Making SGD Parameter-Free

Presented by Qilin Ye

March 26, 2023

§0 Stochastic Gradient Descent

... the same old SGD:

$$x_{t+1} := x_t - \eta \nabla F(x_t)$$

where F is convex & differentiable.

Non-differentiable? Use unbiased **subgradients**: $x_{t+1} := x_t - \eta g_t$.[1]

[1] A subgradient of f satisfies $f(z) \geqslant f(x) + g^T(z - x)$ for all z.

... the same old SGD:

$$x_{t+1} := x_t - \eta \nabla F(x_t)$$

where F is convex & differentiable.

Non-differentiable? Use unbiased **subgradient**s: $x_{t+1} := x_t - \eta g_t$.[1]

[1] A subgradient of f satisfies $f(z) \geqslant f(x) + g^T(z - x)$ for all z.

§0 Stolchastic Gradient Descent... Problems?

Apparently, choosing the correct learning rate is not a trivial job.

(1) Too large? Possible oscillation. Too small? Slow!

(2) Distance between starting point and optimum matters.

(3) The rate of convergence is affected by scaling.

(4) ...

Apparently, choosing the correct learning rate is not a trivial job.

(1) Too large? Possible oscillation. Too small? Slow!

(2) Distance between starting point and optimum matters.

(3) The rate of convergence is affected by scaling.

(4) ...

Apparently, choosing the correct learning rate is not a trivial job.

(1) Too large? Possible oscillation. Too small? Slow!

(2) Distance between starting point and optimum matters.

(3) The rate of convergence is affected by scaling.

(4) ...

We aim to design **parameter-free** algorithms that "automatically" tune the learning rate.

And we aim to obtain "good" **regret** guarantees.

We aim to design **parameter-free** algorithms that “automatically” tune the learning rate.

And we aim to obtain “good” **regret** guarantees.

§0 Notations and Problem Setup

(1) Let $\mathcal{X} \subset \mathbb{R}^d$ be convex closed and let $f : \mathcal{X} \to \mathbb{R}$ be convex.

(2) Let x^* a minimum of f, assuming existence.

(3) Let $\mathcal{O}$ be an oracle that is a subgradient of f in expectation: $\mathbb{E}[\mathcal{O}(x) \mid x] \in \partial f(x)$.

(4) Denote the iterates by $x_0, x_1(\eta), x_2(\eta), \ldots$ and (sub)gradients $g_0, g_1(\eta), g_2(\eta), \ldots$. Define $\overline{x}(\eta) := T^{-1} \sum_{i<T} x_i(\eta)$.

(5) Distance to optimum and running maximum distance:

$$d_t(\eta) := \|x_t(\eta) - x^*\| \qquad \overline{d}_t(\eta) := \max_{i \leqslant t} d_i(\eta).$$

(6) Distance to x_0 and running max distance: $r_t(\eta), \overline{r}_t(\eta)$.

(7) Oracle error & running squared norms of oracles:

$$\Delta_i := g_i - \nabla f(x_i(\eta)) \qquad G_t(\eta) := \sum_{i<t} \|g_i(\eta)\|^2.$$

§0 Notations and Problem Setup

(1) Let $\mathcal{X} \subset \mathbb{R}^d$ be convex closed and let $f : \mathcal{X} \to \mathbb{R}$ be convex.

(2) Let x^* a minimum of f, assuming existence.

(3) Let $\mathcal{O}$ be an oracle that is a subgradient of f in expectation: $\mathbb{E}[\mathcal{O}(x) \mid x] \in \partial f(x)$.

(4) Denote the iterates by $x_0, x_1(\eta), x_2(\eta), \ldots$ and (sub)gradients $g_0, g_1(\eta), g_2(\eta), \ldots$. Define $\overline{x}(\eta) := T^{-1} \sum_{i<T} x_i(\eta)$.

(5) Distance to optimum and running maximum distance:

$$d_t(\eta) := \|x_t(\eta) - x^*\| \qquad \overline{d}_t(\eta) := \max_{i \leqslant t} d_i(\eta).$$

(6) Distance to x_0 and running max distance: $r_t(\eta), \overline{r}_t(\eta)$.

(7) Oracle error & running squared norms of oracles:

$$\Delta_i := g_i - \nabla f(x_i(\eta)) \qquad G_t(\eta) := \sum_{i<t} \|g_i(\eta)\|^2.$$

(1) Let $\mathcal{X} \subset \mathbb{R}^d$ be convex closed and let $f : \mathcal{X} \to \mathbb{R}$ be convex.

(2) Let x^* a minimum of f, assuming existence.

(3) Let $\mathcal{O}$ be an oracle that is a subgradient of f in expectation: $\mathbb{E}[\mathcal{O}(x) \mid x] \in \partial f(x)$.

(4) Denote the iterates by $x_0, x_1(\eta), x_2(\eta), \ldots$ and (sub)gradients $g_0, g_1(\eta), g_2(\eta), \ldots$. Define $\overline{x}(\eta) := T^{-1} \sum_{i<T} x_i(\eta)$.

(5) Distance to optimum and running maximum distance:

$$d_t(\eta) := \|x_t(\eta) - x^*\| \qquad \overline{d}_t(\eta) := \max_{i \leqslant t} d_i(\eta).$$

(6) Distance to x_0 and running max distance: $r_t(\eta), \overline{r}_t(\eta)$.

(7) Oracle error & running squared norms of oracles:

$$\Delta_i := g_i - \nabla f(x_i(\eta)) \qquad G_t(\eta) := \sum_{i<t} \|g_i(\eta)\|^2.$$

§0 Notations and Problem Setup

(1) Let $\mathcal{X} \subset \mathbb{R}^d$ be convex closed and let $f : \mathcal{X} \to \mathbb{R}$ be convex.

(2) Let x^* a minimum of f, assuming existence.

(3) Let $\mathcal{O}$ be an oracle that is a subgradient of f in expectation: $\mathbb{E}[\mathcal{O}(x) \mid x] \in \partial f(x)$.

(4) Denote the iterates by $x_0, x_1(\eta), x_2(\eta), \ldots$ and (sub)gradients $g_0, g_1(\eta), g_2(\eta), \ldots$. Define $\overline{x}(\eta) := T^{-1} \sum_{i<T} x_i(\eta)$.

