Worst-case analyses for first-order optimization methods

Adrien Taylor[*], Baptiste Goujaud[†]

Current version: July 1, 2022

Foreword & Acknowledgements

Those notes were written for accompanying the TraDE-OPT workshop on algorithmic and continuous optimization. If you have any comment, remark, or if you found a typo/mistake, please don't hesitate to feedback the authors!

Funding. A. Taylor acknowledges support from the European Research Council (grant SEQUOIA 724063). This work was partly funded by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). The work of B. Goujaud is partially supported by ANR-19-CHIA-0002-01/chaire SCAI, and Hi!Paris.

[*]INRIA, SIERRA project-team, and D.I. Ecole normale supérieure, Paris, France. Email: adrien.taylor@inria.fr
[†]CMAP, École Polytechnique, Institut Polytechnique de Paris, France. Email: baptiste.goujaud@gmail.com

1 Introduction

This document provides a series of exercises for getting familiar with "performance estimation problems" and the use of semidefinite programming for analyzing the worst-case behaviors of first-order optimization methods. An informal introduction can be found in this blog post.

In short, considering problems of the form $\min_x F(x)$ (we generally denote an optimal solution by $x_\star \in \operatorname{argmin}_x F(x)$), our goal is to access "a priori" the quality of the output (denoted by x_k) of some iterative algorithm. There are typically different ways of doing so, which might or might not be relevent depending on the target applications. In first-order optimization, we often want to upper bound the quality of x_k in one of the following terms (which we all ideally would like to be as small as possible): $\|x_k - x_\star\|^2$, $\|\nabla f(x_k)\|^2$, or $f(x_k) - f(x_\star)$. There are of course other possibilities.

So, our goal is to assess the quality of x_k by providing hopefully meaningfull upper bounds on (one of) those quantities. For doing so, we consider classes of problems (i.e., sets of assumptions on F), and perform worst-case analyses (i.e., we want the bound to be valid for all F satisfying the set of assumptions at hand).

After studying the performance estimation framework for optimization methods, one can realize that it has a broader applicability for performing worst-case studies in numerical analysis (see exercises in Section 3 and suggested readings in Section 5 for further information).

Notation and necessary background material is provided in Section 4.

2 Getting familial with performance estimation problems

Exercise 1 (Gradient method) *For this exercise, consider the problem of "black-box" minimization of a smooth strongly convex function:*

$$f_\star \triangleq \min_{x\in\mathbb{R}^d} f(x), \tag{1}$$

where f is L-smooth and μ-strongly convex (see Definition 2), and where $x_\star \triangleq \operatorname{argmin}_x f(x)$ and $f_\star \triangleq f(x_\star)$ its optimal value. For minimizing (1) we use gradient descent with a pre-determined sequence of step sizes $\{\gamma_k\}_k$; that is, we iterate $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$. The goal of this exercise is to compute $\tau(\mu, L, \gamma_k)$, a.k.a. a convergence rate, the smallest value such that the inequality

$$\|x_{k+1} - x_\star\|^2 \leqslant \tau(\mu, L, \gamma_k)\|x_k - x_\star\|^2$$

is valid for any $d \in \mathbb{N}$, for any L-smooth μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$) and for all $x_k, x_{k+1} \in \mathbb{R}^d$ such that $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$, and $x_\star = \operatorname{argmin}_x f(x)$.

1. *Show that*

$$\begin{aligned}\tau(\mu, L, \gamma_k) = \sup_{\substack{d,f\\ x_k, x_{k+1}, x_\star}} \quad & \frac{\|x_{k+1} - x_\star\|^2}{\|x_k - x_\star\|^2}\\ \text{s.t.} \quad & f \in \mathcal{F}_{\mu,L}\\ & x_{k+1} = x_k - \gamma_k \nabla f(x_k)\\ & \nabla f(x_\star) = 0,\end{aligned}$$

   where f, x_k, x_{k+1}, $x_\star$, and d are the variables and μ, L, γ are parameters.

