\bar{X}_{k-1}$ as
     "warm-start," obtain $X_k$ satisfying $\|\nabla f_{M_k}(X_k)\| \leq \epsilon_k$.
    3. Update solution: Compute $\bar{X}_k := \sum_{j=1}^{k} W_j X_j / \sum_{j=1}^{k} W_j$.
end
```

3. Main Results

In this section, we present our main results which demonstrate consistency and characterize the associated convergence rates of the RA algorithm. For brevity, we defer a number of other results to a later more detailed manuscript. These include statements about central limit theorem, strong invariance law, sequential stopping, the effect of averaging, and "warm starts."

3.1. Assumptions

In what follows, we list four standing assumptions which touch upon the structure of the random function $F(\cdot, Y)$, the finiteness of second moment of the estimator, the growth rate of sample path functions and conditions on the sample size and error tolerance. For convenience, we define the *attractor set* as the set of critical points of f: $\mathcal{X}^* := \{x \in \mathbb{R}^d : \nabla f(x) = 0\}$.

Assumption 1 (Structure of the Integrand) *The random function $F(\cdot, Y) : \mathbb{R}^d \to \mathbb{R}$ is $L(Y)$-smooth, that is, $\|\nabla F(x, Y) - \nabla F(y, Y)\| \leq L(Y)\|x - y\|, \forall x, y \in \mathbb{R}^d$, where $\mathbb{E}[L^2(Y)] < \infty$.*

Assumption 2 (Finite Variance) *Recall the sample-path gradient $\nabla f_{M_k}(x) := M_k^{-1} \sum_{j=1}^{M_k} \nabla F(x, Y_j)$, $x \in \mathbb{R}^d$, where $M_k \in \mathcal{F}_{k-1}$. The variance of the sample-path gradient over the attractor set $\mathcal{X}^*$ is bounded, that is, there exists σ^2 such that* $\sup_{x\in\mathcal{X}^*} \mathbb{E}[\|\nabla f_{M_k}(x)\|^2 \mid \mathcal{F}_{k-1}] \leq \sigma^2 M_k^{-1}$ a.s.

Assumption 3 (Sample-path Growth Condition) *Define the sample-path growth rate*

$$\Lambda_{M_k} := \inf\{\lambda : \|\nabla f_{M_k}(x) - \nabla f_{M_k}(x^*)\| \geq \lambda \|x - x^*\| \quad \forall x^* \in \mathcal{X}^*\}.$$

There exists $\Lambda > 0$ *such that* $\mathbb{E}[\Lambda_{M_k}^{-2} \,|\, \mathcal{F}_{k-1}] \leq \Lambda^{-2}$ a.s.

Assumption 4 (Sample Sizes and Error Tolerances) *The sample size* M_k *and the error tolerance* ϵ_k *used within the* k*-th iteration of RA are adapted to the filtration* $\mathcal{F}_{k-1}$*, and satisfy, with probability one,* $\sum_{k=1}^{\infty} M_k{}^{-1/2} < \infty$ *and* $\sum_{k=1}^{\infty} \left(\mathbb{E}[\epsilon_k^2 \,|\, \mathcal{F}_{k-1}]\right)^{1/2} < \infty$.

3.2. Main Theorems

We start with a fundamental theorem which asserts that RA's iterates almost surely get "trapped" within a fixed bounded region (not depending on ω) after a large enough number of iterations.

Theorem 1 *Suppose Assumptions 1, 2, 3, and 4 hold. Then, given any* $\epsilon > 0$*, the sequence* $\{X_k, k \geq 1\}$ *satisfies, for* $k \geq K(\epsilon)$, $X_k \in \mathcal{H}(\epsilon)$ a.s., *where* $\mathcal{H}(\epsilon) := \{x : \|\nabla f(x)\| \leq \epsilon\}$.

A corollary of Theorem 1 is the strong consistency of RA's iterates.

Theorem 2 (Almost Sure Consistency and L_1 convergence of RA) *Let the postulates of Theorem 1 hold. Then the iterates* $\{X_k, k \geq 1\}$ *generated by RA satisfy, as* $k \to \infty$,

$$\|\nabla f(X_k)\| \to 0 \text{ a.s.}; \quad \mathbb{E}[\|\nabla f(X_k)\|] \to 0.$$

Furthermore, as $k \to \infty$, $\operatorname{dist}(X_k, \mathcal{X}^*) := \inf\{\|x^* - X_k\| : x^* \in \mathcal{X}^*\} \to 0$ a.s.

