Universality of AdaGrad Stepsizes for Stochastic Optimization: Inexact Oracle, Acceleration and Variance Reduction

Anton Rodomanov[1] Xiaowen Jiang[1,2] Sebastian Stich[1]

[1]CISPA Helmholtz Center for Information Security, Germany [2]Saarland University, Germany

Paper (PDF)

Problem Formulation

Consider the composite optimization problem:

$$F^* := \min_{x \in \operatorname{dom}\psi} \big[F(x) := f(x) + \psi(x)\big], \tag{P}$$

where $f\colon \mathbb{R}^d \to \mathbb{R}$ and $\psi\colon \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ are convex, ψ is simple.

Assumptions: ($\|\cdot\|$ is a Euclidean norm, $\|\cdot\|_*$ is its dual)

1. f is (δ_f, L_f)-approximately smooth with components $(\bar{f}, \bar{g})$.
2. Stochastic oracle (SO) $\hat{g} = g(x,\xi)$: $\mathbb{E}_\xi[g(x,\xi)] = \bar{g}(x)$.
3. Either $\sigma^2 := \sup_{x \in \operatorname{dom}\psi} \operatorname{Var}_{\hat{g}}(x) < \infty$, or the variance of $\hat{g}$ is $(\delta_{\hat{g}}, L_{\hat{g}})$-approximately smooth w.r.t. f and $\sigma_*^2 := \operatorname{Var}_{\hat{g}}(x^*) < \infty$ (where $\operatorname{Var}_{\hat{g}}(x) := \mathbb{E}_\xi[\|g(x,\xi) - \bar{g}(x)\|_*^2]$).

Goal: Develop methods for (P) without knowing any of these parameters.

We do so assuming additionally $\operatorname{dom}\psi$ is bounded with known diameter:

4. Bounded domain: $\|x - y\| \le D$, $\forall x, y \in \operatorname{dom}\psi$.

Note: Asm. 4 can always be ensured with $D = 2R_0$ whenever we know $R_0 \ge \|x_0 - x^*\|$ by restricting ψ onto the R_0-ball around x_0.

Approximate Lipschitz Smoothness of Function

Definition (Devolder et al. 2013): f is called (δ_f, L_f)-approximately smooth with components $(\bar{f}, \bar{g})$ if, for any $x, y \in \mathbb{R}^d$,

$$0 \le \big[\beta_{f,\bar{f},\bar{g}}(x,y) := f(y) - \bar{f}(x) - \langle \bar{g}(x), y - x\rangle\big] \le \frac{L_f}{2}\|y - x\|^2 + \delta_f.$$

Examples:

- f is L-smooth $\iff$ $(\bar{f}, \bar{g}) = (f, \nabla f)$ with $L_f = L$, $\delta_f = 0$
- f is (ν, H_ν)-Hölder smooth ($\|\nabla f(x) - \nabla f(y)\|_* \le H_f(\nu)\|x - y\|^\nu$, $\forall x, y$) $\implies$ $(\bar{f}, \bar{g}) = (f, \nabla f)$ with $L_f = \big[\frac{1-\nu}{2(1+\nu)\delta_f}\big]^{\frac{1-\nu}{1+\nu}}[H_f(\nu)]^{\frac{2}{1+\nu}}$ and any $\delta_f > 0$.
- $\phi(x) \le f(x) \le \phi(x) + \delta$, $\forall x$, with L-smooth ϕ $\implies$ $(\bar{f}, \bar{g}) = (\phi, \nabla\phi)$ with $L_f = L$, $\delta_f = \delta$.
- $f(x) = \max_u \Psi(x,u)$ with str. concave Ψ, $\bar{u}(x) \approx_\delta \operatorname{argmax}_u \Psi(x,u)$ $\implies$ $\bar{f}(x) = \Psi(x, \bar{u}(x))$, $\bar{g}(x) = \nabla_u \Psi(x, \bar{u}(x))$ with $\delta_f = \delta$.