   Note that we will (sometimes abusively) use max *instead of* sup *in the sequel as the optimum is usually attained for such problems (for this exercise, this is actually easy to show as the optimization problem is over a compact set).*

2. *Show that*

$$\tau(\mu, L, \gamma_k) = \max_{\substack{x_k, x_{k+1}, x_\star \\ g_k, g_\star \\ f_k, f_\star}} \frac{\|x_{k+1} - x_\star\|^2}{\|x_k - x_\star\|^2}$$

$$\text{s.t. } \exists f \in \mathcal{F}_{\mu,L} \text{ such that } \begin{cases} f_i = f(x_i) & i = k, \star \\ g_i = f'(x_i) & i = k, \star \end{cases}$$

$$x_{k+1} = x_k - \gamma_k g_k$$

$$g_\star = 0.$$

3. *Using Theorem 2, show that*

$$\begin{aligned}\tau(\mu, L, \gamma_k) = \max_{\substack{x_k, x_{k+1}, x_\star \\ g_k, g_\star \\ f_k, f_\star}} \quad & \frac{\|x_{k+1} - x_\star\|^2}{\|x_k - x_\star\|^2} \\ \text{s.t. } \quad & f_\star \geqslant f_k + \langle g_k, x_\star - x_k \rangle + \tfrac{1}{2L}\|g_\star - g_k\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_\star - x_k - \tfrac{1}{L}(g_\star - g_k)\right\|^2 \\ & f_k \geqslant f_\star + \langle g_\star, x_k - x_\star \rangle + \tfrac{1}{2L}\|g_k - g_\star\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_k - x_\star - \tfrac{1}{L}(g_k - g_\star)\right\|^2 \\ & x_{k+1} = x_k - \gamma_k g_k \\ & g_\star = 0.\end{aligned}$$

4. *Show that*

$$\begin{aligned}\tau(\mu, L, \gamma_k) = \max_{\substack{x_k, x_{k+1}, x_\star \\ g_k, g_\star \\ f_k, f_\star}} \quad & \|x_{k+1} - x_\star\|^2 \\ \text{s.t. } \quad & f_\star \geqslant f_k + \langle g_k, x_\star - x_k \rangle + \tfrac{1}{2L}\|g_\star - g_k\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_\star - x_k - \tfrac{1}{L}(g_\star - g_k)\right\|^2 \\ & f_k \geqslant f_\star + \langle g_\star, x_k - x_\star \rangle + \tfrac{1}{2L}\|g_k - g_\star\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_k - x_\star - \tfrac{1}{L}(g_k - g_\star)\right\|^2 \\ & \|x_k - x_\star\|^2 = 1 \\ & x_{k+1} = x_k - \gamma_k g_k \\ & g_\star = 0.\end{aligned}$$

5. *Define G and F*

$$G \triangleq \begin{bmatrix} \|x_k - x_\star\|^2 & \langle g_k, x_k - x_\star \rangle \\ \langle g_k, x_k - x_\star \rangle & \|g_k\|^2 \end{bmatrix}, \quad F \triangleq f_k - f_\star,$$

(note that $G = [x_k - x_\star \quad g_k]^\top [x_k - x_\star \quad g_k] \succcurlyeq 0$). Show that $\tau(\mu, L, \gamma_k)$ can be computed using the following 2×2 semidefinite program (SDP):

$$\begin{aligned}\tau(\mu, L, \gamma_k) = \max_{G, F} \quad & G_{1,1} + \gamma_k^2 G_{2,2} - 2\gamma_k G_{1,2} \\ \text{s.t.} \quad & F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{L}{L-\mu} G_{1,2} \leqslant 0 \\ & -F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{\mu}{L-\mu} G_{1,2} \leqslant 0 \\ & G_{1,1} = 1 \\ & G \succcurlyeq 0,\end{aligned} \tag{2}$$

6. *Define $h_k \triangleq \gamma_k L$ and $\kappa = L/\mu$. Show that $\tau(\mu, L, \gamma_k) = \tau(1/\kappa, 1, h_k)$ (in other words: we can study the case $L = 1$ only and deduce the dependence of τ on L afterwards).*

7. *Complete the PEPit code (alternative in Matlab: PESTO code) for computing $\tau(\mu, L, \gamma_k)$ and compute its value for a few numerical values of μ and γ_k.*

8. *Using Lagrangian duality with the following primal-dual pairing ($\tau, \lambda_1, \lambda_2$ are dual variables):*

$$\begin{aligned}
&F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{L}{L-\mu} G_{1,2} \leqslant 0 && : \lambda_1 \\
&-F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{\mu}{L-\mu} G_{1,2} \leqslant 0 && : \lambda_2 \\
&G_{1,1} = 1 && : \tau
\end{aligned}$$

one can show that

$$\begin{aligned}
\tau(\mu, L, \gamma_k) = \min_{\tau, \lambda_1, \lambda_2 \geqslant 0}\ & \tau \\
\text{s.t. } & S = \begin{bmatrix} \tau - 1 + \frac{\lambda_1 L\mu}{L-\mu} & \gamma_k - \frac{\lambda_1(\mu+L)}{2(L-\mu)} \\ \gamma_k - \frac{\lambda_1(\mu+L)}{2(L-\mu)} & \frac{\lambda_1}{L-\mu} - \gamma_k^2 \end{bmatrix} \succcurlyeq 0 \\
& 0 = \lambda_1 - \lambda_2.
\end{aligned} \tag{3}$$

Note that equility holds due to strong duality (for going further: obtain this dual formulation and prove strong duality using a Slater condition).