The following theorem gives a non-asymptotic rate result in the L_1 norm.

Theorem 3 (Non-Asymptotic Rate in L_1) *Suppose Assumptions 1, 2 and 3 hold. Furthermore, let* $M_k := C_{1,k} M_{k-1}$ *for* $k \geq 2$ *where* $C_{1,k} \in \mathcal{F}_{k-1}$ *such that* $C_{1,k} \in [c_1, \bar{c}_1]$*, with* $1 < c_1 \leq \bar{c}_1 < \infty$ *and* $M_1 = m_1$*. Also, suppose* $\epsilon_k := C_{2,k} M_k^{-1/2}$ *for* $k \geq 2$ *where* $C_{2,k} \in \mathcal{F}_{k-1}$ *such that* $C_{2,k} \in [\underline{c}_2, c_2]$*, with* $0 < \underline{c}_2 \leq c_2 < \infty$*. Then,*

$$\mathbb{E}\left[\|\nabla f(X_k)\|\right] \leq \left(\frac{1}{\sqrt{c_1}}\right)^{k-1} \left(\frac{\mathbb{E}[L](c_2 + \sigma)}{\Lambda m_1}\right).$$

Considering brevity, we have omitted a number of other results including: (i) an optimal work complexity result; (ii) the effect of "warm starts" and "iterate averaging"; (iii) a strong invariance law on the empirical process $\{\sqrt{M_k}(\nabla f_{M_k}(x) - \nabla f(x)) : x \in \mathcal{H}(\epsilon_0)\}$; (iv) a central limit theorem on the sequence $\{\|\nabla f(X_k)\|, k \geq 1\}$ of true gradient norms at RA's iterates; and (v) a sequential stopping theorem that uses (iii) and (iv) to construct a sequential confidence interval on $\|\nabla f(X_k)\|$.

4. Experiments

To investigate the performance of RA, we ran a setting of RA alongside Adam and SGD (two highly popular stochastic gradient methods) in the context of least-squares and image classification. Both experiments were implemented in Python. The Tensorflow library was used in constructing models and computing gradients. We used Tensorflow's implementations of Adam and SGD, using the default settings (though we did explore various choices of step size in the first experiment). For brevity, complete implementation details are provided in the supplementary materials.

4.1. Poorly-Conditioned Least-Squares

Our first experiment consists of a quadratic (least-squares) minimization problem of dimension $1,000$ using a simulated dataset, where the condition number of the observed covariance matrix is approximately 10^6. We ran RA alongside SGD, where SGD was run with various choices of (constant) step size in negative powers of 10. Results from this experiment are displayed in the top row of figure 1. Notably, SGD's performance is highly dependent on the choice of step size with 10^{-6} exhibiting the best performance. RA, on the other hand, descends and appears to converge, importantly without requiring any hyper-parameter tuning.

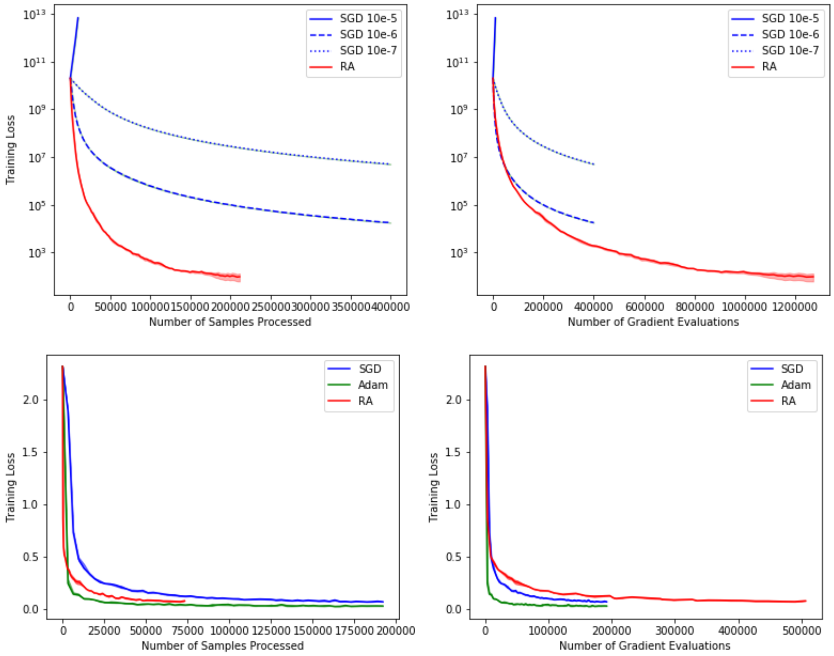

Figure 1: **Numerical Experiments**. The top row displays results from the least-squares experiment; the bottom row displays results from fitting the LeNet model using the MNIST dataset. Training loss is shown as a function of cumulative oracle work (left column) and as a function of cumulative gradient evaluations (right column). The paths represent median loss, while the shaded regions represent the inter-quartile range.

