Efficiency of long steps in gradient method

How to best set the stepsizes of a gradient descent ?

Sophie Lequeu

LINMA2120 : Applied Mathematics seminar
Prof. François Glineur

December 18, 2024

Introduction

$$\min_{x \in \mathbb{R}^n} f(x) \qquad \text{where } f : \mathbb{R}^n \to \mathbb{R}$$

- convex function (not necessarily μ-strongly convex)
- with L-Lipschitz gradient ∇f

$$\text{Gradient Method}: \; x_{k+1} = x_k - \alpha_k \nabla f(x_k) \quad \text{given } x_0 \in \mathbb{R}^n$$

- stepsize α_k not necessarily constant,
- but must be decided a priori
(do not depend on evaluations of f or ∇f)

Introduction

Performance measure :

Convergence rate : $f(x_N) - f(x^*) \leq L \|x_0 - x^*\|^2 \, r_{\mathrm{wc}}(N)$

$:=$ Objective accuracy after N steps

How fast can it converge?

Standard result : $\mathcal{O}(1/N)$ convergence for constant $\alpha = \frac{1}{L}$ stepsize

Can we accelerate convergence without changing the GD algorithm — just by optimizing the stepsizes?

Outline

1. Constant stepsize
2. Teboulle-Vaisbourd increasing stepsizes
3. Das Gupta et al.'s steps
4. Grimmer's patterns
5. Silver steps
6. Numerical experiment

Outline

1. Constant stepsize

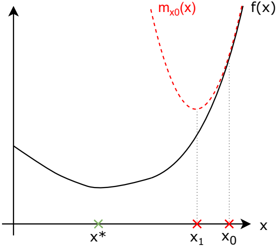
A. Classic Gradient Method : $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$

L-Lipschitz gradient : $\exists L > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \quad \forall x, y \in \mathbb{R}^n$$

L-smooth function :

$$\Leftrightarrow \quad f(x) \leq \underbrace{f(y) + \nabla f(y)^T(x-y) + \frac{L}{2}\|x-y\|^2}_{:= m_y(x)} \quad \forall x, y \in \mathbb{R}^n$$



- Minimizing the quadratic upper bound leads to the well-known Gradient Descent (GD).

$$x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$$

$$\alpha_k = \frac{1}{L} = \frac{h_k}{L} \text{ where } h_k = 1 \quad \forall k \in \mathbb{N}$$

1. Constant stepsize

A. Classic Gradient Method : $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$

- Classic convergence result :

$$\forall N > 0 \qquad f(x_N) - f(x^*) \leq \frac{L\left\|x_0 - x^*\right\|^2}{2N+2}$$

Tight ? No.

[1] Drori and Teboulle, *Performance of first-order methods for smooth convex minimization: a novel approach*.

[2] Teboulle and Vaisbourd, *An elementary approach to tight worst case complexity analysis of gradient based methods*.

1. Constant stepsize

A. Classic Gradient Method : $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$

- Classic convergence result :

$$\forall N > 0 \qquad f(x_N) - f(x^*) \leq \frac{L\left\|x_0 - x^*\right\|^2}{2N+2}$$

Tight ? No.

- Tight refinement of this bound in 2012 using Performance Estimation Techniques[1,2]:

$$\forall N > 0 \qquad f(x_N) - f(x^*) \leq \frac{L\left\|x_0 - x^*\right\|^2}{4N+2}$$

:= Exact convergence rate

[1] Drori and Teboulle, *Performance of first-order methods for smooth convex minimization: a novel approach*.

[2] Teboulle and Vaisbourd, *An elementary approach to tight worst case complexity analysis of gradient based methods*.

1. Constant stepsize

B. Convergence results

- $h = 1$: $f(x_N) - f(x^*) \leq \frac{L\ \|x_0 - x^*\|^2}{4N+2} \quad \forall N > 0$

[3] Taylor, Hendrickx, and Glineur, *Smooth Strongly Convex Interpolation and Exact Worst-case Performance of First-order Methods*.

[4] Teboulle and Vaisbourd, *An elementary approach to tight worst case complexity analysis of gradient based methods*.