(5) Distance to optimum and running maximum distance:

$$d_t(\eta) := \|x_t(\eta) - x^*\| \qquad \overline{d}_t(\eta) := \max_{i \leqslant t} d_i(\eta).$$

(6) Distance to x_0 and running max distance: $r_t(\eta), \overline{r}_t(\eta)$.

(7) Oracle error & running squared norms of oracles:

$$\Delta_i := g_i - \nabla f(x_i(\eta)) \qquad G_t(\eta) := \sum_{i<t} \|g_i(\eta)\|^2.$$

§0 Notations and Problem Setup

(1) Let $\mathcal{X} \subset \mathbb{R}^d$ be convex closed and let $f : \mathcal{X} \to \mathbb{R}$ be convex.

(2) Let x^* a minimum of f, assuming existence.

(3) Let $\mathcal{O}$ be an oracle that is a subgradient of f in expectation: $\mathbb{E}[\mathcal{O}(x) \mid x] \in \partial f(x)$.

(4) Denote the iterates by $x_0, x_1(\eta), x_2(\eta), \ldots$ and (sub)gradients $g_0, g_1(\eta), g_2(\eta), \ldots$. Define $\overline{x}(\eta) := T^{-1} \sum_{i<T} x_i(\eta)$.

(5) Distance to optimum and running maximum distance:

$$d_t(\eta) := \|x_t(\eta) - x^*\| \qquad \overline{d}_t(\eta) := \max_{i \leqslant t} d_i(\eta).$$

(6) Distance to x_0 and running max distance: $r_t(\eta), \overline{r}_t(\eta)$.

(7) Oracle error & running squared norms of oracles:

$$\Delta_i := g_i - \nabla f(x_i(\eta)) \qquad G_t(\eta) := \sum_{i<t} \|g_i(\eta)\|^2.$$

Fact 1

If all $\|g_i\|$'s are uniformly bounded by $L > 0$, then setting η to be the fixed point of

$$\eta \mapsto \frac{\|x_0 - x^*\|}{(\sum_{i<T} \|g_i(\eta)\|^2)^{1/2}} = \frac{d_0}{\sqrt{G_T(\eta)}}$$

satisfies the optimal error bound for the average iterate after T iterations:

$$f(\overline{x}) - f(x^*) \leqslant \frac{d_0\sqrt{G_T(\eta)}}{T} = O(d_0 L T^{-1/2}).$$

Fact 2: SoTA w/out Knowing $d_0 = \|x_0 - x^*\|$ a priori

... gains an additional logarithmic factor:

$$O\left(d_0\sqrt{\log(1+Td_0^2\epsilon^{-2})/T} + \epsilon/T\right).$$

§0 What Did This Paper Do?

(1) For any prescribed $\epsilon > 0$ and $\delta \in (0, 1)$, this paper provides a $1 - \delta$ probability optimality gap with an additional log factor:

$$O\left((d_0 T^{-1/2} + \epsilon T^{-1}) \cdot \log^2(\delta^{-1} \log(d_0 T \epsilon^{-1})\right)$$

(2) Strong localization guarantee: the average iterate (as well as other intermediate outputs) $\overline{x}$ satisfies $\|\overline{x} - x^*\| = O(\|x_0 - x^*\|)$.

(3) Good adaptivity to gradient norms and other scenarios.

(1) For any prescribed $\epsilon > 0$ and $\delta \in (0, 1)$, this paper provides a $1 - \delta$ probability optimality gap with an additional log factor:

$$O\Big((d_0 T^{-1/2} + \epsilon T^{-1}) \cdot \log^2(\delta^{-1} \log(d_0 T \epsilon^{-1})\Big)$$

(2) Strong localization guarantee: the average iterate (as well as other intermediate outputs) $\overline{x}$ satisfies $\|\overline{x} - x^*\| = O(\|x_0 - x^*\|)$.

(3) Good adaptivity to gradient norms and other scenarios.

§0 What Did This Paper Do?

(1) For any prescribed $\epsilon > 0$ and $\delta \in (0, 1)$, this paper provides a $1 - \delta$ probability optimality gap with an additional log factor:

$$O\Big((d_0 T^{-1/2} + \epsilon T^{-1}) \cdot \log^2(\delta^{-1} \log(d_0 T \epsilon^{-1})\Big)$$

(2) Strong localization guarantee: the average iterate (as well as other intermediate outputs) $\overline{x}$ satisfies $\|\overline{x} - x^*\| = O(\|x_0 - x^*\|)$.

(3) Good adaptivity to gradient norms and other scenarios.

This page is intentionally left blank.

§1 High-Level Idea: Using Proxy for d_0

In SGD, the output iterates $x_t(\eta)$ should ideally converge to x^* (recall the optimal bound with knowledge of x^*)

$$\Rightarrow \frac{r_t(\eta)}{\sqrt{G_T(\eta)}} \text{ converges to } \frac{d_0}{\sqrt{G_T(\eta)}}.$$

Instead of computing the uncomputable fixed point, we resort to approximating the fixed point of

$$\eta \mapsto \frac{\bar{r}_T(\eta)}{\sqrt{\alpha G_T(\eta) + \beta}}. \tag{FP1}$$

(*Why* $\bar{r}_T$ *instead of* r_T*?*)

Assuming we have found the η satisfying (FP1), and with probability 1 our oracle $\mathcal{O}(x) = \nabla f(x)$ (i.e. *true* gradient):

Proposition 1

If $\alpha > 1, \beta = 0$, then the average iterate $\overline{x} := T^{-1} \sum_{i<T} x_i(\eta)$ satisfies

$$\|\overline{x} - x^*\| \leqslant \frac{2\alpha}{\alpha - 1} \|x_0 - x^*\| = \frac{2\alpha}{\alpha - 1} d_0$$

and

$$f(\overline{x}) - f(x^*) \leqslant \frac{\alpha^{3/2}}{\alpha - 1} \cdot \frac{d_0 \sqrt{G_T(\eta)}}{T} \sim \frac{d_0 \sqrt{G_T(\eta)}}{T}.$$

(*This is the optimal regret bound!*)

The mapping $\varphi : \eta \mapsto \bar{r}_T(\eta)/\sqrt{\alpha G_T(\eta) + \beta}$ may not be continuous...

