Exact convergence rates of the last iterate in subgradient methods

François Glineur and Moslem Zamani

Information and Communication Technologies, Electronics and Applied Mathematics Institute, and Center for Operations Research and Econometrics
UCLouvain

Results from preprint https://arxiv.org/abs/2307.11134

Contents

Subgradient methods

Last-iterate convergence

- Performance estimation
- Convergence rates

Extensions

- Last-iterate optimal subgradient method
- Normalized step sizes

Conclusions

Subgradient methods

Subgradient methods

Objective: minimize a function $f : \mathbb{R}^d \to \mathbb{R}$ that is

- convex

  $$\partial f(x) = \{g \text{ such that } f(y) \geq f(x) + g^T(y - x) \text{ for all } y\} \neq \emptyset$$

- B-Lipschitz continous

  $$g \in \partial f(x) \Rightarrow \|g\| \leq B$$

- with minimizer x^*

Method: subgradient method with fixed step sizes $\{h_k\}$

$$x_{k+1} = x_k - h_k g_k \text{ for some } g_k \in \partial f(x_k)$$

starting from x_0

Performance criteria

Target: convergence rate after N iterations, either

- $\min_{0 \le k \le N} f(x_k) - f(x_*)$ (method is not monotone)
- $f(\frac{1}{N+1}\sum_{k=0}^{N} x_k) - f(x_*)$ (average)
- $f(x_N) - f(x_*)$

Initial iterate assumption:

$$\|x_0 - x_*\| \le R$$

Homogeneity: rates in function values must be proportional to BR

Lower bound: no method can achieve better rate than

$$\frac{BR}{\sqrt{N+1}}$$

Lower bound proof (variation of [Drori, Teboulle 2016])

Consider following function with $d = N + 1$, $B = 1$ and $x_* = 0$

$$f(x) = \max\{0, x_1, x_2, \dots, x_{N+1}\} = \left[\max_{1 \le k \le N+1} x_k\right]_+$$

Choose starting point $x_0 = (1, 1, \dots, 1)$ with $R = \sqrt{N+1}$

- As long as $f(x_k) > 0$, subgradient $g_k \in \partial f(x_k)$ can be chosen as a basis vector e_i for some $1 \le i \le N + 1$ (and $\|g_k\| = B$)
- Induction hypothesis (H_k) (easy to check for $k = 0$)
  x_k contains at least $N + 1 - k$ components equal to 1
- Assume (H_k) for $k \le N$. Then $f(x_k) \ge 1$. So subgradient g_k can be chosen as some basis vector e_i, and x_{k+1} can differ only by at most one component from x_k, implying (H_{k+1}) holds
- Conclusion:

$$f(x_k) \;\ge\; 1 = \frac{BR}{\sqrt{N+1}} \quad \text{for all } 0 \le k \le N$$

(also for other criteria / for steps with several past subgradients)

Standard convergence analysis

Only two ingredients:

(1) *subgradient inequality* and (2) *square distance telescoping*

Ingredient (1)

$$\begin{aligned}\|x^{k+1} - x^\star\|^2 &= \|x^k - h_k g^k - x^\star\|^2 \\ &= \|x^k - x^\star\|^2 + h_k^2 \|g^k\|^2 - 2h_k \langle g^k, x^k - x^\star \rangle \\ &\leq \|x^k - x^\star\|^2 + h_k^2 \|g^k\|^2 - 2h_k \Big(f(x^k) - f(x^\star) \Big) .\end{aligned}$$

(where we have only used *subgradient inequality* $f(x^\star) - f(x^k) \geq \langle g^k, x^\star - x^k \rangle$ between x^* and x^k)

This gives an upper bound on the accuracy $f(x^k) - f(x^*)$

$$h_k\big(f(x^k) - f(x^*)\big) \leq \tfrac{1}{2}\|x^k - x^*\|^2 - \tfrac{1}{2}\|x^{k+1} - x^*\|^2 + \tfrac{1}{2} h_k^2 B^2$$

using bound on subgradient norm $\|g_k\| \leq B$

Standard convergence analysis (cont.)