Approximate Lipschitz Smoothness of Variance

Definition (new): Variance of $\hat{g}$ is $(\delta_{\hat{g}}, L_{\hat{g}})$-approximately smooth w.r.t. f:

$$\mathbb{E}_\xi[\|[g(x,\xi) - g(y,\xi)] - [\bar{g}(x) - \bar{g}(y)]\|_*^2] \le 2L_{\hat{g}}[\beta_{f,\bar{f},\bar{g}}(x,y) + \delta_{\hat{g}}].$$

c.f.: $\|\nabla f(x) - \nabla f(y)\|_*^2 \le 2L[f(y) - f(x) - \langle \nabla f(x), y - x\rangle]$.

Main example: $f(x) = \mathbb{E}_\xi[f_\xi(x)]$, where each f_ξ is convex and (δ_ξ, L_ξ)-approx. smooth with components $(\bar{f}_\xi, \bar{g}_\xi)$. Then, $g(x,\xi) = \bar{g}_\xi(x)$ satisfies the variance condition with $\bar{f}(x) = \mathbb{E}_\xi[\bar{f}_\xi(x)]$, $\bar{g}(x) = \mathbb{E}_\xi[\bar{g}_\xi(x)]$, and $L_{\hat{g}} = L_{\max}$, $\delta_{\hat{g}} = \mathbb{E}_\xi[\delta_\xi]$, where $L_{\max} := \sup_\xi L_\xi$.

Another example: σ^2-bounded variance $\implies$ $L_{\hat{g}} = \frac{2\sigma^2}{\delta_{\hat{g}}}$ for any $\delta_{\hat{g}} > 0$.

Note: If $\hat{g}_b$ is the mini-batch version of $\hat{g}$ of size b, then $L_{\hat{g}_b} = \frac{1}{b}L_{\hat{g}}$, $\delta_{\hat{g}_b} = \delta_{\hat{g}}$.

Universal SGD

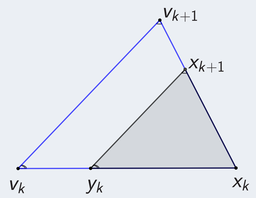
Algorithm $\operatorname{UniSgd}_{\hat{g},\psi}(x_0, N; D, M_0 = 0)$

$g_0 \cong \hat{g}(x_0)$.
for $k = 0, \dots, N-1$ **do**
  $x_{k+1} = \operatorname{Prox}_\psi(x_k, g_k, M_k)$, $g_{k+1} \cong \hat{g}(x_{k+1})$.
  $M_{k+1} = \sqrt{M_k^2 + \frac{1}{D^2}\|g_{k+1} - g_k\|_*^2}$.
return $(\bar{x}_N, x_N, M_N)$, where $\bar{x}_N := \frac{1}{N}\sum_{i=1}^N x_i$.

Universal Fast SGD

Algorithm $\operatorname{UniFastSgd}_{\hat{g},\psi}(x_0; D)$

$v_0 = x_0$, $M_0 = A_0 = 0$.
for $k = 0, 1, \dots$ **do**
  $a_{k+1} = \frac{1}{2}(k+1)$, $A_{k+1} = A_k + a_{k+1}$.
  $y_k = \frac{A_k}{A_{k+1}}x_k + \frac{a_{k+1}}{A_{k+1}}v_k$, $g_{y_k} \cong \hat{g}(y_k)$.
  $v_{k+1} = \operatorname{Prox}_\psi(v_k, g_{y_k}, \frac{M_k}{a_{k+1}})$.
  $x_{k+1} = \frac{A_k}{A_{k+1}}x_k + \frac{a_{k+1}}{A_{k+1}}v_{k+1}$, $g_{x_{k+1}} \cong \hat{g}(x_{k+1})$.
  $M_{k+1} = \sqrt{M_k^2 + \frac{a_{k+1}^2}{D^2}\|g_{x_{k+1}} - g_{y_k}\|_*^2}$.