4.2. LeNet on MNIST

For our second experiment, we used a variant of the LeNet Convolutional Neural Network applied to the MNIST dataset. In the bottom row of Figure 1, we see that RA and Adam show similar performance in terms of oracle work (SGD shows worse performance), but RA lags behind Adam and SGD in terms of the total number of gradient computations. Thus RA may be most advantageous when computation is cheap relative to sampling, as may be the case when parallel architecture is available.

References

[1] A. S. Berahas, J. Nocedal, and M. Takác. A multi-batch l-bfgs method for machine learning. In *Advances in Neural Information Processing Systems*, pages 1055–1063, 2016.

[2] Patrick Billingsley. *Probability and Measure*. John Wiley and Sons, third edition, 1995.

[3] Raghu Bollapragada, Jorge Nocedal, Dheevatsa Mudigere, Hao-Jun Shi, and Ping Tak Peter Tang. A progressive batching l-BFGS method for machine learning. volume 80 of *Proceedings of Machine Learning Research*, pages 620–629. PMLR, 2018.

[4] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer. A stochastic quasi-Newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016.

[5] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *NIPS*, 2014.

[6] M. P. Friedlander and M. Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 34(3):A1380–A1405, 2012.

[7] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *NIPS*, 2013.

[8] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] A. Mokhtari and A. Ribeiro. Global convergence of online limited memory bfgs. *Journal of Machine Learning Research*, 16(1):3151–3181, 2015.

[10] Philipp Moritz, Robert Nishihara, and Michael I. Jordan. A linearly-convergent stochastic l-bfgs algorithm. *arXiv:1508.02087*, 2016.

[11] B. T. Polyak. Some methods of speeding up the convergnce of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[12] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal of Control and Optimization*, 30(4):838–855, 1992.

[13] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[14] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *arXiv:1309.2388*, 2016.

[15] N. N. Schraudolph, J. Yu, and S. Günter. A stochastic quasi-newton method for online convex optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 436–443, 2007.

[16] J. Sohl-Dickstein, B. Poole, and S. Ganguli. Fast large-scale optimization by unifying stochastic gradient and quasi-Newton methods. In *International Conference on Machine Learning*, pages 604–612, 2014.

[17] C. Zhou, W. Gao, and D. Goldfarb. Stochastic adaptive quasi-Newton methods for minimizing expected values. In *International Conference on Machine Learning*, pages 4150–4159, 2017.

5. Supplementary Materials

5.1. Proofs of Theorems 1, 2, and 3

5.1.1. PROOF OF THEOREM 1

Before proving Theorem 1, we state a necessary lemma that follows from Assumption 1.

Lemma 4 (Smooth Objective) *Suppose Assumption 1 holds. Then the function* $f(x) = \mathbb{E}[F(x, Y)]$, $x \in \mathbb{R}^d$ *is* $\mathbb{E}[L]$*-smooth, that is,*

$$\|\nabla f(x) - \nabla f(y)\| \leq \mathbb{E}[L(Y)]\|x - y\|, \quad x, y \in \mathbb{R}^d.$$

We now move to theorem 1.

Theorem 1 *Suppose Assumptions 1, 2, 3, and 4 hold. Then, given any* $\epsilon > 0$*, the sequence* $\{X_k, k \geq 1\}$ *satisfies, for* $k \geq K(\epsilon)$*,* $X_k \in \mathcal{H}(\epsilon)$ a.s., *where* $\mathcal{H}(\epsilon) := \{x : \|\nabla f(x)\| \leq \epsilon\}$.