Ingredient (2) From

$$h_k\big(f(x^k) - f(x^*)\big) \leq \tfrac{1}{2}\|x^k - x^*\|^2 - \tfrac{1}{2}\|x^{k+1} - x^*\|^2 + \tfrac{1}{2}h_k^2 B^2$$

telescoping (summing from $k = 0$ to $k = N$) gives

$$\sum_{k=0}^{N} h_k\big(f(x^k) - f(x^\star)\big) \leq \tfrac{1}{2}\left\|x^0 - x^\star\right\|^2 - \tfrac{1}{2}\|x^{N+1} - x^\star\|^2 + \tfrac{1}{2}B^2 \sum_{k=0}^{N} h_k^2$$

hence

$$\min_{0\leq k\leq N} f(x^k) - f(x^\star) \leq \frac{\frac{1}{2}\|x^0 - x^\star\|^2 + \frac{1}{2}B^2 \sum_{k=0}^{N} h_k^2}{\sum_{k=0}^{N} h_k}$$

Standard convergence analysis (end.)

$$\min_{0 \le k \le N} f(x^k) - f(x^\star) \le \frac{\frac{1}{2}\|x^0 - x^\star\|^2 + \frac{1}{2}B^2 \sum_{k=0}^N h_k^2}{\sum_{k=0}^N h_k}$$

- Right-hand side is convex and symmetric in stepsizes h_k, hence optimal values are constant $h_k = h$ for all k

$$\min_{0 \le k \le N} f(x^k) - f(x^\star) \le \frac{\frac{1}{2}\|x^0 - x^\star\|^2 + \frac{1}{2}B^2 (N+1) h^2}{(N+1)h}$$

- Optimal h is then $h_k = \frac{R}{B}\frac{1}{\sqrt{N+1}}$ leading to an *optimal* rate

$$\min_{0 \le k \le N} f(x^k) - f(x^\star) \le \frac{BR}{\sqrt{N+1}}$$

(same rate holds for average iterate since
$f(\frac{1}{N+1}\sum_{k=0}^N x_k) \le \frac{1}{N+1}\sum_{k=0}^N f(x_k)$)

End of story?

What about last-iterate convergence?

$$\min_{0\leq k\leq N} f(x^k) - f(x^\star) \leq \frac{BR}{\sqrt{N+1}}$$

- Says nothing about convergence of last iterate x_N
- O. Shamir, Open problem: Is averaging needed for strongly convex stochastic gradient descent? *JMLR* (2012)
- Practitioners often use the last iterate
- Storing best iterate might not be feasible (storage requirements, objective computation)
- Algorithm may correspond to a real-word dynamical system (see for example work by Nesterov and Shikhman)

Goal of this talk: study last-iterate convergence with and without performance estimation

Short history of our results

- 2012-2013: Drori and Teboulle introduce performance estimation problems (PEP)
  main idea: compute worst-case convergence rates
- 2013-2017: with Taylor and Hendrickx we further develop SDP-based PEP approach
- 2017: *Yurii* asks us "With your tool, can you tell the convergence rate of the last iterate in subgradient method?"
  We find a purely numerical rate (see next page), and no proof
- 2023: with Zamani we get back to the question and obtain a full PEP proof and a bit later a classic proof

Puzzle time

Puzzle: can you guess the convergence rate?

For constant stepsize $h = 1$ one can compute
using either PESTO (Matlab) or PEPIT (Python) toolboxes

$$f(x_N) - f(x_*) \leq BR\left[1 - N + \frac{1}{2}\left(s_N - s_N^{-1}\right)^2\right]$$

where the rate involves a mysterious sequence $\{s_k\}$:

$$s_0 = 1,\ s_1 = 2,\ s_2 = 2.5,\ s_3 = 2.9,$$

Puzzle time

Puzzle: can you guess the convergence rate?