1. Constant stepsize

B. Convergence results

- $h = 1$: $f(x_N) - f(x^*) \leq \frac{L\,\|x_0 - x^*\|^2}{4N+2} \quad \forall N > 0$
- $h \in (0,1)$: $f(x_N) - f(x^*) \leq \frac{L\,\|x_0 - x^*\|^2}{4Nh+2} \quad \forall N > 0$

[3] Taylor, Hendrickx, and Glineur, *Smooth Strongly Convex Interpolation and Exact Worst-case Performance of First-order Methods*.

[4] Teboulle and Vaisbourd, *An elementary approach to tight worst case complexity analysis of gradient based methods*.

1. Constant stepsize

B. Convergence results

- $h = 1$: $f(x_N) - f(x^*) \leq \frac{L\,\|x_0 - x^*\|^2}{4N+2} \quad \forall N > 0$
- $h \in (0,1)$: $f(x_N) - f(x^*) \leq \frac{L\,\|x_0 - x^*\|^2}{4Nh+2} \quad \forall N > 0$
- $h \in (1, \frac{3}{2})$[3,4]:

  $$f(x_N) - f(x^*) \leq \frac{L\,\|x_0 - x^*\|^2}{2} \max\left(\frac{1}{2Nh+1}, (1-h)^{2N}\right)$$

- $h \in (\frac{3}{2}, 2)$: Same as above, but conjectured

---

[3] Taylor, Hendrickx, and Glineur, *Smooth Strongly Convex Interpolation and Exact Worst-case Performance of First-order Methods*.

[4] Teboulle and Vaisbourd, *An elementary approach to tight worst case complexity analysis of gradient based methods*.

1. Constant stepsize

C. Optimal constant stepsize

$$f(x_N) - f(x^*) \leq \frac{L \|x_0 - x^*\|^2}{2} \max\left(\frac{1}{2Nh+1}, (1-h)^{2N}\right)$$

Suggests an optimal constant stepsize $h_k = h_{opt} \quad \forall k \in \mathbb{N}$, s.t.

$$\frac{1}{2Nh_{opt}+1} = (1-h_{opt})^{2N}$$

Drawbacks :

- Need to specify N a priori
- Worst-case complexity valid for a specific N
- (No closed form expression)

Comparison table

Stepsizes α_k	1 **unique method**	**Guarantee** $\forall N$	**Worst-case rate** $r(N)$ **Acceleration ?**
Classic $\frac{1}{L}$	✓	✓	$\frac{1}{4N+2}$
Optimal constant $\frac{h_{opt}}{L}$	X	✓	$\mathcal{O}(\frac{1}{8N})$

Outline

2. Teboulle-Vaisbourd increasing stepsizes

Introduced by Teboulle and Vaisbourd[5] in 2022 :

$$h_k \in [1, 2) \quad \forall k \in \mathbb{N}$$

Recurrence :

$$h_0 = \sqrt{2}$$

$$h_k = \frac{-LT_{k-1} + \sqrt{(LT_{k-1})^2 + 8(LT_{k-1} + 1)}}{2}$$

where $T_{k-1} = \sum_{i=0}^{k-1} \frac{h_i}{L}$

Advantage :

- No need to choose N in advance, guarantee for all $N \in \mathbb{N}$

Drawback :

- Rate $\frac{1}{2(2LT_{N-1}+1)} \gtrapprox \frac{1}{2Nh_{opt}+1}$ (numerically, due to recurrence formula. About 99%)

[5] Teboulle and Vaisbourd, *An elementary approach to tight worst case complexity analysis of gradient based methods*.

Comparison table

Stepsizes α_k	1 **unique method**	**Guarantee** $\forall N$	**Worst-case rate** $r(N)$ **Acceleration ?**
Classic $\frac{1}{L}$	✓	✓	$\frac{1}{4N+2}$
Optimal constant $\frac{h_{opt}}{L}$	X	✓	$\mathcal{O}(\frac{1}{8N})$
Non-constant T-V	✓	✓	$\frac{1}{2(2LT_{N-1}+1)}$

From the small-steps regime... to periodically taking longer steps

Small-steps regime :

- $h_k \in (0, 2)$
- Guaranteed $\{f(x_k)\}_{k\in\mathbb{N}}$ decreases at each iteration
- *Greedy/too shortsighted* approach ?

