Performance Estimation Problem

Introduction

Zhenghao Xu

College of Computer Science and Technology
Zhejiang University

November 12, 2021

Table of Contents

Table of Contents

Performance Estimation Problem

Performance Estimation Problem (PEP) is a worst-case performance analysis approach first introduced by [Drori and Teboulle, 2014].

It was originally developed to analyze the exact worst-case bound of Gradient Method (GM) for smooth convex functions.

It formulates the worst-case performance of a problem-method pair as an optimization problem (PEP) and turns it into a semidefinite programming (SDP) form that we can solve.

Application

PEP approach allows us to:

1. solve the PEP for a (tight) lower bound example,
2. solve the dual of PEP to get (tight) upper bound,
3. solve the minimax problem of PEP to get acceleration,
4. verify / help design Lyapunov function.

Brief History

The framework of PEP was mainly developed by three groups of people:

- [Drori and Teboulle, 2014] first brought up the framework and get a better upper bound for GM by factor of 2.
- [Kim and Fessler, 2016, Kim and Fessler, 2017] derived Optimized Gradient Method (OGM) from PEP that performs better than Nesterov's by factor of 2.
- [Taylor et al., 2017a, Taylor et al., 2018b] provided convex interpolation conditions for smooth strongly convex function class, guarantees tightness of SDP formulation.

Capability

Then this approach then extends to other settings.

- different performance metrics: $f(x_N) - f_*$, $\|\nabla f(x_N)\|$, $\min_i \|x_i - x_{i-1}\|$, etc.;
- different objective function classes: smooth/nonsmooth, convex/strongly convex, convex composite, relatively smooth, smooth adaptable, etc.;
- different methods: gradient, proximal gradient, proximal point, mirror descent, Bregman proximal gradient, Bregman proximal point, stochastic gradient, decentralized gradient, etc.;

Capability

- Acceleration: [d'Aspremont et al., 2021].
- Proximal: [Kim and Fessler, 2018, Taylor et al., 2018a].
- Non-Euclidean (Bregman): [Dragomir et al., 2019].
- Stochastic: [Taylor and Bach, 2019].
- Monotone inclusion: [Ryu et al., 2020].
- Fixed-point iteration: [Lieder, 2021].
- Decentralized: [Colla and Hendrickx, 2021].
- Polyak step: [Barr'e et al., 2020].
- A toolbox of PEP: [Taylor et al., 2017b].

Table of Contents

Setup

We illustrate PEP approach for convex, smooth minimization with fixed step gradient method (GM).
[Drori and Teboulle, 2014, Taylor et al., 2017a]

We consider the following minimization problem in the class of convex smooth functions.

$$\min_{x \in \mathbb{R}^d} \quad f(x) \tag{P}$$

where $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$ is L-smooth convex differentiable function:

$$f(x) + \langle \nabla f(x), y - x \rangle \leq f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + L\|y - x\|^2.$$

We use the GM to minimize (P):

$$x_{k+1} = x_k - \lambda \nabla f(x_k). \tag{GM}$$

PEP formulation

To get the exact worst-case performance of (P) with (GM) after N iterations we formulate the PEP:

$$\max \quad f(x_N) - f_* \tag{PEP}$$

$$s.t. \quad \begin{cases} f \in \mathcal{F}_{0,L}(\mathbb{R}^d), & \text{(function class)} \\ x_{i+1} = x_i - \lambda \nabla f(x_i), & \text{(method)} \\ \nabla f(x_*) = 0, & \text{(optimality)} \\ \|x_* - x_0\| \leq R. & \text{(bounded initial)} \end{cases}$$

Above formulation is impractical to solve for f is infinite dimensional.
We need to find a way to characterize function f only through its oracle at iterates.