For constant stepsize $h = 1$ one can compute
using either PESTO (Matlab) or PEPIT (Python) toolboxes

$$f(x_N) - f(x_*) \leq BR\left[1 - N + \tfrac{1}{2}\left(s_N - s_N^{-1}\right)^2\right]$$

where the rate involves a mysterious sequence $\{s_k\}$:

$$s_0 = 1,\ s_1 = 2,\ s_2 = 2.5,\ s_3 = 2.9,\ s_4 = 3.24482758621,\ \ldots$$

or

$$s_0 = 1,\ s_1 = 2,\ s_2 = \frac{5}{2},\ s_3 = \frac{29}{10},$$

Puzzle time

Puzzle: can you guess the convergence rate?

For constant stepsize $h = 1$ one can compute
using either PESTO (Matlab) or PEPIT (Python) toolboxes

$$f(x_N) - f(x_*) \leq BR\left[1 - N + \tfrac{1}{2}\left(s_N - s_N^{-1}\right)^2\right]$$

where the rate involves a mysterious sequence $\{s_k\}$:

$$s_0 = 1,\ s_1 = 2,\ s_2 = 2.5,\ s_3 = 2.9,\ s_4 = 3.24482758621,\ \ldots$$

or

$$s_0 = 1,\ s_1 = 2,\ s_2 = \frac{5}{2},\ s_3 = \frac{29}{10},\ s_4 = \frac{941}{290},\ \ldots$$

Puzzle time

Puzzle: can you guess the convergence rate?

For constant stepsize $h = 1$ one can compute
using either PESTO (Matlab) or PEPIT (Python) toolboxes

$$f(x_N) - f(x_*) \leq BR\left[1 - N + \tfrac{1}{2}\left(s_N - s_N^{-1}\right)^2\right]$$

where the rate involves a mysterious sequence $\{s_k\}$:

$$s_0 = 1,\ s_1 = 2,\ s_2 = 2.5,\ s_3 = 2.9,\ s_4 = 3.24482758621,\ \ldots$$

or

$$s_0 = 1,\ s_1 = 2,\ s_2 = \frac{5}{2},\ s_3 = \frac{29}{10},\ s_4 = \frac{941}{290},\ \ldots$$

Answer: $s_{k+1} = s_k + \frac{1}{s_k}$

This talk

Take-home messages:

- Performance estimation applied to subgradient methods
- Exact convergence rates can be obtained for the last iterate: suboptimal by a factor $O(\sqrt{\log(N)})$
- New last-iterate optimal method can be designed with linearly decreasing step sizes
- Extensions to constrained case, to normalized steps
- Inspiration for results provided by performance estimation but ultimately all proofs converted to classical style using a new key lemma

Contents

Subgradient methods

Last-iterate convergence

- Performance estimation
- Convergence rates

Extensions

- Last-iterate optimal subgradient method
- Normalized step sizes

Conclusions

Last-iterate convergence

Tool: performance estimation

For a given PEP (Performance Estimation Problem) we can

- compute the exact value of the performance criteria's worst-case = optimal value of PEP problem
- identify an explicit function (and starting point) achieving this worst-case value = primal solution of PEP problem + interpolation
- obtain an independently-checkable proof that this worst-case value is a valid (upper) bound on the performance criteria = dual multiplier of PEP problem
- all three steps can be done either numerically or analytically

For a large class of first-order methods, including fixed-step subgradient methods, these can be computed *exactly* using a semidefinite programming (SDP) problem.

Interpolation conditions for nonsmooth convex functions

To perform PEP for subgradient methods on a class of functions

we need the corresponding *interpolation conditions* explicitly

> given a list of values $(x_i, f_i, g_i)_{i \in I}$,
> does there exist a convex f with B-bounded subgradients such that
> $f(x_i) = f_i$ and $g_i \in \partial f(x_i)$ for all $i \in I = \{*, 0, 1, \ldots N\}$