$\Longrightarrow$ a fixed point may not exist!

Workaround: find a small interval where $\eta \mapsto \varphi(\eta) - \eta$ changes sign...

$\Longrightarrow$ Bisection!

The mapping $\varphi : \eta \mapsto \bar{r}_T(\eta)/\sqrt{\alpha G_T(\eta) + \beta}$ may not be continuous...

$\Rightarrow$ a fixed point may not exist!

Workaround: find a small interval where $\eta \mapsto \varphi(\eta) - \eta$ changes sign...

$\Rightarrow$ Bisection!

The mapping $\varphi : \eta \mapsto \bar{r}_T(\eta)/\sqrt{\alpha G_T(\eta) + \beta}$ may not be continuous...

$\Rightarrow$ a fixed point may not exist!

Workaround: find a small interval where $\eta \mapsto \varphi(\eta) - \eta$ changes sign...

$\Rightarrow$ Bisection!

Also, what if the exact gradient assumption is removed?

This page is intentionally left blank.

Algorithm 1: Parameter-free SGD step size tuning (exact gradient version)[2]

Inputs: initial learning rate $\eta_\epsilon > 0$, total gradient budget $B \in \mathbb{N}$, damping parameters $\{\alpha^{(k)}, \beta^{(k)}\}$.

for $k = 2, 4, 8, 16, \ldots$ **do**

  if $k > B/4$ **then return** x_0 (??) ▷ edge case, bad B; make it larger!

  $T_k \leftarrow \lfloor B/(2k) \rfloor$ ▷ dynamically adjust SGD complexity based on k

  $\eta_0 \leftarrow \texttt{RootFindingBisection}(\eta_\epsilon, 2^{2^k}\eta_\epsilon; T_k, \alpha^{(k)}, \beta^{(k)})$

  if `Bisection` says k is OK (??) **then return** $T_k^{-1} \sum_{i<T_k} x_i(\eta_0)$

[2] (??) to be explained later.

§2 The Algorithm (exact gradient version)

Algorithm 2: Parameter-free SGD step size tuning (exact gradient version)[2]

Inputs: initial learning rate $\eta_\epsilon > 0$, total gradient budget $B \in \mathbb{N}$, damping parameters $\{\alpha^{(k)}, \beta^{(k)}\}$.
for $k = 2, 4, 8, 16, \ldots$ **do**
    if $k > B/4$ **then return** x_0 (??) ▷ edge case, bad B; make it larger!
    $T_k \leftarrow \lfloor B/(2k) \rfloor$ ▷ dynamically adjust SGD complexity based on k
    $\eta_0 \leftarrow \texttt{RootFindingBisection}(\eta_\epsilon, 2^{2^k}\eta_\epsilon; T_k, \alpha^{(k)}, \beta^{(k)})$
    if Bisection says k is OK (??) **then return** $T_k^{-1} \sum_{i<T_k} x_i(\eta_0)$

[2] (??) to be explained later.

Algorithm 3: Parameter-free SGD step size tuning (exact gradient version)[2]

Inputs: initial learning rate $\eta_\epsilon > 0$, total gradient budget $B \in \mathbb{N}$, damping parameters $\{\alpha^{(k)}, \beta^{(k)}\}$.

for $k = 2, 4, 8, 16, \ldots$ **do**

  if $k > B/4$ **then return** x_0 (??) ▹ edge case, bad B; make it larger!

  $T_k \leftarrow \lfloor B/(2k) \rfloor$ ▹ dynamically adjust SGD complexity based on k

  $\eta_0 \leftarrow$ `RootFindingBisection`$(\eta_\epsilon, 2^{2^k}\eta_\epsilon; T_k, \alpha^{(k)}, \beta^{(k)})$

  if `Bisection` says k is OK (??) **then return** $T_k^{-1} \sum_{i<T_k} x_i(\eta_0)$

[2] (??) to be explained later.

§2 The Algorithm (exact gradient version)

Algorithm 4: Parameter-free SGD step size tuning (exact gradient version)[2]

Inputs: initial learning rate $\eta_\epsilon > 0$, total gradient budget $B \in \mathbb{N}$, damping parameters $\{\alpha^{(k)}, \beta^{(k)}\}$.

for $k = 2, 4, 8, 16, \ldots$ **do**

- **if** $k > B/4$ **then return** x_0 (??) ▹ edge case, bad B; make it larger!
- $T_k \leftarrow \lfloor B/(2k) \rfloor$ ▹ dynamically adjust SGD complexity based on k
- $\eta_0 \leftarrow \texttt{RootFindingBisection}(\eta_\epsilon, 2^{2^k}\eta_\epsilon; T_k, \alpha^{(k)}, \beta^{(k)})$
- **if** `Bisection` says k is OK (??) **then return** $T_k^{-1} \sum_{i<T_k} x_i(\eta_0)$

[2] (??) to be explained later.

Lemma 1

With appropriate parameters, under exact gradient setting,

$$\eta \leqslant \varphi(\eta) \Rightarrow \bar{d}_T(\eta) \leqslant \frac{\alpha+1}{\alpha-1} \cdot d_0 \quad \text{and} \quad \bar{r}_T(\eta) \leqslant \frac{2\alpha}{\alpha-1} d_0.$$

Proof. First notice that $d_{i+1}^2 = \|x_i - \eta g_i - x^*\|^2 = d_i^2 - 2\eta \langle g_i, x_i - x^* \rangle + \eta^2 \|g_i\|^2$. Then, by (sub)gradient and convexity of f, we also have $\langle g_i, x_i - x^* \rangle \geqslant f(x_i) - f(x^*) \geqslant 0$. Summation over all $i < t$ gives $d_t^2 \leqslant d_0^2 + \eta^2 G_t$. The remainder of the proof are pure algebraic manipulations and are likely not ideal for presentation.