PEP Discretization

$$
\begin{cases}
f \in \mathcal{F}_{0,L}(\mathbb{R}^d), \\
x_{i+1} = x_i - \lambda \nabla f(x_i), \\
\nabla f(x_*) = 0, \\
\|x_* - x_0\| \le R.
\end{cases}
$$

$$\Longleftrightarrow$$

$$
\begin{cases}
f_i \in \mathbb{R}, g_i, x_i \in \mathbb{R}^d, \\
\exists f \in \mathcal{F}_{0,L}(\mathbb{R}^d) \text{ such that } f(x_i) = f_i, \nabla f(x_i) = g_i, \quad \text{(interpolation)} \\
x_{i+1} = x_i - \lambda g_i, \\
g_* = 0, \\
\|x_* - x_0\| \le R.
\end{cases}
$$

Convex Interpolation

Theorem (Smooth strongly convex interpolation, [Taylor et al., 2017a])

For $0 \le \mu < L \le \infty$, $f_i, g_i, x_i \in \mathbb{R}^d$, I *is an index set,*

$$\exists f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d), \forall i \in I, \quad f_i = f(x_i), g_i \in \partial f(x_i)$$

if and only if for any $i, j \in I$,

$$f_i - f_j - \langle g_j, x_i - x_j \rangle \ge \frac{1}{2(1-\mu/L)} \left(\frac{1}{L}\|g_i - g_j\|^2 + \mu\|x_i - x_j\|^2 - 2\frac{\mu}{L}\langle g_i - g_j, x_i - x_j \rangle \right).$$

With this interpolation theorem, we may replace the function class constraints.

Convex Interpolation

$$
\begin{cases}
f_i \in \mathbb{R}, g_i, x_i \in \mathbb{R}^d, \\
\exists f \in \mathcal{F}_{0,L}(\mathbb{R}^d) \text{ such that } f(x_i) = f_i, \nabla f(x_i) = g_i, \quad \text{(interpolation)} \\
x_{i+1} = x_i - \lambda g_i, \\
g_* = 0, \\
\|x_* - x_0\| \le R.
\end{cases}
$$

$$\Longleftrightarrow$$

$$
\begin{cases}
f_i \in \mathbb{R}, g_i, x_i \in \mathbb{R}^d, \\
f_i - f_j - \langle g_j, x_i - x_j \rangle \ge \frac{1}{2L}\|g_i - g_j\|^2, & \forall i, j \in I, \\
x_{i+1} = x_i - \lambda g_i, & \forall i = 0, \dots, N-1, \\
g_* = 0, \\
\|x_* - x_0\| \le R.
\end{cases}
$$

PEP Semidefinite Form

This formulation is non-convex for its inner product terms.
We may turn this into a semidefinite programming (SDP).

Let $G = P^\top P$, $P = \left(P_x, P_g\right) = (\underbrace{x_*, x_0, \ldots, x_N}_{P_x}, \underbrace{g_*, g_0, \ldots, g_N}_{P_g})$.

Denote $G^{\alpha,\beta} = P_\alpha^\top P_\beta$, $\alpha, \beta \in \{x, g\}$.
Then $G_{i,j}^{\alpha,\beta} = \langle \alpha_i, \beta_j \rangle$, $i, j \in I = \{*, 0, \ldots, N\}$.

Thus all the inner product terms are linear in element of G, constraints become linear.
With additional $G \succeq 0$ the PEP becomes a SDP.

PEP Semidefinite Form

$$
\begin{cases}
f_i \in \mathbb{R}, g_i, x_i \in \mathbb{R}^d, \\
f_i - f_j - \langle g_j, x_i - x_j \rangle \geq \frac{1}{2L} \|g_i - g_j\|^2, & \forall i, j \in I, \\
x_{i+1} = x_i - \lambda g_i, & \forall i = 0, \dots, N-1, \\
g_* = 0, \\
\|x_* - x_0\| \leq R.
\end{cases}
$$

$$\Longleftrightarrow$$

$$
\begin{cases}
f_i \in \mathbb{R}, G \in \mathbb{R}^{(2N+4)\times(2N+4)} \\
f_i - f_j - G^{g,x}_{j,i} + G^{g,x}_{j,j} \geq \frac{1}{2L} \left(G^{g,g}_{i,i} - 2G^{g,g}_{i,j} + G^{g,g}_{j,j} \right), \quad \forall i, j \in I, \\
G^{g,x}_{j,i+1} = G^{g,x}_{j,i} - \lambda G^{g,g}_{j,i}, \quad \forall j \in I, i = 0, \dots, N-1, \\
G^{g,g}_{*,*} = 0, \\
G^{x,x}_{*,*} - 2G^{x,x}_{*,0} + G^{x,x}_{0,0} \leq R^2, \\
G \succeq 0.
\end{cases}
$$