Necessary and sufficient conditions:

$$f(x_i) = f_i \text{ and } g_i \in \partial f(x_i) \text{ for every } i \in I$$

$$\Leftrightarrow$$

$$f_j \geq f_i + g_i^T(x_j - x_i) \text{ for every } i, j \in I$$

$$\|g_i\| \leq B \text{ for every } i \in I$$

Leads to a convex, tractable formulation as a SDP

Results: average iterate with constant stepsize

Worst-case for constant stepsize subgradient method

$$x_{i+1} = x_i - h(\tfrac{R}{B})g_i$$

applied to convex function with B-bounded subgradients

- For *average* value of iterates $\hat{f}_N = \frac{f(x_0)+f(x_1)+\ldots+f(x_N)}{N+1}$, tight worst-case is

$$\hat{f}_N - f(x_*) \leq \begin{cases} BR\left(\frac{1}{2}h + \frac{1}{2(N+1)}\frac{1}{h}\right) & \text{when } h \geq \frac{1}{N+1} \\ BR\left(1 - \frac{N}{2}h\right) & \text{when } h \leq \frac{1}{N+1} \end{cases}$$

  (recovers result shown earlier for large h)

- Optimal constant step-size is then $h^* = \frac{1}{\sqrt{N+1}}$ (belongs to "large step" case) leading to tight worst-case

$$\hat{f}_N - f(x_*) \leq \frac{BR}{\sqrt{N+1}}$$

Results: last iterate with constant stepsize

- Define sequence $\{s_i\}_{i\geq 0} = \{1, 2, \frac{5}{2}, \frac{29}{10}, \dots\}$ with
$$s_0 = 1, s_{i+1} = s_i + \frac{1}{s_i} \text{ for all } i \geq 0$$
- No closed form but s_N^2 grows like $2(N+1) + \frac{1}{2}\log(N)$, also appears in [Nesterov 2009] (again!) for primal-dual subgradient
- For value of *last* iterate $f(x_N)$, tight worst-case is
$$f(x_N) - f(x_*) \leq \begin{cases} BR\left[(\frac{1}{2}s_N^2 - N)h + \frac{1}{2s_N^2}\frac{1}{h}\right] & \text{when } h \geq \frac{1}{s_N^2} \\ BR(1 - Nh) & \text{when } h \leq \frac{1}{s_N^2} \end{cases}$$

- No previous result with correct asymptotic rate for last iterate
- [Harvey,Liaw,Plan,Randhawa 2019] prove a $\frac{\log N}{32\sqrt{N}}$ lower bound when $B = 1$ with stepsize $h_i = \frac{1}{\sqrt{i}}$, and prove a high probability $\mathcal{O}(\frac{\log N}{\sqrt{N}})$ upper bound in stochastic case

Results: optimal stepsize and variants

- To perform N subgradient iterations, optimal stepsize is then

$$h^* = \frac{1}{s_N\sqrt{s_N^2 - 2N}}$$

and corresponding exact worst-case convergence rate becomes

$$f(x_N) - f(x_*) \leq BR\sqrt{1 - \frac{2N}{s_N^2}} \lesssim BR \cdot \sqrt{\frac{1 + \frac{1}{4}\log(N)}{N+1}}$$

- Using $h = \frac{1}{\sqrt{N+1}}$ (now known to be suboptimal for last iterate) leads to slightly worse

$$f(x_N) - f(x_*) \leq BR \cdot \left(\frac{\frac{5}{4} + \frac{1}{4}\log(N)}{\sqrt{N+1}}\right)$$

Results were obtained using the following PEP

$$
\begin{aligned}
\max\ & f^{N+1} - f^{\star} \\
\text{s. t. }\ & f^{i} - f^{j} - \left\langle \tfrac{B}{Rh}(x^{j} - x^{j+1}), x^{i} - x^{j} \right\rangle \geq 0 \quad i \in \{1, \ldots, N+1, \star\}, j \in \{1, \ldots, N\} \\
& f^{i} - f^{N+1} - \left\langle g^{N+1}, x^{i} - x^{N+1} \right\rangle \geq 0 \quad i \in \{1, \ldots, N+1, \star\} \\
& f^{i} - f^{\star} \geq 0 \quad i \in \{1, \ldots, N+1\} \\
& R^{2}h^{2} - \left\| x^{k} - x^{k+1} \right\|^{2} \geq 0 \qquad k \in \{1, \ldots, N\} \\
& B^{2} - \left\| g^{N+1} \right\|^{2} \geq 0 \\
& R^{2} - \left\| x^{1} - x^{\star} \right\|^{2} \geq 0.
\end{aligned}
$$

PEP-based proof is ... straightforward?