Proof Fix a point $x^* \in \mathcal{X}^*$ and recall that $M_k \in \mathcal{F}_{k-1}$. Observe that for any $t > 0$,

$$\begin{aligned}
\sum_{k=1}^{\infty} \mathbb{P}\left(\|\nabla f(X_k)\| > t \,|\, \mathcal{F}_{k-1}\right) &\leq \sum_{k=1}^{\infty} \frac{1}{t}\mathbb{E}\left[\|\nabla f(X_k)\| \,|\, \mathcal{F}_{k-1}\right] \\
&\leq \sum_{k=1}^{\infty} \frac{1}{t}\mathbb{E}\left[\mathbb{E}[L]\,\|X_k - x^*\| \,|\, \mathcal{F}_{k-1}\right] \\
&\leq \frac{\mathbb{E}[L]}{t}\sum_{k=1}^{\infty} \mathbb{E}\left[\Lambda_{M_k}^{-1}\|\nabla f_{M_k}(X_k) - \nabla f_{M_k}(x^*)\| \,|\, \mathcal{F}_{k-1}\right] \\
&\leq \frac{\mathbb{E}[L]}{t}\sum_{k=1}^{\infty} \mathbb{E}\left[\Lambda_{M_k}^{-1}\left(\|\nabla f_{M_k}(x^*)\| + \epsilon_k\right) \,|\, \mathcal{F}_{k-1}\right] \\
&\leq \frac{\mathbb{E}[L]\sqrt{2}}{t}\sum_{k=1}^{\infty} \left(\mathbb{E}\left[\Lambda_{M_k}^{-2} \,|\, \mathcal{F}_{k-1}\right]\right)^{1/2}\left(\mathbb{E}\left[\|\nabla f_{M_k}(x^*)\|^2 + \epsilon_k^2 \,|\, \mathcal{F}_{k-1}\right]\right)^{1/2} \\
&\leq \frac{\mathbb{E}[L]\sqrt{2}}{t\Lambda}\sum_{k=1}^{\infty} \left(\mathbb{E}\left[\|\nabla f_{M_k}(x^*)\|^2 + \epsilon_k^2 \,|\, \mathcal{F}_{k-1}\right]\right)^{1/2} \\
&\leq \frac{\mathbb{E}[L]\sqrt{2}}{t\Lambda}\sum_{k=1}^{\infty} \left(\frac{\sigma^2}{M_k} + \mathbb{E}\left[\epsilon_k^2 \,|\, \mathcal{F}_{k-1}\right]\right)^{1/2} \\
&\leq \frac{\mathbb{E}[L]\sqrt{2}}{t\Lambda}\left(\sum_{k=1}^{\infty} \frac{\sigma}{\sqrt{M_k}} + \sum_{k=1}^{\infty} \left(\mathbb{E}[\epsilon_k^2 \,|\, \mathcal{F}_{k-1}]\right)^{1/2}\right) \\
&< \infty \quad \text{a.s.,}
\end{aligned} \tag{1}$$

where the first inequality in (1) is due to Markov [2], the second follows since the function f is $\mathbb{E}[L]$-smooth by Lemma 4, the third due to the definition of the sample-path growth-rate in Assumption 3, the fourth inequality due to the definition of X_k, the fifth due to the Cauchy-Schwarz [2] inequality, the sixth due to applying the minimum sample-path growth assumption in Assumption 3, the seventh due to the estimator assumption in Assumption 2, the eighth holds since $(a+b)^{1/2} \leq a^{1/2} + b^{1/2}$ for non-negative a, b, and the last due to Assumption 4. Conclude from (1) and the filtered Borel-Cantelli's lemma [2] that for any t,

$$\mathbb{P}\left(\|\nabla f(X_k)\| > t \text{ i.o.}\right) = 0, \tag{2}$$

implying in turn that

$$\mathbb{P}\left(X_k \notin \mathcal{H}(\epsilon) \text{ i.o.}\right) = 0, \tag{3}$$

thus proving the assertion of the theorem. ■

5.1.2. PROOF OF THEOREM 2

Theorem 2 (Almost Sure Consistency and L_1 convergence of RA) *Let the postulates of Theorem 1 hold. Then the iterates $\{X_k, k \geq 1\}$ generated by RA satisfy, as $k \to \infty$,*

$$\|\nabla f(X_k)\| \to 0 \text{ a.s.}; \quad \mathbb{E}[\|\nabla f(X_k)\|] \to 0.$$

Furthermore, the iterates $\{X_k, k \geq 1\}$ also satisfy, as $k \to \infty$,

$$\text{dist}(X_k, \mathcal{X}^*) \to 0 \text{ a.s.}$$

.