PEP Semidefinite Form

$$
\max_{\substack{f_*, f_0, \ldots, f_N \in \mathbb{R} \\ G \in \mathbb{R}^{(2N+4)\times(2N+4)}}} \quad f_N - f_* \qquad \text{(sdp-PEP)}
$$

$$
s.t. \quad \begin{cases}
f_i - f_j - G^{g,x}_{j,i} + G^{g,x}_{j,j} \geq \frac{1}{2L}\left(G^{g,g}_{i,i} - 2G^{g,g}_{i,j} + G^{g,g}_{j,j}\right), \quad \forall i, j \in I, \\
G^{g,x}_{j,i+1} = G^{g,x}_{j,i} - \lambda G^{g,g}_{j,i}, \quad \forall j \in I, i = 0, \ldots, N-1, \\
G^{g,g}_{*,*} = 0, \\
G^{x,x}_{*,*} - 2G^{x,x}_{*,0} + G^{x,x}_{0,0} \leq R^2, \\
G \succeq 0.
\end{cases}
$$

As long as an interior solution exists, this problem has strong duality.

Table of Contents

Solve PEP

If strong duality holds, we have two paths:

- directly solve (sdp-PEP),
- or, relax it and solve its dual problem.

The first approach is easy for numerical solution. We can easily get lower bound instance following this path. But for (analytical) dual solution, we have to guess through observation.

The latter one is difficult to tackle when problem and method given are complex. But in some cases this can be done and we get analytical solution.

Approach I: Numerically Inspired Proof

To get a proof for tight upper bound, we can follow these steps:

1. solve (sdp-PEP) numerically,
2. guess analytical dual solutions associated with inequalities,
3. sum up these inequalities with dual vars as weights,
4. substitute variables with equality constraints,
5. verify the result using computer algebra system like Mathematica.

The obtained proof is completely arithmetic.

Here the PEP serves as a scaffolding: the proof does not really require solving PEP, which just leads us to find this arithmetic proof lack of intuition.

We can check the feasibility of guessed dual "solution" through checking the dual constraints. (Same as Approach II.)

An example from Bregman:

Theorem 1 (BPPSG convergence rate) *Let $L > 0$, (f, h, ϕ) are a tuple of function satistying Assumption 1. Then the sequence $\{y_k\}_{k\geq 0}$ generated by Algorithm 1 with constant step size $\lambda \in (0, 1/L]$ satisfies for all $N > 0$,*

$$f(y_N) - f(x_*) \leq \frac{D_h(x_*, x_0)}{\lambda(2N+1)}. \tag{6}$$

Proof We make use of the convexity inequalities in (r-PEP-P) and perform weighted sum of these inequalities, where weights are given by the corresponding dual solutions of (r-PEP-P).

The proof itself is purely arithmetic. We use the notations defined in the last section. The nonzero dual solution and corresponding constraint inequalities involved in the proof:

- $c^{(1)}_{N,i} = \frac{2}{2N+1}, (i = 1, \ldots, N+1)$ for convexity of f between u_* and u_i:
$$f_i - f_* + \langle g_i, u_* - u_i\rangle \leq 0,$$
- $c^{(2)}_{N,i} = \frac{2i-1}{2N+1}, (i = 1, \ldots, N)$ for convexity of f between u_i and u_{i+1}:
$$f_{i+1} - f_i + \langle g_{i+1}, u_i - u_{i+1}\rangle \leq 0,$$
- $c^{(3)}_{N,N} = \frac{1}{2N+1}$ for convexity of f between u_1 and u_0
$$f_0 - f_1 + \langle g_0, u_1 - u_0\rangle \leq 0,$$
- $c^{(4)}_{N,i} = \frac{2i-1}{4N+2}, (i = 2, \ldots, N+1)$ for convexity of $\frac{1}{\lambda}h - f$ between u_i and u_{i-1}:
$$\frac{1}{\lambda}h_{i-1} - f_{i-1} - (\frac{1}{\lambda}h_i - f_i) + \left\langle \frac{1}{\lambda}s_{i-1} - g_{i-1}, u_i - u_{i-1}\right\rangle \leq 0,$$
- $c^{(5)}_{N,i} = \frac{2i-1}{4N+2}, (i = 1, \ldots, N)$ for convexity of $\frac{1}{\lambda}h - f$ between u_i and u_{i+1}:
$$\frac{1}{\lambda}h_{i+1} - f_{i+1} - (\frac{1}{\lambda}h_i - f_i) + \left\langle \frac{1}{\lambda}s_{i+1} - g_{i+1}, u_i - u_{i+1}\right\rangle \leq 0,$$