Define $f^i = f(x^i)$ and $\sigma_i = \frac{1}{s_{i+1}}, \quad i \in \{0, 1, ..., N\}$ and observe that

$$f^{N+1} - BR\left(\left(\tfrac{1}{2}s_{N+1}^2 - N\right)h + \tfrac{1}{2s_{N+1}^2 h}\right) + \sum_{i=1}^{N} \tfrac{B\sigma_{N-i}^2}{2Rh}\left(R^2h^2 - \left\|x^i - x^{i+1}\right\|^2\right)$$

$$+ \sum_{i=1}^{N}\sum_{j=i+1}^{N} \sigma_{N-j}\left(\sigma_{N-i} - \sigma_{N+1-i}\right)\left(f^i - f^j - \left\langle \tfrac{B}{Rh}(x^j - x^{j+1}), x^i - x^j \right\rangle\right)$$

$$+ \sum_{i=1}^{N}\left(\sigma_{N-i} - \sigma_{N+1-i}\right)\left(f^i - f^{N+1} - \left\langle g^{N+1}, x^i - x^{N+1}\right\rangle\right) + \tfrac{B\sigma_N^2}{2Rh}\left(R^2 - \left\|x^1\right\|^2\right)$$

$$+ \sigma_N \sum_{i=1}^{N} \sigma_{N-i}\left(-f^i - \left\langle \tfrac{B}{Rh}(x^i - x^{i+1}), -x^i \right\rangle\right) + \tfrac{Rh}{2B}\left(B^2 - \left\|g^{N+1}\right\|^2\right)$$

$$+ \sigma_N\left(-f^{N+1} + \left\langle g^{N+1}, x^i \right\rangle\right)$$

$$= \tfrac{-Rh}{2B}\left\|g^{N+1} - \tfrac{B}{Rh}x^{N+1} + \tfrac{B}{Rh}\sum_{i=1}^{N}\left(\sigma_{N-i} - \sigma_{N+1-i}\right)x^i\right\|^2 \leq 0.$$

Post-PEP reflections

- After staring at the PEP proof, we noticed similarities between inequality multipliers
- Grouping similar terms, we obtain Jensen-like inequalities (insight: applying Jensen $\leftrightarrow$ some sum of interpolation inequalities)
- Simplifying further we obtain a classic-style proof, that is no longer looking computer generated
- We encapsulate the main part of the proof in a *key Lemma*
- Key Lemma fully reverse-engineered from PEP but can be easily checked by hand

Key Lemma for subgradient methods

Lemma ([Zamani,G 2023])

Consider the subgradient method with fixed step sizes $\{h_k\}$

$x_{k+1} = x_k - h_k g_k$ *for some* $g_k \in \partial f(x_k)$ *for* $k = 0, 1, \ldots, N-1$

Choose $h_N > 0$ *and introduce* $N+2$ *weights* v_k *that satisfy*

$$1 = v_{-1} \leq v_0 \leq v_1 \leq \cdots \leq v_{N-1} \leq v_N$$

Then iterates after N *iterations of the subgradient method satisfy*

$$\sum_{k=0}^{N} \left(h_k v_k^2 - (v_k - v_{k-1}) \sum_{i=k}^{N} h_i v_i \right) \left(f(x^k) - f(x^*) \right)$$

$$\leq \frac{1}{2} \underbrace{\left\| x^0 - x^* \right\|^2}_{R} + \frac{1}{2} \sum_{k=0}^{N} h_k^2 v_k^2 \underbrace{\left\| g^k \right\|^2}_{B}$$

Why is the key Lemma useful?