Convergence Rates

Method	Convergence rate	SO complexity
UniSgd	$\frac{L_f D^2}{k} + \delta_f + \min\big\{\frac{\sigma D}{\sqrt{k}}, \frac{\sigma_* D}{\sqrt{k}} + \frac{L_{\hat{g}} D^2}{k} + \delta_{\hat{g}}\big\}$	k
UniFastSgd	$\frac{L_f D^2}{k^2} + k\delta_f + \min\big\{\frac{\sigma D}{\sqrt{k}}, \frac{\sigma_* D}{\sqrt{k}} + \frac{L_{\hat{g}} D^2}{k} + \delta_{\hat{g}}\big\}$	k
UniSvrg	$\frac{(L_f + L_{\hat{g}})D^2}{2^t} + \delta_f + \delta_{\hat{g}}$	$2^t + n\log t$
UniFastSvrg	$\frac{(L_f + L_{\hat{g}})D^2}{n(t - \log\log n)^2} + t(\delta_f + \delta_{\hat{g}})$	nt

Note: Rates are in terms of expected function residual and BigO-notation. Assume $\bar{g}$ is n times more expensive than $\hat{g}$. For UniFastSvrg, set $N = \Theta(n)$.

Corollary: Problems with Hölder-Smooth Components

Problem: $f(x) = \mathbb{E}_\xi[f_\xi(x)]$ with convex and $(\nu, H_\xi(\nu))$-Hölder-smooth f_ξ.
Standard mini-batch oracle: $g_b(x, \xi_{[b]}) = \frac{1}{b}\sum_{j=1}^b \nabla f_{\xi_j}(x)$.

Method	SO complexity
UniSgd	$\big(\frac{H_f(\nu)D^{1+\nu}}{\epsilon}\big)^{\frac{2}{1+\nu}} + \frac{1}{b}\min\big\{\frac{\sigma^2 D^2}{\epsilon^2}, \big(\frac{H_{\max}(\nu)}{\epsilon}\big)^{\frac{2}{1+\nu}}D^2 + \frac{\sigma_*^2 D^2}{\epsilon^2}\big\}$
UniFastSgd	$\big(\frac{H_f(\nu)D^{1+\nu}}{\epsilon}\big)^{\frac{2}{1+3\nu}} + \frac{1}{b}\min\big\{\frac{\sigma^2 D^2}{\epsilon^2}, \big(\frac{H_{\max}(\nu)}{\epsilon}\big)^{\frac{2}{1+\nu}}D^2 + \frac{\sigma_*^2 D^2}{\epsilon^2}\big\}$
UniSvrg	$\big[N_\nu(\epsilon) := \big(\frac{H_f(\nu)D^{1+\nu}}{\epsilon}\big)^{\frac{2}{1+\nu}} + \frac{1}{b}\big(\frac{H_{\max}(\nu)}{\epsilon}\big)^{\frac{2}{1+\nu}}D^2\big] + n_b \log_+ N_\nu(\epsilon)$
UniFastSvrg	$\big[\frac{n_b^\nu H_f(\nu)D^{1+\nu}}{\epsilon}\big]^{\frac{2}{1+3\nu}} + \big[\frac{n_b^\nu H_{\max}(\nu)D^{1+\nu}}{b^{(1+\nu)/2}\epsilon}\big]^{\frac{2}{1+3\nu}} + n_b \log\log n_b$

Note: Complexity is for reaching ϵ-solution for expected function value (using BigO-notation), σ and σ_* refer to the variance of $\hat{g}_1$, $H_{\max}(\nu) := \sup_\xi H_\xi(\nu)$. Assume $\bar{g}$ is n_b times more expensive than $\hat{g}_b$. For UniFastSvrg, $N = \Theta(n_b)$.

Universal SVRG

SVRG oracle: $G(x,\xi) := g(x,\xi) - g(\tilde{x},\xi) + \bar{g}(\tilde{x})$.

Algorithm $\operatorname{UniSvrg}_{\hat{g},\bar{g},\psi}(x_0; D)$

$\tilde{x}_0 = x_0$, $M_0 = 0$.
for $t = 0, 1, \dots$ **do**
  $\hat{G}_t = \operatorname{SvrgOrac}_{\hat{g},\bar{g}}(\tilde{x}_t)$, $(\tilde{x}_{t+1}, x_{t+1}, M_{t+1}) \cong \operatorname{UniSgd}_{\hat{G}_t,\psi}(x_t, 2^{t+1}; D, M_t)$.