The weighted sum is

$$\begin{aligned}
&\sum_{i=1}^{N+1} \frac{2}{2N+1}\left(f_i - f_* + \langle g_i, u_* - u_i\rangle\right) \\
&+ \sum_{i=1}^{N} \frac{2i-1}{2N+1}\left(f_{i+1} - f_i + \langle g_{i+1}, u_i - u_{i+1}\rangle\right) \\
&+ \frac{1}{2N+1}\left(f_0 - f_1 + \langle g_0, u_1 - u_0\rangle\right) \\
&+ \sum_{i=2}^{N+1} \frac{2i-1}{4N+2}\left(\frac{1}{\lambda}h_{i-1} - f_{i-1} - (\frac{1}{\lambda}h_i - f_i) + \left\langle \frac{1}{\lambda}s_{i-1} - g_{i-1}, u_i - u_{i-1}\right\rangle\right) \\
&+ \sum_{i=1}^{N} \frac{2i-1}{4N+2}\left(\frac{1}{\lambda}h_{i+1} - f_{i+1} - (\frac{1}{\lambda}h_i - f_i) + \left\langle \frac{1}{\lambda}s_{i+1} - g_{i+1}, u_i - u_{i+1}\right\rangle\right) \\
&+ \frac{1}{2N+1}\left(\frac{1}{\lambda}h_0 - f_0 - (\frac{1}{\lambda}h_1 - f_1) + \left\langle \frac{1}{\lambda}s_0 - g_0, u_1 - u_0\right\rangle\right) \\
&+ \frac{1}{2N+1}\left(\frac{1}{\lambda}h_{N+1} - f_{N+1} - (\frac{1}{\lambda}h_* - f_*) + \left\langle \frac{1}{\lambda}s_{N+1} - g_{N+1}, u_* - u_{N+1}\right\rangle\right) \\
=&f_{N+1} - f_* - \frac{1}{\lambda(2N+1)}(h_* - h_0) + \frac{1}{2N+1}\left\langle \frac{1}{\lambda}s_0, -u_0\right\rangle \\
&+ \underbrace{\left\langle \sum_{i=1}^{N+1} \frac{2}{2N+1}g_i + \frac{1}{2N+1}\left(\frac{1}{\lambda}s_{N+1} - g_{N+1}\right), u_*\right\rangle}_{I} \\
&+ \underbrace{\sum_{i=2}^{N} \frac{2i-1}{4N+2}\left\langle \frac{1}{\lambda}s_{i-1} - g_{i-1} + \frac{1}{\lambda}s_{i+1} + g_{i+1} - \frac{2}{\lambda}s_i, u_i\right\rangle}_{II} \\
&+ \frac{1}{4N+2}\left\langle \frac{1}{\lambda}s_2 + g_2 - \frac{3}{\lambda}s_1 - g_1 + \frac{2}{\lambda}s_0, u_1\right\rangle \\
&+ \frac{1}{2}\left\langle \frac{1}{\lambda}s_N - g_N - \frac{1}{\lambda}s_{N+1} - g_{N+1}, u_{N+1}\right\rangle \\
:=&E \leq 0.
\end{aligned}$$

An example from Bregman:

By equation (5), written with new notations as

$$\begin{cases} s_1 = s_0 - \lambda g_1, \\ s_{k+1} = s_k - \lambda\left(g_k + g_{k+1}\right) \quad k = 1, \ldots, N, \end{cases}$$

we have

$$\frac{1}{\lambda} s_{i-1} - g_{i-1} + \frac{1}{\lambda} s_{i+1} + g_{i+1} - \frac{2}{\lambda} s_i = 0, \quad \forall 2 \leq i \leq N, \tag{7}$$