Proof Since Theorem 1 holds for arbitrary $\epsilon > 0$, $\|\nabla f(X_k)\| \to 0$ a.s. holds trivially as a consequence. Also, $\mathbb{E}[\|\nabla f(X_k)\|] \to 0$ holds from the assertion $\|\nabla f(X_k)\| \to 0$ a.s. and the uniform integrability of $\{\nabla f(X_k), k \geq 1\}$ evident from the bound on the tail probability of $\nabla f(X_k)$ in the proof of Theorem 1. Finally, the assertion dist$(X_k, \mathcal{X}^*) \to 0$ a.s. holds as well due to the definition of the set $\mathcal{X}^*$. ■

5.1.3. PROOF OF THEOREM 3

Theorem 3 (Non-Asymptotic Rate in L_1) *Suppose Assumptions 1, 2 and 3 hold. Let $M_k := C_{1,k} M_{k-1}$ for $k \geq 2$ where $C_{1,k} \in \mathcal{F}_{k-1}$ such that $C_{1,k} \in [c_1, \bar{c}_1]$, with $1 < c_1 \leq \bar{c}_1 < \infty$ and $M_1 = m_1$. Also, suppose $\epsilon_k := C_{2,k} M_k^{-1/2}$ for $k \geq 2$ where $C_{2,k} \in \mathcal{F}_{k-1}$ such that $C_{2,k} \in [\underline{c}_2, c_2]$, with $0 < \underline{c}_2 \leq c_2 < \infty$. Then,*

$$\mathbb{E}\left[\|\nabla f(X_k)\|\right] \leq \left(\frac{1}{\sqrt{c_1}}\right)^{k-1} \left(\frac{\mathbb{E}[L](c_2 + \sigma)}{\Lambda m_1}\right).$$

Proof Consider any point $x^* \in \mathcal{X}^*$, we have that

$$
\begin{aligned}
\mathbb{E}\left[\|\nabla f(X_k)\|\right] &\leq \mathbb{E}[L]\mathbb{E}\left[\|X_k - x^*\|\right] \\
&\leq \mathbb{E}[L]\mathbb{E}\left[\frac{\|\nabla f_{M_k}(X_k) - \nabla f_{M_k}(x^*)\|}{\Lambda_{M_k}}\right] \\
&\leq \mathbb{E}[L]\mathbb{E}\left[\frac{\|\nabla f_{M_k}(X_k)\|}{\Lambda_{M_k}} + \frac{\|\nabla f_{M_k}(x^*)\|}{\Lambda_{M_k}}\right] \\
&= \mathbb{E}[L]\left(\mathbb{E}\left[\mathbb{E}\left[\frac{\|\nabla f_{M_k}(X_k)\|}{\Lambda_{M_k}} \,|\, \mathcal{F}_{k-1}\right]\right] + \mathbb{E}\left[\mathbb{E}\left[\frac{\|\nabla f_{M_k}(x^*)\|}{\Lambda_{M_k}} \,|\, \mathcal{F}_{k-1}\right]\right]\right) \\
&\leq \mathbb{E}[L]\mathbb{E}\left[\mathbb{E}\left[\Lambda_{M_k}^{-2} \,|\, \mathcal{F}_{k-1}\right]^{1/2} \mathbb{E}\left[\|\nabla f_{M_k}(X_k)\|^2 \,|\, \mathcal{F}_{k-1}\right]^{1/2}\right] \\
&\quad + \mathbb{E}[L]\mathbb{E}\left[\mathbb{E}\left[\Lambda_{M_k}^{-2} \,|\, \mathcal{F}_{k-1}\right]^{1/2} \mathbb{E}\left[\|\nabla f_{M_k}(x^*)\|^2 \,|\, \mathcal{F}_{k-1}\right]^{1/2}\right] \\
&\leq \mathbb{E}[L]\Lambda^{-1}\left(\mathbb{E}\left[\mathbb{E}\left[\epsilon_k^2 \,|\, \mathcal{F}_{k-1}\right]^{1/2}\right] + \mathbb{E}\left[\mathbb{E}\left[\|\nabla f_{M_k}(x^*)\|^2 \,|\, \mathcal{F}_{k-1}\right]^{1/2}\right]\right) \\
&\leq \mathbb{E}[L]\Lambda^{-1}\left(\mathbb{E}\left[\frac{c_2}{\sqrt{M_k}} + \frac{\sigma}{\sqrt{M_k}}\right]\right) \\
&\leq \left(\frac{1}{\sqrt{c_1}}\right)^{k-1}\left(\frac{\mathbb{E}[L](c_2+\sigma)}{\Lambda\sqrt{m_1}}\right),
\end{aligned}
$$

where the first inequality is due to Lemma 4, the second inequality is due to Assumption 3, the third inequality is due to the fact that $\|a - b\| \leq \|a\| + \|b\|$, the fifth inequality follows from Cauchy-Schwarz [2], the sixth inequality follows from $\|\nabla f_{M_K}(X_K)\| \leq \epsilon_k$ and Assumption 3, the seventh inequality follows since $\epsilon_k \leq c_2/\sqrt{M_k}$ and Assumption 2, and the last inequality is due to the fact that $M_k \geq c_1 M_{k-1}$. ■