Show that any feasible point $(\tau, \lambda_1, \lambda_2)$ to (3) corresponds to an upper bound on $\tau(\mu, L, \gamma_k)$ (i.e., $\tau(\mu, L, \gamma_k) \leqslant \tau$).

9. *Is there a simple closed-form expression for $\tau(\mu, L, \gamma_k)$? Hint #1: can we solve (3) in closed-form? Hint #2: the objective is linear in τ; the optimal solution (if it exists) is therefore necessarily on the boundary of the PSD cone; hence τ must be such that at least one eigenvalue of S is zero.*

   Does it match the numerical values obtained using the previous codes for computing $\tau(\mu, L, \gamma_k)$ numerically?

10. *How can we adapt the SDP formulation (2) for computing the smallest possible τ such that the inequality*

$$\|\nabla f(x_{k+1})\|^2 \leqslant \tau \|\nabla f(x_k)\|^2$$

is valid for any $d \in \mathbb{N}$, for any L-smooth μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$) and for all $x_k, x_{k+1} \in \mathbb{R}^d$ such that $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$? Modify your previous code for computing such a bound. Can you guess a closed-form expression for it?

For going further: a dual problem is given by

$$\begin{aligned}
\min_{\tau, \lambda_1, \lambda_2 \geqslant 0}\ & \tau \\
\text{s.t. } & S = \begin{bmatrix} \tau + \lambda_1 \frac{(1-\gamma_k L)(1-\gamma_k \mu)}{L-\mu} & -\lambda_1 \frac{2-\gamma_k(L+\mu)}{2(L-\mu)} \\ -\lambda_1 \frac{2-\gamma_k(L+\mu)}{2(L-\mu)} & \frac{\lambda_1}{L-\mu} - 1 \end{bmatrix} \succcurlyeq 0 \\
& 0 = \lambda_1 - \lambda_2.
\end{aligned} \tag{4}$$

Is there a simple closed-form solution for this problem?

11. *How can we adapt the SDP formulation (2) for computing the smallest possible τ such that the inequality*

$$f(x_{k+1}) - f(x_\star) \leqslant \tau (f(x_k) - f(x_\star))$$

is valid for any $d \in \mathbb{N}$, for any L-smooth μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,L}$) and for all $x_k, x_{k+1} \in \mathbb{R}^d$ such that $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$? Modify your previous code for computing such a bound. Can you guess a closed-form expression for it?

For going further: using the following primal-dual pairing

$$\ldots,$$

a dual problem is given by

$$\ldots,$$

Is there a simple closed-form solution for this problem?

12. *Can we use this formalism for computing worst-case guarantees for a few iterations simultaneously? That is, to compute* $\tau(\mu, L, \{\gamma_k\}_{k=0,\ldots,N-1})$ *the smallest value such that the inequality*

$$\|x_N - x_\star\|^2 \leqslant \tau(\mu, L, \{\gamma_k\}_{k=0,\ldots,N-1})\|x_0 - x_\star\|^2$$

is valid for any $d \in \mathbb{N}$*, for any* L*-smooth* μ*-strongly convex function* f *(notation* $f \in \mathcal{F}_{\mu,L}$*) and for all* $x_0, x_1, \ldots, x_N \in \mathbb{R}^d$ *such that* $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$ *(*$k = 0, \ldots, N-1$*), and* $x_\star = \operatorname{argmin}_x f(x)$*.*

13. *What happens if* $\mu = 0$*? Can you isolate the problem on a simple counter example? You can, for example, use this PEPit code (alternative in Matlab: PESTO code). Can you imagine a solution for avoiding such pathological behaviors in the analyses? What about studying guarantees of type*

$$f(x_N) - f_\star \leqslant \tau(\mu, L, \{\gamma_k\}_{k=0,\ldots,N-1})\|x_0 - x_\star\|^2$$

instead? Modify your code for studying such worst-case bounds, and try it numerically for the choice $\gamma_k = 1/L$*,* $L = 1$ *and* $\mu = 0$*. Guess the depence on* N *based on a few numerical trials.*

14. *Based on your current experience, what are, according to you, the key elements which allowed casting the worst-case analysis as an SDP?*

15. *Can you write a standard proof for the linear convergence in terms of distance to an optimal point? in terms of convergence in gradient norm? and function values?*

3 Further exercises

Exercise 2 (Sublinear convergence of gradient descent and acceleration) *Show that the smallest* τ *such that the inequality*

...