Could we do better by periodically taking larger steps ?
Can we improve long run guarantees ?

Periodically taking longer steps :

- Alternate
- Inspiration : Young (1953) on L-smooth, μ-strongly convex quadratics[6]

[6] Young, *On Richardson's Method for Solving Linear Systems with Positive Definite Matrices.*

From the small-steps regime...
to periodically taking longer steps

Intuition : bad cases for one are good cases for the other !

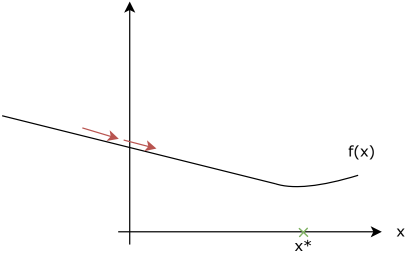


Figure: Bad case for small steps (ex.)

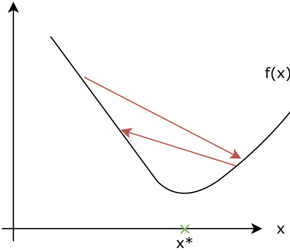


Figure: Bad case for large steps (ex.)

Outline

3. Das Gupta et al.'s steps

Numerically computed optimized sequence $\{h_k\}_{k=1}^{N}$ for each value of N

- Solving the minimization problem for specific N using PEP and Branch-and-Bound[7](on `MIT Supercloud` for $25 \leq N \leq 50$)

- Conjecture : faster than $\mathcal{O}(\frac{1}{N})$!
  Estimated :

$$r(N) = \frac{0.156}{N^{1.178}}$$

[7] Gupta, Parys, and Ryu, *Branch-and-Bound Performance Estimation Programming: A Unified Methodology for Constructing Optimal Optimization Methods*.

3. Das Gupta et al.'s steps

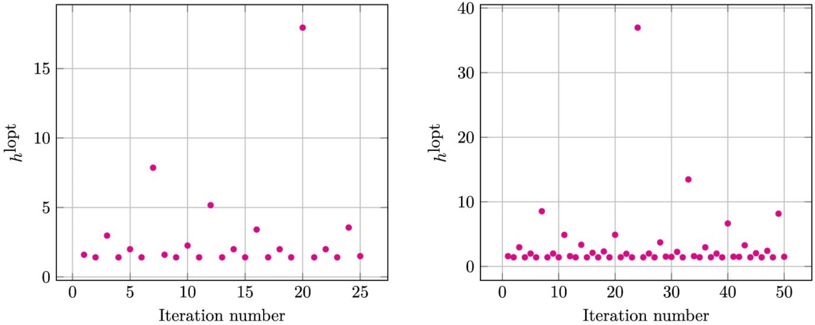


Figure: Das Gupta's steps for $N = 25$ and $N = 50$ resp[7].

[7] Gupta, Parys, and Ryu, *Branch-and-Bound Performance Estimation Programming: A Unified Methodology for Constructing Optimal Optimization Methods*.

Comparison table

Stepsizes α_k	1 **unique method**	**Guarantee** $\forall N$	**Worst-case rate** $r(N)$ **Acceleration ?**
Classic $\frac{1}{L}$	✓	✓	$\frac{1}{4N+2}$
Optimal constant $\frac{h_{opt}}{L}$	X	✓	$\mathcal{O}(\frac{1}{8N})$
Non-constant T-V	✓	✓	$\frac{1}{2(2LT_{N-1}+1)}$
Das Gupta's steps	X	✓	$0.156/N^{1.178}$ $(est.)$

Outline

4. Grimmer's patterns

Ben Grimmer
@prof_grimmer

I've proven the strangest result of my career..
The classic idea that gradient descent's rate is best with constant stepsizes 1/L is wrong. The idea that we need stepsizes in (0,2/L) for convergence is wrong.
Periodic long steps are better, provably.
arxiv.org/abs/2307.06324