5.2. Full Implementations Details of Experiments

5.2.1. RA SPECIFICS

For the least-squares and logistic regression experiments, we used the following sampling schedule for RA: $q_k := 1 + 7k^{-1.7}$, $M_k := q_k M_{k-1}$, and $M_1 = 2$. The multiplicative factor q_k for the sample size was defined so that M_k increases rapidly at first, but then slows as the sample sizes become large. Due to the high variance in sample path objective functions for CNNs when using a small batch size, we set $M_1 := 100$ for the two CNN experiments (while adjusting q_k accordingly).

For the deterministic solver, L-BFGS with backtracking line search was used across all experiments. Importantly, the stored gradient and iterate differences used in L-BFGS were carried across outer iterations (instead of restarting L-BFGS from scratch for every new sample-path problem).

5.2.2. DEFINING ϵ_k

To define the threshold ϵ_k for terminating L-BFGS during the k^{th} iteration, let S_k be the set of M_k of iid copies of Y queried by the oracle at iteration k. Furthermore, let S_k^σ be a random subset of S_k

with size $m_\sigma < M_k$. We estimate the variance of the gradient norm at x_k as

$$\hat{\sigma}_k^2 := \frac{1}{m_\sigma - 1} \sum_{Y_i \in S_k^\sigma} \Big(\|G(x_{k-1}, Y_i)\| - \overline{\|G(x_{k-1}, Y_i)\|} \Big)^2.$$

and set

$$\epsilon_k := \frac{\hat{\sigma}_k}{\sqrt{M_k}}.$$

Although the computational burden of calculating $\hat{\sigma}_k^2$ is relatively low for small m_σ, we can further reduce its cost by only re-computing $\hat{\sigma}_k^2$ every m iterations.

5.2.3. IMPLEMENTATION OF ADAM AND SGD

In the first experiment, we ran SGD with various choices of step-sizes. This was done to investigate the sensitivity of step size on the algorithms' performance. For the remaining experiments we used the Tensorflow default (constant) step sizes for SGD and Adam of 0.01 and 0.001, respectively (and the same for Adam's other hyper-parameters). In each plot, the paths represent median values of the vertical axis variable across 3 runs, and the shaded regions represent the interquartile range. For each experiment, all replications of each algorithm begin from the same initial solution.

5.2.4. POORLY-CONDITIONED LEAST-SQUARES

The first experiment consists of a standard least-squares minimization on a simulated dataset:

$$\underset{\beta \in \mathbb{R}^p}{\text{minimize}}\ f(\beta) := \mathbb{E}[||Y - X^T\beta||^2]$$

where $X_i \sim N(0, I_p)$, $Y_i | X_i \sim N(X_i^T \beta, I_p)$, and the true solution β is set as $\beta := (1, 2, ..., p)^T$. For this problem we set $p := 1,000$ and $N = 30,000$. The condition number of the observed matrix $n^{-1}X^TX$ was approximately 10^6.

5.2.5. LENET ON MNIST

For our second experiment we use a variant of the LeNet CNN applied to the MNIST dataset. This neural net has 2 convolutional layers with max pooling followed by a fully-connected layer. The convolutional layers have 5x5 kernels with 20 and 50 output channels respectively. The fully connected layer has 500 neurons with ReLU activations. Parameters are initialized using Kaiming Uniform initializion.