Key Lemma inequality:

$$\sum_{k=0}^{N}\Big(h_k v_k^2 - (v_k - v_{k-1})\sum_{i=k}^{N} h_i v_i\Big)\big(f(x^k) - f(x^*)\big) \leq \tfrac{1}{2}v_0^2 R^2 + \tfrac{1}{2}B^2 \sum_{k=0}^{N} h_k^2 v_k^2$$

for any weights $1 = v_{-1} \leq v_0 \leq v_1 \leq \cdots \leq v_{N-1} \leq v_N$

- constant $v_k = 1$ recovers usual (average) rate
- but a suitable choice of $\{v_k\}$ allows us to modify coefficients in front of $f(x^k) - f(x^*)$
- in particular one can cancel all coefficients except last one in front of $f(x^N) - f(x^*)$

Idea of the proof of the key Lemma

Inequality to prove:

$$\sum_{k=0}^{N}\Big(h_k v_k^2 - (v_k - v_{k-1})\sum_{i=k}^{N} h_i v_i\Big)\big(f(x^k) - f(x^*)\big) \leq \tfrac{1}{2} v_0^2 R^2 + \tfrac{1}{2} B^2 \sum_{k=0}^{N} h_k^2 v_k^2$$

Proof uses a *generalization* of the standard telescoping proof

1. From weights v_k define *auxiliary sequence* z^k recursively

$$z^0 = x^* \quad \text{and} \quad z^k = \big(1 - \frac{v_{k-1}}{v_k}\big)x^k + \big(\frac{v_{k-1}}{v_k}\big)z^{k-1}$$

This implies

$$z^k = \big(\frac{v_0}{v_k}\big)x^* + \sum_{i=1}^{k}\big(\frac{v_i - v_{i-1}}{v_k}\big)x^i$$

(note z^k is a convex combination of x^* and iterates x^i)

Idea of the proof of the key Lemma (cont.)

2. Subgradient inequality between x^k and z^k (instead of x^*) gives

$$h_k v_k^2 \big(f(x^k) - f(z^k)\big) \leq \tfrac{1}{2} v_{k-1}^2 \|x^k - z^k\|^2 - \tfrac{1}{2} v_k^2 \|x^{k+1} - z^{k+1}\|^2 + \tfrac{1}{2} B^2 h_k^2 v_k^2$$

3. Telescoping (summing from $k = 0$ to $k = N$) gives that

$$\begin{aligned} &\sum_{k=0}^{N} h_k v_k^2 \big(f(x^k) - f(z^k)\big) \\ \leq \quad & \tfrac{1}{2} v_{-1}^2 \|x^0 - z^0\|^2 - \tfrac{1}{2} v_N^2 \|x^{N+1} - z^{N+1}\|^2 + \tfrac{1}{2} B^2 \sum_{k=0}^{N} h_k^2 v_k^2 \end{aligned}$$

implying

$$\sum_{k=0}^{N} h_k v_k^2 \big(f(x^k) - f(z^k)\big) \leq \tfrac{1}{2} \|x^0 - x^*\|^2 + \tfrac{1}{2} B^2 \sum_{k=0}^{N} h_k^2 v_k^2$$

Idea of the proof of the key Lemma (cont.)

4. Finally we need to find a lower bound on $f(x^k) - f(z^k)$ terms:

$$z^k = (\tfrac{v_0}{v_k})\hat{x} + \textstyle\sum_{i=1}^{k} (\tfrac{v_i - v_{i-1}}{v_k}) x^i$$

implies, by Jensen's inequality

$$f(z^k) \leq (\tfrac{v_0}{v_k}) f(\hat{x}) + \textstyle\sum_{i=1}^{k} (\tfrac{v_i - v_{i-1}}{v_k}) f(x^i)$$

hence

$$h_k v_k^2 \big(f(z^k) - f(x^*)\big) \geq h_k v_k \textstyle\sum_{i=1}^{k} (v_i - v_{i-1}) \big(f(x^i) - f(x^*)\big)$$

which combined with inequality from the previous step 3. gives

$$\begin{aligned} &\sum_{k=0}^{N} \left(h_k v_k^2 - (v_k - v_{k-1}) \sum_{i=k}^{N} h_i v_i \right) \Big(f(x^k) - f(x^*) \Big) \\ \leq \; &\sum_{k=0}^{N} h_k v_k^2 \big(f(x^k) - f(z^k)\big) \leq \tfrac{1}{2} \|x^0 - x^*\|^2 + \tfrac{1}{2} B^2 \sum_{k=0}^{N} h_k^2 v_k^2 \end{aligned}$$