Lemma 1

With appropriate parameters, under exact gradient setting,

$$\eta \leqslant \varphi(\eta) \Rightarrow \overline{d}_T(\eta) \leqslant \frac{\alpha+1}{\alpha-1} \cdot d_0 \quad \text{and} \quad \overline{r}_T(\eta) \leqslant \frac{2\alpha}{\alpha-1} d_0.$$

Proof. First notice that $d_{i+1}^2 = \|x_i - \eta g_i - x^*\|^2 = d_i^2 - 2\eta \langle g_i, x_i - x^* \rangle + \eta^2 \|g_i\|^2$. Then, by (sub)gradient and convexity of f, we also have $\langle g_i, x_i - x^* \rangle \geqslant f(x_i) - f(x^*) \geqslant 0$. Summation over all $i < t$ gives $d_t^2 \leqslant d_0^2 + \eta^2 G_t$. The remainder of the proof are pure algebraic manipulations and are likely not ideal for presentation.

Lemma 2

With appropriate parameters, under exact gradient setting, if the following holds, then $\eta > \varphi(\eta)$:

$$\eta > \eta_{\max} := \frac{2\alpha}{\alpha - 1} \cdot \frac{d_0}{\sqrt{\alpha \|g_0\|^2 + \beta}}.$$

Proof. If $\eta > \eta_{\max}$ but $\eta \leqslant \varphi(\eta)$, using $\|g_0\|^2 \leqslant \sum \|g_i\|^2 = G_T(\eta)$ we obtain

$$\frac{\bar{r}_T(\eta)}{\sqrt{\alpha\|g_0\|^2 + \beta}} \geqslant \frac{\bar{r}_T(\eta)}{\sqrt{\alpha G_T(\eta) + \beta}} = \varphi(\eta) \geqslant \eta > \eta_{\max} = \frac{2\alpha}{\alpha - 1} \cdot \frac{d_0}{\sqrt{\alpha\|g_0\|^2 + \beta}},$$

contradicting the previous lemma.

Upshot: if k is such that $2^{2^k}\eta_\epsilon > \dfrac{2\alpha}{\alpha - 1} \cdot \dfrac{d_0}{\sqrt{\alpha\|g_0\|^2 + \beta}}$, then

$$\eta \mapsto \varphi(\eta) - \eta$$

changes sign on $[\eta_\epsilon, 2^{2^k}\eta_\epsilon]$. Time for bisection!

§2 The Algorithm – RootFindingBisection

Function `RootFindingBisection`($\eta_{\text{low}}, \eta_{\text{high}}; T, \alpha, \beta$):

  define φ by $\varphi(\eta) = \bar{r}_T(\eta)/\sqrt{\alpha G_T(\eta) + \beta}$ ▷ bisection target

  if $\eta_{\text{high}} \leqslant \varphi(\eta_{\text{high}})$ **then return** ∞ ▷ η_{high} too low, need to increase

  if $\eta_{\text{low}} > \varphi(\eta_{\text{low}})$ **then return** η_{low} ▷ η_{low} is sufficient (if small)

  while $\eta_{\text{high}} > 2\eta_{\text{low}}$ **do**

    ▷ loop invariant: $\eta_{\text{low}} < \eta_{\text{high}}, \eta_{\text{low}} \leqslant \varphi(\eta_{\text{low}}), \eta_{\text{high}} > \varphi(\eta_{\text{high}})$

    $\eta_{\text{mid}} \leftarrow \sqrt{\eta_{\text{low}}\eta_{\text{high}}}$

    if $\eta_{\text{mid}} \leqslant \varphi(\eta_{\text{mid}})$ **then** $\eta_{\text{low}} \leftarrow \eta_{\text{mid}}$ **else** $\eta_{\text{high}} \leftarrow \eta_{\text{mid}}$

  if $\bar{r}_T(\eta_{\text{high}}) \leqslant \bar{r}_T(\eta_{\text{low}}) \cdot \varphi(\eta_{\text{high}})/\eta_{\text{high}}$ **then return** η_{high}

  else return η_{low}

§2 The Algorithm – `RootFindingBisection`

Function `RootFindingBisection`($\eta_{\text{low}}, \eta_{\text{high}}; T, \alpha, \beta$)**:**

    define φ by $\varphi(\eta) = \bar{r}_T(\eta)/\sqrt{\alpha G_T(\eta) + \beta}$ ▷ bisection target

    if $\eta_{\text{high}} \leqslant \varphi(\eta_{\text{high}})$ **then return** ∞ ▷ η_{high} too low, need to increase

    if $\eta_{\text{low}} > \varphi(\eta_{\text{low}})$ **then return** η_{low} ▷ η_{low} is sufficient (if small)

    while $\eta_{\text{high}} > 2\eta_{\text{low}}$ **do**

        ▷ loop invariant: $\eta_{\text{low}} < \eta_{\text{high}}, \eta_{\text{low}} \leqslant \varphi(\eta_{\text{low}}), \eta_{\text{high}} > \varphi(\eta_{\text{high}})$

        $\eta_{\text{mid}} \leftarrow \sqrt{\eta_{\text{low}}\eta_{\text{high}}}$

        if $\eta_{\text{mid}} \leqslant \varphi(\eta_{\text{mid}})$ **then** $\eta_{\text{low}} \leftarrow \eta_{\text{mid}}$ **else** $\eta_{\text{high}} \leftarrow \eta_{\text{mid}}$

    if $\bar{r}_T(\eta_{\text{high}}) \leqslant \bar{r}_T(\eta_{\text{low}}) \cdot \varphi(\eta_{\text{high}})/\eta_{\text{high}}$ **then return** η_{high}

    else return η_{low}

§2 The Algorithm – `RootFindingBisection`

```
Function RootFindingBisection(η_low, η_high; T, α, β):
    define φ by φ(η) = r̄_T(η)/√(αG_T(η) + β)                ▷ bisection target
    if η_high ⩽ φ(η_high) then return ∞        ▷ η_high too low, need to increase
    if η_low > φ(η_low) then return η_low           ▷ η_low is sufficient (if small)
    while η_high > 2η_low do
        ▷ loop invariant: η_low < η_high, η_low ⩽ φ(η_low), η_high > φ(η_high)
        η_mid ← √(η_low η_high)
        if η_mid ⩽ φ(η_mid) then η_low ← η_mid else η_high ← η_mid
    if r̄_T(η_high) ⩽ r̄_T(η_low) · φ(η_high)/η_high then return η_high
    else return η_low
```