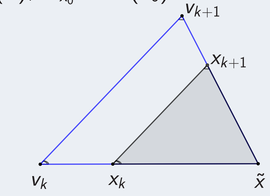
Universal Fast SVRG

Algorithm $\operatorname{UniFastSvrg}_{\hat{g},\bar{g},\psi}(x_0, N; D)$

$\tilde{x}_0 = \operatorname{Prox}_\psi(x_0, \bar{g}(x_0), 0)$, $v_0 = x_0$, $M_0 = 0$, $A_0 = \frac{1}{N}$.
for $t = 0, 1, \dots$ **do**
  $a_{t+1} = \sqrt{A_t}$, $A_{t+1} = A_t + a_{t+1}$.
  $(\tilde{x}_{t+1}, v_{t+1}, M_{t+1}) \cong \operatorname{UniTriSvrgEpoch}_{\hat{g},\bar{g},\psi}(\tilde{x}_t, v_t, M_t, A_t, a_{t+1}, N; D)$.

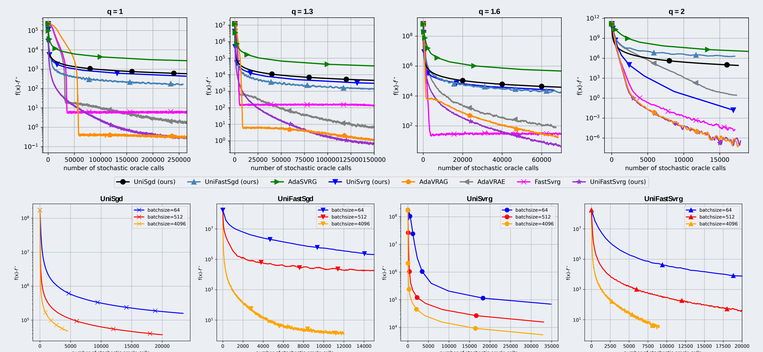
Algorithm $\operatorname{UniTriSvrgEpoch}_{\hat{g},\bar{g},\psi}(\tilde{x}, v_0, M_0, A, a, N; D)$

$A_+ = A + a$, $x_0 = \frac{A}{A_+}\tilde{x} + \frac{a}{A_+}v_0$, $\hat{G} = \operatorname{SvrgOrac}_{\hat{g},\bar{g}}(\tilde{x})$, $G_{x_0} \cong \hat{G}(x_0)$.
for $k = 0, \dots, N-1$ **do**
  $v_{k+1} = \operatorname{Prox}_\psi(v_k, G_{x_k}, \frac{M_k}{a})$.
  $x_{k+1} = \frac{A}{A_+}\tilde{x} + \frac{a}{A_+}v_{k+1}$, $G_{x_{k+1}} \cong \hat{G}(x_{k+1})$.
  $M_{k+1} = \sqrt{M_k^2 + \frac{a^2}{D^2}\|G_{x_{k+1}} - G_{x_k}\|_*^2}$.
return $(\bar{x}_N, v_N, M_N)$, where $\bar{x}_N := \frac{1}{N}\sum_{k=1}^N x_k$.



Experiments

Polyhderon feasibility problem: $\min_{\|x\| \le R}\big\{f(x) := \frac{1}{n}\sum_{i=1}^n [\langle a_i, x\rangle - b_i]_+^q\big\}$.



AdaSVRG from (Dubois-Taine et al. 2022), AdaVRAG/AdaVRAE from (Liu et al. 2022), FastSVRG ≈ VRADA from (Song et al. 2020).

References

O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146:37–75, 2013. DOI: 10.1007/s10107-013-0677-5.

B. Dubois-Taine, S. Vaswani, R. Babanezhad, M. Schmidt, and S. Lacoste-Julien. SVRG meets AdaGrad: painless variance reduction. *Machine Learning*, 111(12):4359–4409, 2022.

Z. Liu, T. D. Nguyen, A. Ene, and H. Nguyen. Adaptive Accelerated (Extra-)Gradient Methods with Variance Reduction. In *International Conference on Machine Learning*, pages 13947–13994, 2022.

C. Song, Y. Jiang, and Y. Ma. Variance Reduction via Accelerated Dual Averaging for Finite-Sum Optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 833–844, 2020.