Using the key Lemma

So we have proved

Lemma
Iterates of the subgradient methods satisfy

$$\sum_{k=0}^{N}\left(h_k v_k^2 - (v_k - v_{k-1})\sum_{i=k}^{N} h_i v_i\right)\left(f(x^k) - f(x^*)\right)$$
$$\leq \tfrac{1}{2}R^2 + \tfrac{1}{2}B^2\sum_{k=0}^{N} h_k^2 v_k^2$$

Proof of last-iterate convergence rate:

Choose weights v_k that cancel all coefficients of $f(x^k)$ except $f(x^N)$, which are

$$v_k = \frac{1}{s_{N+1-k}}$$

Exactness of convergence rate

All PEP rates are *exact* by design

(cannot be improved, even by a multiplicative/additive constant)

Follows from PEP solution, but can be made constructive by building an *explicit worst-case function*

- Function of the type $f(x) = [\max_k\{g_k^T x\}]_+$
- Recursive definition, coefficients g_k not straightforward
- Sugbradients for all iterates are g_k, have maximum norm B
- Subgradient inequality is satisfied between all pairs of iterates
- Matches exactly the announced convergence rate for the last iterate

Extensions

Last-iterate optimal subgradient method

Define the following new linearly decreasing stepsize schedule

$$x_{k+1} = x_k - \frac{R}{B}\frac{(N+1-k)}{(N+1)^{3/2}} g_k$$

Leads the optimal rate for the last iterate [Zamani,G 2023]

$$f(x_N) - f(x_*) \leq \frac{BR}{\sqrt{N+1}}$$

- Improves $\frac{15BD}{\sqrt{N+1}}$ [Jain,Nagaraj,Netrapalli 2021] for diameter D
- Same proof technique, key lemma with optimized weights v_k
- *Schedule dependence on N* is forced for optimal method (already impossible to find fixed stepsizes h_1 and h_2 that are optimal for both $N = 1$ and $N = 2$)
- *Open question*:
  Existence of a last-iterate optimal method with stepsizes independent from N and with momentum terms?

Subgradient method with normalized step sizes

Stepsizes so far feature a $\frac{R}{B}$ factor, require knowledge of R and B

- constant stepsizes $h_k = \frac{R}{B}h$ for some h
- optimal stepsizes $h_k = \frac{R}{B}\frac{(N+1-k)}{(N+1)^{3/2}}$

Need for B can be removed using normalized step sizes $\{t_k\}$

$$x_{k+1} = x_k - t_k \frac{g_k}{\|g_k\|} \text{ for some } g_k \in \partial f(x_k)$$

- All previous results are also valid with exactly the same rates if we assume $t_k = h_k B$
- constant stepsizes $t_k = Rh$ for some h
- optimal stepsizes $t_k = R\frac{(N+1-k)}{(N+1)^{3/2}}$
- Proof using key Lemma with adapted weights
- Removing dependence on R seems harder ($\rightarrow$ parameter-free)

Projected subgradient method

Solve convex constrained optimization

$$\min_{x \in X} f(x)$$

with the projected subgradient method with fixed step sizes $\{h_k\}$

$$x_{k+1} = \mathbb{P}\left[x_k - h_k g_k\right] \text{ for some } g_k \in \partial f(x_k)$$

($\mathbb{P}$ is orthogonal projection on convex set X)

- All results are also valid, with exactly the same rates (both constant and optimal stepsizes, also normalized)
- Straightforward adaptation of the key Lemma using non-expansiveness of the projection operator

Conclusions

Conclusions

Take-home messages:

- Performance estimation applied to subgradient methods
- Exact convergence rates can be obtained for the last iterate: suboptimal by a factor $O(\sqrt{\log(N)})$
- New last-iterate optimal method can be designed with linearly decreasing step sizes
- Extensions to constrained case, to normalized steps
- Inspiration for results provided by performance estimation but ultimately all proofs converted to classical style using a new key lemma

For all your performance estimation needs:

https://github.com/PerformanceEstimation

Thank you Yurii!