1. *show that it can be formulated as an SDP.*
2. *numerical trials (code: XXX).*
3. *Can we compute guarantees of type*

$$\min_{0 \leqslant i \leqslant N} \|\nabla f(x_i)\|^2 \leqslant \tau \|x_0 - x_\star\|^2$$

using semidefinite programming?

4. *Modify your code for computing the worst-case ratios* $\frac{\min_{0\leqslant i\leqslant N}\|\nabla f(x_i)\|^2}{\|x_0 - x_\star\|^2}$ *and* $\frac{\|\nabla f(x_N)\|^2}{\|x_0 - x_\star\|^2}$ *as functions of* N*. What can you conclude?*
5. *Modify your code for computing worst-case guarantees for the following variant of Nesterov's accelerated gradient method:*

$$XXX$$

in terms of the same ratios, and compare them to those of gradient descent (as functions of N*). What can you conclude?*

Exercise 3 (Subgradient method) *Show that the smallest* τ *such that the inequality*

...

1. *show that it can be formulated as ...*
2. *show that the previous problem can be framed using a discrete version...*
3. *numerical trials (code: XXX). Ex: modify the code to compute worst gradient norm, and worst best gradient norm among iterates*

Exercise 4 (Acceleration and Lyapunov analyses) *Show that the smallest τ such that the inequality*

...

1. *show that it can be formulated as ...*
2. *show that the previous problem can be framed using a discrete version...*
3. *numerical trials*

Exercise 5 (Fixed-point iterations) *Show that the smallest τ such that the inequality*

...

1. *two methods: Halpern and Kras...*
2. *show that it can be formulated as ...*
3. *show that the previous problem can be framed using a discrete version...*
4. *show that it is equivalent to the SDP XXXX*
5. *using duality show that ... (dual SDP)*
6. *numerical trials*

Exercise 6 (Stochastic gradient descent) *Show that the smallest τ such that the inequality*

...

1. *show that it can be formulated as ...*
2. *show that the previous problem can be framed using a discrete version...*
3. *show that it is equivalent to the SDP XXXX*
4. *using duality show that ... (dual SDP)*
5. *numerical trials*

Exercise 7 (Proximal point method) *Show that the smallest τ such that the inequality*

...

1. *show that it can be formulated as ...*
2. *show that the previous problem can be framed using a discrete version...*
3. *show that it is equivalent to the SDP XXXX*
4. *numerical trials*

Exercise 8 (Proximal gradient method) *Show that the smallest τ such that the inequality*

...

1. *show that it can be formulated as ...*
2. *show that the previous problem can be framed using a discrete version...*
3. *show that it is equivalent to the SDP XXXX*
4. *numerical trials*

Exercise 9 (Douglas-Rachford splitting) *Show that the smallest τ such that the inequality*

...

1. *show that it can be formulated as ...*
2. *show that the previous problem can be framed using a discrete version...*
3. *show that it is equivalent to the SDP XXXX*
4. *numerical trials*

Exercise 10 (Frank-Wolfe) *Show that the smallest τ such that the inequality*

...

1. *show that it can be formulated as ...*
2. *show that the previous problem can be framed using a discrete version...*
3. *show that it is equivalent to the SDP XXXX*
4. *numerical trials*

Exercise 11 (Alternate projections & Dykstra) *Show that the smallest τ such that the inequality*

...

1. *show that it can be formulated as ...*
2. *show that the previous problem can be framed using a discrete version...*
3. *show that it is equivalent to the SDP XXXX*
4. *numerical trials*

4 Background material and useful facts

4.1 Standard definitions

smoothness, strong convexity...

Definition 1 *cpp*

Definition 2 *sm str cvx (notation...)*

Definition 3 *lip cvx*

Definition 4 *lip cvx*

4.2 Interpolation/extension theorems

This section gathers useful elements allowing to answer certain questions ...

Theorem 1 *interpolation 1... (ccp)*

Theorem 2 *interpolation 2... (smooth str convex)*

Theorem 3 *interpolation 3... (smooth nonconvex)*

Theorem 4 *nonsmooth*

Theorem 5 *indicator*

4.3 Other useful inequalities

4.4 SDP duality

++primal and dual SDP formulations

Theorem 6 *slater*

5 Going further - suggested readings

Designing methods.

Adaptive methods.

Primal-dual methods. Ernest'

Mirror descent. Radu's

Identifying lower complexity bounds. QG, Radu's

Continuous-time analyses.

Identifying counter-examples

Other analyses.