Classic (Smooth+Convex) Gradient Descent Theory

Consider solving an L-smooth, convex minimization problem

$$p_\star = \min_{x \in \mathbb{R}^n} f(x)$$

by gradient descent with stepsizes $h = (h_0, h_1, h_2, \dots)$

$$x_{k+1} = x_k - \frac{h_k}{L} \nabla f(x_k).$$

The (previously best known) theory says to take constant stepsizes

$$h = (1, 1, 1 \dots) \text{ giving } f(x_T) - f(x_\star) \le \frac{LD^2}{2(T+1)}.$$

3:34 PM · Jul 14, 2023 · **677.5K** Views

4. Grimmer's patterns

Constructed *straightforward* patterns[8] $\{h_k\}_{k=0}^{t-1}$ of length t, and applied it periodically :

$$x_{k+1} = x_k - \frac{h_{(k \bmod t)}}{L} \nabla f(x_k)$$

For $N = st \quad s \in \mathbb{N}$,

$$f(x_N) - f(x^*) \leq \frac{L \left\| x_0 - x^* \right\|^2}{\operatorname{avg}(h) N} + \mathcal{O}\left(\frac{1}{N^2}\right)$$

Conjectured[9]:

$$\mathcal{O}\left(\frac{1}{N \log(N)}\right)$$

[8] Grimmer, *Provably Faster Gradient Descent via Long Steps*.

[9] Grimmer, Shu, and Wang, *Accelerated Gradient Descent via Long Steps*.

4. Grimmer's patterns

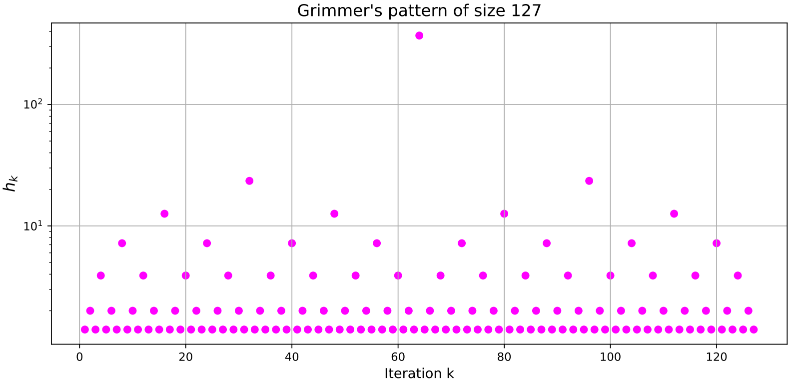


Figure: Straightforward[10] pattern of size 127, guaranteeing $\mathcal{O}\left(\frac{1}{5.8346303N}\right)$

[10] Grimmer, *Provably Faster Gradient Descent via Long Steps*.

Comparison table

Stepsizes α_k	1 **unique method**	**Guarantee** $\forall N$	**Worst-case rate** $r(N)$ **Acceleration ?**
Classic $\frac{1}{L}$	✓	✓	$\frac{1}{4N+2}$
Optimal constant $\frac{h_{opt}}{L}$	X	✓	$\mathcal{O}(\frac{1}{8N})$
Non-constant T-V	✓	✓	$\frac{1}{2(2LT_{N-1}+1)}$
Das Gupta's steps	X	✓	$0.156/N^{1.178}$ $(est.)$
Grimmer's patterns	✓	X	$\frac{1}{avg(h)N}$ (conj. $\frac{1}{N\log(N)}$)

Outline

5. Silver steps

September 2023 (shortly after Grimmer's publication), Altschuler and Parrilo[11]:

$$H_1 = \sqrt{2} = h_1$$
$$H_{2k+1} = [H_k, 1 + \rho^{k-1}, H_k] \quad \forall k \in \mathbb{N}_{++}$$

where $\rho = 1 + \sqrt{2}$, Silver ratio

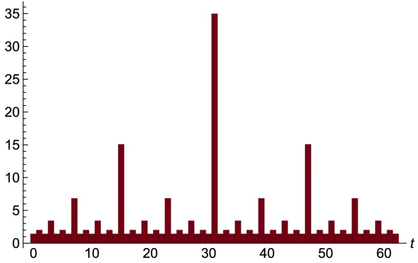


Figure: Silver schedule[13] of length 63

- H_k schedule of length k
- constructed recursively,
- non-monotonic,
- fractal-like,
- defined for $N = 2^k - 1$ (but no need to know N in advance).