§2 The Algorithm – `RootFindingBisection`

Function `RootFindingBisection`($\eta_{\text{low}}, \eta_{\text{high}}; T, \alpha, \beta$)**:**

  define φ by $\varphi(\eta) = \bar{r}_T(\eta)/\sqrt{\alpha G_T(\eta) + \beta}$ ▷ bisection target

  if $\eta_{\text{high}} \leqslant \varphi(\eta_{\text{high}})$ **then return** ∞ ▷ η_{high} too low, need to increase

  if $\eta_{\text{low}} > \varphi(\eta_{\text{low}})$ **then return** η_{low} ▷ η_{low} is sufficient (if small)

  while $\eta_{\text{high}} > 2\eta_{\text{low}}$ **do**

    ▷ loop invariant: $\eta_{\text{low}} < \eta_{\text{high}}, \eta_{\text{low}} \leqslant \varphi(\eta_{\text{low}}), \eta_{\text{high}} > \varphi(\eta_{\text{high}})$

    $\eta_{\text{mid}} \leftarrow \sqrt{\eta_{\text{low}}\eta_{\text{high}}}$

    if $\eta_{\text{mid}} \leqslant \varphi(\eta_{\text{mid}})$ **then** $\eta_{\text{low}} \leftarrow \eta_{\text{mid}}$ **else** $\eta_{\text{high}} \leftarrow \eta_{\text{mid}}$

  if $\bar{r}_T(\eta_{\text{high}}) \leqslant \bar{r}_T(\eta_{\text{low}}) \cdot \varphi(\eta_{\text{high}})/\eta_{\text{high}}$ **then return** η_{high}

  else return η_{low}

§2 The Algorithm – `RootFindingBisection`

Function `RootFindingBisection`$(\eta_{\text{low}}, \eta_{\text{high}}; T, \alpha, \beta)$**:**

- define φ by $\varphi(\eta) = \bar{r}_T(\eta)/\sqrt{\alpha G_T(\eta) + \beta}$ ▷ bisection target
- **if** $\eta_{\text{high}} \leqslant \varphi(\eta_{\text{high}})$ **then return** ∞ ▷ η_{high} too low, need to increase
- **if** $\eta_{\text{low}} > \varphi(\eta_{\text{low}})$ **then return** η_{low} ▷ η_{low} is sufficient (if small)
- **while** $\eta_{\text{high}} > 2\eta_{\text{low}}$ **do**
  - ▷ loop invariant: $\eta_{\text{low}} < \eta_{\text{high}}, \eta_{\text{low}} \leqslant \varphi(\eta_{\text{low}}), \eta_{\text{high}} > \varphi(\eta_{\text{high}})$
  - $\eta_{\text{mid}} \leftarrow \sqrt{\eta_{\text{low}}\eta_{\text{high}}}$
  - **if** $\eta_{\text{mid}} \leqslant \varphi(\eta_{\text{mid}})$ **then** $\eta_{\text{low}} \leftarrow \eta_{\text{mid}}$ **else** $\eta_{\text{high}} \leftarrow \eta_{\text{mid}}$
- **if** $\bar{r}_T(\eta_{\text{high}}) \leqslant \bar{r}_T(\eta_{\text{low}}) \cdot \varphi(\eta_{\text{high}})/\eta_{\text{high}}$ **then return** η_{high}
- **else return** η_{low}

§2 Properties of `RootFindingBisection`

(1) Each iteration halves $\log(\eta_{\text{high}}/\eta_{\text{low}})$

$\Rightarrow$ number of iterations is $\log\log(\eta_{\text{high}}/\eta_{\text{low}})$.

Consequently, by Lemma 2, when our algorithm terminates, $k \leqslant 2\log\log^{+}(\eta_{\max}/\eta_{\epsilon})$.

(2) Approximates the (possibly non-existent) root $\eta = \varphi(\eta)$ up to a factor of 2, even when root is non-existent. If the returned interval is $[\eta^*_{\text{low}}, \eta^*_{\text{high}}]$ then

$$\frac{\bar{r}_T(\eta_0)}{2\sqrt{\alpha G_T(\eta^*_{\text{high}})+\beta}} \leqslant \eta_0 \leqslant \frac{\bar{r}_T(\eta^*_{\text{low}})}{\sqrt{\alpha G_T(\eta_0)+\beta}}.$$

(1) Each iteration halves $\log(\eta_{\text{high}}/\eta_{\text{low}})$
$\implies$ number of iterations is $\log\log(\eta_{\text{high}}/\eta_{\text{low}})$.
Consequently, by Lemma 2, when our algorithm terminates, $k \leqslant 2\log\log^{+}(\eta_{\max}/\eta_{\epsilon})$.

(2) Approximates the (possibly non-existent) root $\eta = \varphi(\eta)$ up to a factor of 2, even when root is non-existent. If the returned interval is $[\eta^*_{\text{low}}, \eta^*_{\text{high}}]$ then

$$\frac{\bar{r}_T(\eta_0)}{2\sqrt{\alpha G_T(\eta^*_{\text{high}}) + \beta}} \leqslant \eta_0 \leqslant \frac{\bar{r}_T(\eta^*_{\text{low}})}{\sqrt{\alpha G_T(\eta_0) + \beta}}.$$

Proposition 2

Let $\eta_0 = \texttt{RootFindingBisection}(\eta_{\text{low}}, \eta_{\text{high}}; T, \alpha, \beta)$, where $\alpha > 1, \beta > 0, T \in \mathbb{N}$, and each $\eta > 0$. Asssume $\eta_{\text{high}} > \varphi(\eta_{\text{high}})$. Let $\bar{x} = T^{-1} \sum_{i<T} x_i(\eta_0)$ be the average iterate. Under exact gradient setting:

(1) if $\eta_{\text{low}} \leqslant \varphi(\eta_{\text{low}})$ then for some $\eta' \in [\eta_0, 2\eta_0]$,

$$\|\bar{x} - x_0\| \leqslant \frac{2\alpha}{\alpha - 1} d_0 \quad \text{and} \quad f(\bar{x}) - f(x^*) \leqslant \frac{2\alpha}{\alpha - 1} \cdot \frac{d_0\sqrt{\alpha G_T(\eta') + \beta}}{T};$$