[11] Altschuler and Parrilo, *Acceleration by Stepsize Hedging II: Silver Stepsize Schedule for Smooth Convex Optimization*.

5. Silver steps

Provable rate improvement[12]:

$$f(x_N)-f(x^*) \leq L\|x_0-x^*\|^2 \frac{1}{2N^{log_2(1+\sqrt{2})}} \quad \approx \mathcal{O}\left(1/N^{1.2716}\right)$$

But no guarantee for the iterates that are not of the form :

$$N = 2^k - 1 \quad k \in \mathbb{N}$$

Altschuler and Parrilo, *Acceleration by Stepsize Hedging II: Silver Stepsize Schedule for Smooth Convex Optimization*.

[13] Grimmer, Shu, and Wang, *Accelerated Gradient Descent via Long Steps*.

5. Silver steps

Provable rate improvement[12]:

$$f(x_N)-f(x^*) \leq L\|x_0-x^*\|^2 \frac{1}{2N^{log_2(1+\sqrt{2})}} \quad \approx \mathcal{O}\left(1/N^{1.2716}\right)$$

But no guarantee for the iterates that are not of the form :

$$N = 2^k - 1 \quad k \in \mathbb{N}$$

Altschuler and Parrilo, *Acceleration by Stepsize Hedging II: Silver Stepsize Schedule for Smooth Convex Optimization*.

At about the same time, Grimmer proved that by using non-constant, non-periodic stepsizes, he could achieve[13]: (somewhat weaker)

$$f(x_N) - f(x^*) \leq L\|x_0 - x^*\|^2 \frac{11.7816}{N^{1.0564}}$$

[13] Grimmer, Shu, and Wang, *Accelerated Gradient Descent via Long Steps*.

Comparison table

Stepsizes α_k	1 **unique method**	**Guarantee** $\forall N$	**Worst-case rate** $r(N)$ **Acceleration ?**
Classic $\frac{1}{L}$	✓	✓	$\frac{1}{4N+2}$
Optimal constant $\frac{h_{opt}}{L}$	X	✓	$\mathcal{O}(\frac{1}{8N})$
Non-constant T-V	✓	✓	$\frac{1}{2(2LT_{N-1}+1)}$
Das Gupta's steps	X	✓	$0.156/N^{1.178}$ $(est.)$
Grimmer's patterns	✓	X	$\frac{1}{avg(h)N}$ (conj. $\frac{1}{N\log(N)}$)
Silver steps	✓	X	$\mathcal{O}(1/N^{1.2716})$
Grimmer's NCNP	X	✓	$\mathcal{O}(1/N^{1.0564})$

Very recently... (26/11/2024)

There exists a stepsize schedule[14] providing anytime convergence ($\forall N \in \mathbb{N}$) s.t.

$$f(x_N) - f(x^*) \leq \mathcal{O}\left(\frac{\|x_0 - x^*\|^2}{N^\theta}\right)$$

where $\theta = \frac{7+\log_2 \rho}{8} > 1.03$

A lot weaker than Silver steps' $\mathcal{O}(1/N^{1.2716})$

[14] Zhang et al., *Anytime Acceleration of Gradient Descent*.

Even more recently (08/12/2024)

There exists a stepsize schedule[15] providing anytime convergence ($\forall N \in \mathbb{N}$) s.t.

$$f(x_N) - f(x^*) \leq \mathcal{O}\left(\frac{\|x_0 - x^*\|^2}{N^\theta}\right)$$

where $\theta = \frac{2\log_2 \rho}{1+\log_2 \rho} \approx 1.119$

[15] Zhang et al., *Anytime Acceleration of Gradient Descent V2*.

Comparison table

Stepsizes α_k	1 **unique method**	**Guarantee** $\forall N$	**Worst-case rate** $r(N)$ **Acceleration ?**
Classic $\frac{1}{L}$	✓	✓	$\mathcal{O}(1/N)$
Optimal constant $\frac{h_{opt}}{L}$	X	✓	$\mathcal{O}(1/N)$
Non-constant T-V	✓	✓	$\mathcal{O}(1/N)$
Das Gupta's steps	X	✓	$\mathcal{O}(1/N^{1.178})$ $(est.)$
Grimmer's patterns	✓	X	$\mathcal{O}(1/N)$ (conj. $\frac{1}{N\log(N)}$)
Silver steps	✓	X	$\mathcal{O}(1/N^{1.2716})$
Grimmer's NCNP	X	✓	$\mathcal{O}(1/N^{1.0564})$
Zhang's anytime	✓	✓	$\mathcal{O}(1/N^{1.119})$

Outline

6. Numerical experiment

Theoretical worst-case rates $r(N)$ as a function of N :

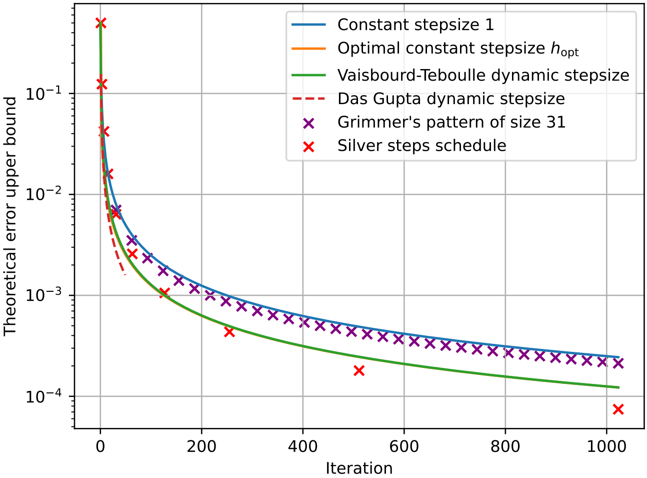


Figure: Theoretical rates for different stepsize schedules

6. Numerical experiment

Example on logistic regression problem ($X \in \mathbb{R}^{(455\times 23)}$) :

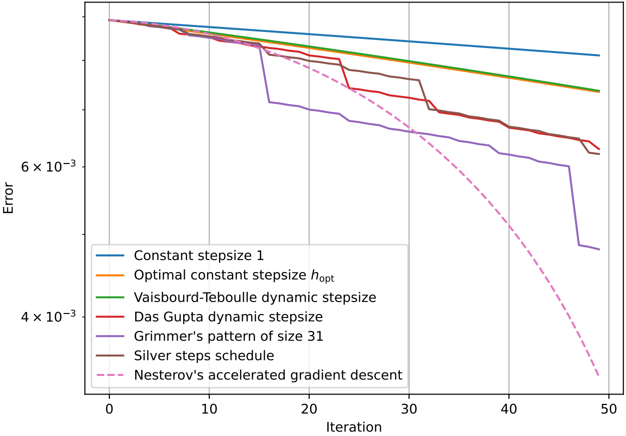


Figure: Evolution of the error $f(x_k) - f(x^*) \quad k \in \mathbb{N}$ on the logistic regression problem

6. Numerical experiment

Example on minimization of $\|Ax - b\|^2$ ($A \in \mathbb{R}^{(20\times 20)}$) :

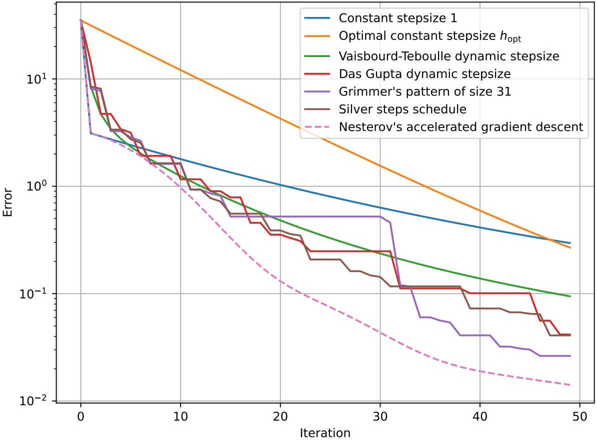


Figure: Evolution of the error $f(x_k) - f(x^*) \quad k \in \mathbb{N}$ on the linear system solving problem

Conclusion

Take-home message :

Longer steps in the Gradient Method can converge,
and can even **accelerate** convergence !