(2) if $\eta_{\text{low}} > \varphi(\eta_{\text{low}})$, then $\eta_0 = \eta_{\text{low}}$, and

$$\|\bar{x} - x_0\| \leqslant \eta_0\sqrt{\alpha G_T(\eta_0) + \beta} \quad \text{and} \quad f(\bar{x}) - f(x^*) \leqslant \frac{d_0\sqrt{\alpha G_T(\eta_0) + \beta} + \eta_0 G_T(\eta_0)}{T}.$$

Proposition 2

Let $\eta_0 = \texttt{RootFindingBisection}(\eta_{\text{low}}, \eta_{\text{high}}; T, \alpha, \beta)$, where $\alpha > 1, \beta > 0, T \in \mathbb{N}$, and each $\eta > 0$. Asssume $\eta_{\text{high}} > \varphi(\eta_{\text{high}})$. Let $\overline{x} = T^{-1} \sum_{i<T} x_i(\eta_0)$ be the average iterate. Under exact gradient setting:

(1) if $\eta_{\text{low}} \leqslant \varphi(\eta_{\text{low}})$ then for some $\eta' \in [\eta_0, 2\eta_0]$,

$$\|\overline{x} - x_0\| \leqslant \frac{2\alpha}{\alpha - 1} d_0 \qquad \text{and} \qquad f(\overline{x}) - f(x^*) \leqslant \frac{2\alpha}{\alpha - 1} \cdot \frac{d_0\sqrt{\alpha G_T(\eta') + \beta}}{T};$$

(2) if $\eta_{\text{low}} > \varphi(\eta_{\text{low}})$, then $\eta_0 = \eta_{\text{low}}$, and

$$\|\overline{x} - x_0\| \leqslant \eta_0\sqrt{\alpha G_T(\eta_0) + \beta} \quad \text{and} \quad f(\overline{x}) - f(x^*) \leqslant \frac{d_0\sqrt{\alpha G_T(\eta_0) + \beta} + \eta_0 G_T(\eta_0)}{T}.$$

Theorem: (exact gradient version)

Let $\alpha^{(k)} = 3, \beta^{(k)} = 0, n_\epsilon > 0, B \in \mathbb{N}$, and $x_0 \in \mathbb{R}^d$. With a total gradient budget of B, under exact gradient setting our algorithm will tune the learning rate to some $\eta \geqslant \eta_\epsilon$. Using the average $\bar{x}$ of

$$T \geqslant \max\left(1, \frac{B}{12 \log\log^+(\|x_0 - x^*\|/(\eta_\epsilon \|g_0\|))}\right)$$

iterates, one of the following holds.

(1) $\eta > \eta_\epsilon$, and

$$\|\bar{x} - x^*\| \leqslant 4\|x_0 - x^*\|, \qquad f(\bar{x}) - f(x^*) \leqslant \sqrt{27} \cdot \frac{\|x_0 - x^*\|\sqrt{G_T(\eta')}}{T},$$

(2) or $\eta = \eta_\epsilon$, and

$$\|\bar{x} - x^*\| \leqslant \eta_\epsilon \sqrt{3G_T(\eta_\epsilon)}, \qquad f(\bar{x}) - f(x^*) \leqslant \frac{2\eta_\epsilon G_T(\eta_\epsilon)}{T}.$$

This page is intentionally left blank.

(1) Key observation: when O outputs exact gradients, $g_i(\eta) \equiv \nabla f(x_i(\eta))$.

(2) This means that under exact gradient setting,

$$\sum_{i<T} \langle \Delta_i(\eta), x_i(\eta) - x^* \rangle = 0.$$

(3) Generalize above into "approximately:" for $T \in \mathbb{N}$, and $\alpha, \beta, \eta > 0$, define the "**good events**" to be

$$\mathfrak{E}(\eta) = \mathfrak{E}(\eta; T, \alpha, \beta) := \bigcap_{t \leqslant T} \left\{ \sum_{i<t} \langle \Delta_i(\eta), x_i(\eta) - x^* \rangle \geqslant -\frac{1}{4} \max(\bar{d}_t(\eta), \eta\sqrt{\beta}) \sqrt{\alpha G_t(\eta) + \beta} \right\}.$$

(1) Key observation: when O outputs exact gradients, $g_i(\eta) \equiv \nabla f(x_i(\eta))$.

(2) This means that under exact gradient setting,

$$\sum_{i<T} \langle \Delta_i(\eta), x_i(\eta) - x^* \rangle = 0.$$

(3) Generalize above into "approximately:" for $T \in \mathbb{N}$, and $\alpha, \beta, \eta > 0$, define the "**good events**" to be

$$\mathfrak{E}(\eta) = \mathfrak{E}(\eta; T, \alpha, \beta) := \bigcap_{t \leqslant T} \left\{ \sum_{i<t} \langle \Delta_i(\eta), x_i(\eta) - x^* \rangle \geqslant -\frac{1}{4} \max(\bar{d}_t(\eta), \eta\sqrt{\beta}) \sqrt{\alpha G_t(\eta) + \beta} \right\}.$$

§3 Moving Forward — Defining "Good Events"

(1) Key observation: when O outputs exact gradients, $g_i(\eta) \equiv \nabla f(x_i(\eta))$.

(2) This means that under exact gradient setting,

$$\sum_{i<T} \langle \Delta_i(\eta), x_i(\eta) - x^* \rangle = 0.$$

(3) Generalize above into "approximately:" for $T \in \mathbb{N}$, and $\alpha, \beta, \eta > 0$, define the "**good events**" to be

$$\mathfrak{E}(\eta) = \mathfrak{E}(\eta; T, \alpha, \beta) := \bigcap_{t \leqslant T} \left\{ \sum_{i<t} \langle \Delta_i(\eta), x_i(\eta) - x^* \rangle \geqslant -\frac{1}{4} \max(\bar{d}_t(\eta), \eta\sqrt{\beta}) \sqrt{\alpha G_t(\eta) + \beta} \right\}.$$

What Was That Mess?