Open questions :

- Can we extend Silver steps rate to other values of N ?
- Can we improve $\mathcal{O}(1/N^{1.2716})$

  (reach Nesterov AGM's rate $\mathcal{O}(1/N^2)$?)

  or prove that it is optimal ?

Note : Keep in mind that we considered worst-case complexity.

Communication strategy

- Start with an introduction and a plan showing the direction of the presentation,
- Have a common thread (in this case, chronological),
- Summarize previous results for comparison, in a table,
- Present an example / numerical experiment,
- Conclude with open questions and research directions.

References I

Altschuler, Jason M. and Pablo A. Parrilo. *Acceleration by Stepsize Hedging II: Silver Stepsize Schedule for Smooth Convex Optimization*. 2023. arXiv: 2309.16530 [math.OC]. URL: https://arxiv.org/abs/2309.16530.

Drori, Yoel and Marc Teboulle. *Performance of first-order methods for smooth convex minimization: a novel approach*. 2012. arXiv: 1206.3209 [math.OC]. URL: https://arxiv.org/abs/1206.3209.

Glineur, François et al. *Performance Estimation of Optimization Methods: A Guided Tour*. Slides presented at the Workshop on Nonsmooth Optimization and Applications in Honor of the 75th Birthday of Boris Mordukhovich (NOPTA 2024). Antwerp, Belgium: University of Antwerp, 2024. URL: https:perso.uclouvain.be/francois.glineur/files/talks/NOPTA2024.pdf.

Grimmer, Benjamin. *Provably Faster Gradient Descent via Long Steps*. 2024. arXiv: 2307.06324 [math.OC]. URL: https://arxiv.org/abs/2307.06324.

Grimmer, Benjamin, Kevin Shu, and Alex L. Wang. *Accelerated Gradient Descent via Long Steps*. 2023. arXiv: 2309.09961 [math.OC]. URL: https://arxiv.org/abs/2309.09961.

Gupta, Shuvomoy Das, Bart P. G. Van Parys, and Ernest K. Ryu. *Branch-and-Bound Performance Estimation Programming: A Unified Methodology for Constructing Optimal Optimization Methods*. 2023. arXiv: 2203.07305 [math.OC]. URL: https://arxiv.org/abs/2203.07305.

Taylor, Adrien B., Julien M. Hendrickx, and François Glineur. *Smooth Strongly Convex Interpolation and Exact Worst-case Performance of First-order Methods*. 2016. arXiv: 1502.05666 [math.OC]. URL: https://arxiv.org/abs/1502.05666.

Teboulle, M. and Y. Vaisbourd. *An elementary approach to tight worst case complexity analysis of gradient based methods*. 2023. URL: https://doi.org/10.1007/s10107-022-01899-0.

References II

Vernimmen, Pierre. "Tight convergence analysis of exact and inexact gradient methods with constant and silver schedules". Master's thesis. UCLouvain, EPL, 2024.

Young, David. *On Richardson's Method for Solving Linear Systems with Positive Definite Matrices*. 1953. DOI: 10.1002/sapm1953321243. URL: https://doi.org/10.1002/sapm1953321243.

Zhang, Zihan et al. *Anytime Acceleration of Gradient Descent*. 2024. arXiv: 2411.17668 [cs.LG]. URL: https://arxiv.org/abs/2411.17668.

— .*Anytime Acceleration of Gradient Descent V2*. 2024. arXiv: 2411.17668 [cs.LG]. URL: https://arxiv.org/abs/2411.17668.

Graphs and small numerical experiment : https://github.com/SophieL1/LINMA2120-Longer-steps-in-GD

Questions and answers

Many methods exist beyond fixed gradient steps :

First order methods :

- GD with Armijo Line Search (if L unknown)
- Nesterov's Accelerated Gradient Method[16] (AGM) reaching $\mathcal{O}(N^{-2})$

Second order methods :

- Newton's method
- BFGS

[16] Nesterov, 1983

Questions and answers

Performance estimation of an optimization method[17] :

Find the worst-case instance

- of a given optimization problem (a given method), with a fixed number of iterations N,
- for function f belonging to a given class (convex and L-smooth),
- from any starting point x_0,
- and given a certain performance criteria (e.g. objective accuracy)

$$\max_{f, x_0, x_1, \ldots x_N} f(x_N) - f(x^*)$$

can be computed exactly using a semidefinite programming (SDP) problem.

[17] Glineur et al., *Performance Estimation of Optimization Methods: A Guided Tour*.

Questions and answers

Numerical examples :

1. Logistic regression

$$\min_{\theta \in \mathbb{R}^{n+1}} f(\theta) \equiv -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log \left(m_\theta \left(x^{(i)} \right) \right) + \left(1 - y^{(i)} \right) \log \left(1 - m_\theta \left(x^{(i)} \right) \right) \right]$$

where $m_\theta(x) = g\left(\theta^T x\right)$, with $g(z) = \frac{1}{1+e^{-z}}$.

Dataset : Wisconsin Breast Cancer dataset, training set of 455 samples, 23 features.
Additional trick : translated by $-\nabla f(x^*)^T x$ so that the gradient is 0 at x^*.

2. Minimizing $\|Ax - b\|^2$

with A and b randomly generated.

Questions and answers

Log-log graph and complexity :

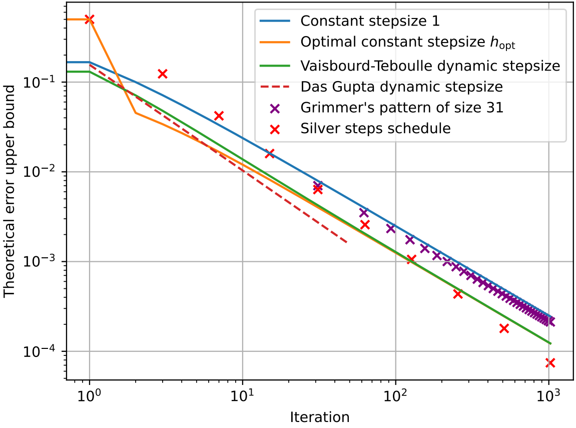


Figure: Log-log graph of theoretical rates

Questions and answers

Additional notes :

- In the comparison table, the **unique method** criteria asks if there exists a unique method for all N, meaning we do not need to choose N a priori ; or if the method is not unique and the schedule depends on the value of N.
- If there exists a unique method, the next column : **guarantee** $\forall N$, asks if we can stop the method at anytime and still have a garantee.
- If the method was not unique (i.e. designed based on the value of N), this second criteria asks if such a method can be designed for any value N.
- In the comparison table, red was used for the original GM rate, orange for an improvement by a constant factor, yellow for a conjectured improvement, and green for an improvement of the rate ($\mathcal{O}$ improvement).

Questions and answers

Additional notes (continued) :

- Some methods do not guarantee a decrease in the objective function gap at each iteration. On the graphs of the different methods applied to the examples, for these methods, the best previous iterate was taken. It leads to plateaus, but avoids going up.

Questions and answers

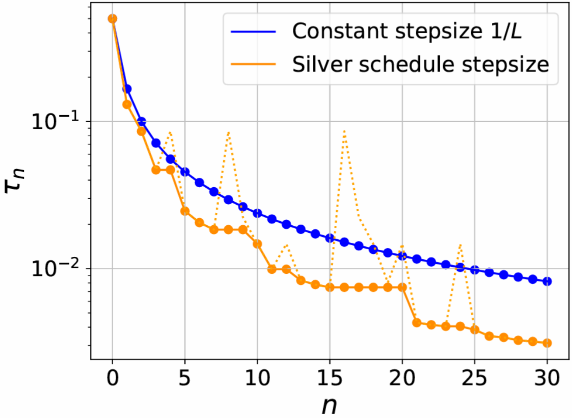


Figure: Comparison between classic GD and Silver steps[19]
Source : Pierre Vernimmen

[19] Vernimmen, "Tight convergence analysis of exact and inexact gradient methods with constant and silver schedules".