Lemma 1 (exact gradient version)

With appropriate parameters, under exact gradient setting,

$$\eta \leqslant \varphi(\eta) \Rightarrow \overline{d}_T(\eta) \leqslant \frac{\alpha+1}{\alpha-1} \cdot d_0 \quad \text{and} \quad \overline{r}_T(\eta) \leqslant \frac{2\alpha}{\alpha-1} d_0.$$

becomes ...

Lemma 1 (stochastic version)

With appropriate parameters, under $\mathfrak{E}(\eta; T, \alpha, \beta)$, i.e., the "good event" setting, if $\eta \leqslant \varphi(\eta)$, then

$$\overline{d}_T(\eta) \leqslant \frac{3\alpha+2}{\alpha+2} d_0 \qquad \text{and} \qquad \overline{r}_T(\eta) \leqslant \frac{4\alpha}{\alpha-2} d_0.$$

What Was That Mess?

Lemma 1 (exact gradient version)

With appropriate parameters, under exact gradient setting,

$$\eta \leqslant \varphi(\eta) \Rightarrow \overline{d}_T(\eta) \leqslant \frac{\alpha+1}{\alpha-1} \cdot d_0 \quad \text{and} \quad \overline{r}_T(\eta) \leqslant \frac{2\alpha}{\alpha-1} d_0.$$

becomes ...

Lemma 1 (stochastic version)

With appropriate parameters, under $\mathfrak{E}(\eta; T, \alpha, \beta)$, i.e., the "good event" setting, if $\eta \leqslant \varphi(\eta)$, then

$$\overline{d}_T(\eta) \leqslant \frac{3\alpha+2}{\alpha+2} d_0 \qquad \text{and} \qquad \overline{r}_T(\eta) \leqslant \frac{4\alpha}{\alpha-2} d_0.$$

What Was That Mess?

Proposition 2 (exact gradient version)

Let $\eta_0 = \texttt{RootFindingBisection}(\eta_{\text{low}}, \eta_{\text{high}}; T, \alpha, \beta)$, where $\alpha > 1, \beta > 0, T \in \mathbb{N}$, and each $\eta > 0$. Asssume $\eta_{\text{high}} > \varphi(\eta_{\text{high}})$. Let $\overline{x} = T^{-1} \sum_{i<T} x_i(\eta_0)$ be the average iterate. Under exact gradient setting:

(1) if $\eta_{\text{low}} \leqslant \varphi(\eta_{\text{low}})$ then for some $\eta' \in [\eta_0, 2\eta_0]$,

$$\|\overline{x} - x_0\| \leqslant \frac{2\alpha}{\alpha - 1} d_0 \qquad \text{and} \qquad f(\overline{x}) - f(x^*) \leqslant \frac{2\alpha}{\alpha - 1} \cdot \frac{d_0\sqrt{\alpha G_T(\eta') + \beta}}{T};$$

(2) if $\eta_{\text{low}} > \varphi(\eta_{\text{low}})$, then $\eta_0 = \eta_{\text{low}}$, and

$$\|\overline{x} - x_0\| \leqslant \eta_0\sqrt{\alpha G_T(\eta_0) + \beta} \quad \text{and} \quad f(\overline{x}) - f(x^*) \leqslant \frac{d_0\sqrt{\alpha G_T(\eta_0) + \beta} + \eta_0 G_T(\eta_0)}{T}.$$

What Was That Mess?

Proposition 2 (stochastic version)

Let $\eta_0 = \texttt{RootFindingBisection}(\eta_{\text{low}}, \eta_{\text{high}}; T, \alpha, \beta)$, where $\alpha > 2, \beta > 0, T \in \mathbb{N}$, and $\eta_{\text{high}} = 2^{2^k}\eta_{\text{low}}$ for some k. Assume $\eta_{\text{high}} > \varphi(\eta_{\text{high}})$. Let $\bar{x} = T^{-1}\sum_{i<T} x_i(\eta_0)$ be the average iterate. Assume the "good events" $\bigcap_{j=0}^{2^k} \mathfrak{E}(2^j\eta_{\text{low}}; T, \alpha, \beta)$ all hold.

(1) If $\eta_{\text{low}} \leqslant \varphi(\eta_{\text{low}})$, then for some $\eta' \in [\eta_0, 2\eta_0]$,

$$\|\bar{x} - x_0\| \leqslant \frac{4\alpha}{\alpha - 2} d_0 \quad \text{and} \quad f(\bar{x}) - f(x^*) \leqslant \frac{9\alpha - 2}{2(\alpha - 2)} \cdot \frac{d_0\sqrt{\alpha G_T(\eta') + \beta}}{T};$$

(2) If $\eta_{\text{low}} > \varphi(\eta_{\text{low}})$ and in addition $\mathfrak{E}(\eta_{\text{low}}; T, \alpha, \beta)$ holds, then $\eta_0 = \eta_{\text{low}}$, and

$$\|\bar{x} - x_0\| \leqslant \eta_{\text{low}}\sqrt{\alpha G_T(\eta_{\text{low}}) + \beta} \quad \text{and} \quad f(\bar{x}) - f(x^*) \leqslant \frac{5}{4}\frac{d_0\sqrt{\alpha G_T(\eta_{\text{low}}) + \beta} + \eta_{\text{low}}(\alpha G_T(\eta_{\text{low}} + \beta))}{T}.$$

What Was That Mess?

Lemma 2 (exact gradient version)

With appropriate parameters, under exact gradient setting, if the following holds, then $\eta > \varphi(\eta)$:

$$\eta > \eta_{\max} := \frac{2\alpha}{\alpha - 1} \cdot \frac{d_0}{\sqrt{\alpha\|g_0\|^2 + \beta}}.$$

Consequently, when our algorithm terminates, $k \leqslant 2 \log \log^+(\eta_{\max}/\eta_\epsilon)$.

becomes...

Lemma 2 (stochastic version)

With appropriate parameters, if "good event" $\mathfrak{E}(\eta; T, \alpha, \beta)$ holds, then if the following implies $\eta > \varphi(\eta)$:

$$\eta > \eta_{\max} := \frac{4\alpha}{\alpha - 2} \cdot \frac{d_0}{\sqrt{\alpha\|g_0\|^2 + \beta}}.$$

Consequently, if $\bigcap_{k=2,4,8,\ldots} \mathfrak{E}(2^{2^k}\eta_\epsilon; T_k, \alpha^{(k)}, \beta^{(k)})$ holds, when our algorithm terminates, $k \leqslant 2 \log \log^+(\eta_{\max}/\eta_\epsilon)$.

What Was That Mess?

Lemma 2 (exact gradient version)

With appropriate parameters, under exact gradient setting, if the following holds, then $\eta > \varphi(\eta)$:

$$\eta > \eta_{\max} := \frac{2\alpha}{\alpha - 1} \cdot \frac{d_0}{\sqrt{\alpha \|g_0\|^2 + \beta}}.$$

Consequently, when our algorithm terminates, $k \leqslant 2 \log \log^{+}(\eta_{\max}/\eta_{\epsilon})$.

becomes...

Lemma 2 (stochastic version)

With appropriate parameters, if "good event" $\mathfrak{E}(\eta; T, \alpha, \beta)$ holds, then if the following implies $\eta > \varphi(\eta)$:

$$\eta > \eta_{\max} := \frac{4\alpha}{\alpha - 2} \cdot \frac{d_0}{\sqrt{\alpha \|g_0\|^2 + \beta}}.$$

Consequently, if $\bigcap_{k=2,4,8,\ldots} \mathfrak{E}(2^{2^k} \eta_{\epsilon}; T_k, \alpha^{(k)}, \beta^{(k)})$ holds, when our algorithm terminates, $k \leqslant 2 \log \log^{+}(\eta_{\max}/\eta_{\epsilon})$.

For the remainder of the analysis, just like Fact 1, we assume the gradient oracle is uniformly bounded by $L > 0$.

Lemma 3: "good events" are likely

Let $T \in \mathbb{N}, \eta > 0, \delta \in (0,1)$ be given. Define $C = \log(60\delta^{-1}\log^2(6T))$.
If $\alpha \geqslant 1024C$ and $\beta \geqslant 1024C^2L^2$ then $\mathbb{P}(\mathfrak{E}(\eta; T, \alpha, \beta)) \geqslant 1 - \delta$.

For the remainder of the analysis, just like Fact 1, we assume the gradient oracle is uniformly bounded by $L > 0$.

Lemma 3: "good events" are likely

Let $T \in \mathbb{N}, \eta > 0, \delta \in (0, 1)$ be given. Define $C = \log(60\delta^{-1} \log^2(6T))$.
If $\alpha \geqslant 1024C$ and $\beta \geqslant 1024C^2L^2$ then $\mathbb{P}(\mathfrak{E}(\eta; T, \alpha, \beta)) \geqslant 1 - \delta$.

Proposition 3

Let budget B, initial step size $\eta_\epsilon > 0$, and failure probability $\delta \in (0,1)$ be given. Let $\alpha^{(k)} = 1024C_k$ and $\beta^{(k)} = 1024C_k^2 L^2$, where $C_k = 2k + \log(60\delta^{-1}\log^2(6B))$. Then, $\mathbb{P}(\bigcap_{k=2,4,8,\ldots}\bigcap_{j=0,1,\ldots,2^k} \mathfrak{E}(2^j n_\epsilon; B, \alpha^{(k)}, \beta^{(k)})) \geqslant 1-\delta$.

Proof. Notice that $C_k = \log(60\log^2(6B)/(2^{-2k}\delta))$ so by the previous lemma, with $T = B$, $\alpha = \alpha^{(k)}, \beta = \beta^{(k)}$, and failure probability $2^{-2k}\delta$, for any η,

$$1 - \mathbb{P}(\mathfrak{E}(\eta; B, \alpha^{(k)}, \beta^{(k)})) \leqslant 2^{-2k}\delta.$$

By union bound

$$1 - \mathbb{P}\Big(\bigcap_{j=0}^{2^k} \mathfrak{E}(2^j\eta_\epsilon; B, \alpha^{(k)}, \beta^{(k)})\Big) \leqslant (2^k+1)2^{-2k}\delta \leqslant 2^{-(k-1)}\delta$$

and finally

$$1 - \mathbb{P}\Big(\bigcap_{k=2,4,8,\ldots}\bigcap_{j=0}^{2^k} \mathfrak{E}(2^j\eta_\epsilon; B, \alpha^{(k)}, \beta^{(k)})\Big) \leqslant \sum_{k\geqslant 1} 2^{-k}\delta = \delta.$$

Theorem: (stochastic version)

For any failure probability $\delta \in (0,1)$, budget $B \in \mathbb{N}$, starting point $x_0 \in \mathbb{R}^d$, and initial step size $\eta_\epsilon > 0$, with $\{\alpha^{(k)}, \beta^{(k)}\}$ specified as in the previous proposition, the algorithm (i) makes $\leqslant B$ gradient queries, (ii) fine-tunes the step size to $\eta \geqslant \eta_\epsilon$, and (iii) returns $\bar{x} = T^{-1} \sum_{i<T} x_i(\eta) \in \mathbb{R}^d$.

Define $C = -\log\delta + \log\log^+(B\|x^* - x_0\|/(\eta_\epsilon L))$. Then, for some $\eta' \in [\eta, 2\eta]$, the event $\{(1) \text{ and } ((2) \text{ or } (3))\}$ happens with probability $\geqslant 1 - \delta$.

$$T \geqslant \max\left(1, \frac{B}{8\log\log^+(\|x_0 - x^*\|/(n_\epsilon L))}\right) \tag{1}$$

$$\|\bar{x} - x^*\| \leqslant 6\|x_0 - x^*\| \quad \text{and} \quad f(\bar{x}) - f(x^*) = O\left(\frac{\|x_0 - x^*\|\sqrt{CG_T(\eta') + C^2L^2}}{T}\right) \tag{2}$$

$$\|\bar{x} - x^*\| = O\left(\eta_\epsilon\sqrt{CG_T(\eta_\epsilon) + C^2L^2}\right) \quad \text{and} \quad f(\bar{x}) - f(x^*) = O\left(\frac{\eta_\epsilon(CG_T(\eta_\epsilon) + C^2L^2)}{T}\right) \tag{3}$$

References

Yair Carmon and Oliver Hinder. "Making SGD Parameter-Free". In: (2022). arXiv: 2205.02160 [math.OC].