$$s_{N+1} + \lambda g_{N+1} + 2\lambda \sum_{i=1}^{N} g_i = s_0. \tag{8}$$

Thus $II = 0$ and

$$I = \frac{1}{\lambda(2N+1)} \langle s_0, u_* \rangle .$$

The weighted sum of inequalities becomes

$$\begin{aligned} E =& f_{N+1} - f_* - \frac{1}{\lambda(2N+1)}(h_* - h_0) \\ &+ \frac{1}{\lambda(2N+1)} \langle s_0, u_* - u_0 \rangle \\ &+ \frac{1}{4N+2} \left\langle -\frac{2}{\lambda} s_1 - 2g_1 + \frac{2}{\lambda} s_0, u_1 \right\rangle \\ &+ \frac{1}{2} \left\langle \frac{1}{\lambda} s_N - g_N - \frac{1}{\lambda} s_{N+1} - g_{N+1}, u_{N+1} \right\rangle \\ =& f_{N+1} - f_* - \frac{1}{\lambda(2N+1)} D_h(u_*, u_0) \leq 0. \end{aligned}$$

Replace the symbol back and we obtain the convergence result

$$f(y_N) - f(x_*) \leq \frac{1}{\lambda(2N+1)} D_h(x_*, x_0). \tag{9}$$

Approach I: Numerically Inspired Proof

Here the inequalities seem "loosely combined".

It turns out that following the spirit of PEP, we may design Lyapunov functions all inspired from numerical result and then verified by combination of inequalities like above.
The formulation goes like:

$$\max \quad \mathcal{V}_{k+1} - \mathcal{V}_k \le 0$$
$$s.t. \quad \begin{cases} \text{interpolation condition between } x_k,\ x_{k+1} \text{ and } x_*, \\ \text{step between } x_k \text{ and } x_{k+1}, \end{cases}$$

where $\mathcal{V}_k$ is the Lyapunov function at k-th iteration.
See `https://www.di.ens.fr/~ataylor/share/Slides_CWIinria2020.pdf`.

Approach II: Dual Analytical Solution

Example from [Drori and Teboulle, 2014]:

Relaxed PEP:

$$
\begin{aligned}
\max_{G \in \mathbb{R}^{(N+1)\times d}, \delta \in \mathbb{R}^{N+1}} \quad & LR^2 \delta_N \\
\text{s.t.} \quad & \operatorname{tr}(G^T A_{i-1,i} G) \leq \delta_{i-1} - \delta_i, \quad i = 1, \dots, N, \\
& \operatorname{tr}(G^T D_i G + \nu u_i^T G) \leq -\delta_i, \quad i = 0, \dots, N.
\end{aligned}
\tag{G$'$}
$$

Corresponding dual problem:

$$
\min_{\lambda \in \mathbb{R}^N, t \in \mathbb{R}} \{ \frac{1}{2} LR^2 t : \ \lambda \in \Lambda, \ S(\lambda, t) \succeq 0 \}, \tag{DG$'$}
$$

$$
S(\lambda, t) = \begin{pmatrix} (1-h)S_0 + hS_1 & q \\ q^T & t \end{pmatrix}, \qquad
S_0 = \begin{pmatrix} 2\lambda_1 & -\lambda_1 & & & & \\ -\lambda_1 & 2\lambda_2 & -\lambda_2 & & & \\ & -\lambda_2 & 2\lambda_3 & -\lambda_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -\lambda_{N-1} & 2\lambda_N & -\lambda_N \\ & & & & -\lambda_N & 1 \end{pmatrix}
$$

$$
S_1 = \begin{pmatrix} 2\lambda_1 & \lambda_2 - \lambda_1 & \dots & \lambda_N - \lambda_{N-1} & 1 - \lambda_N \\ \lambda_2 - \lambda_1 & 2\lambda_2 & & \lambda_N - \lambda_{N-1} & 1 - \lambda_N \\ \vdots & & \ddots & & \vdots \\ \lambda_N - \lambda_{N-1} & \lambda_N - \lambda_{N-1} & & 2\lambda_N & 1 - \lambda_N \\ 1 - \lambda_N & 1 - \lambda_N & \dots & 1 - \lambda_N & 1 \end{pmatrix}.
$$

Approach II: Dual Analytical Solution

Example from [Drori and Teboulle, 2014]:

Let $\lambda_i = \frac{i}{2N+1-i}$, $i = 1, \ldots, N$, $t = \frac{1}{2Nh+1}$.
Then $\lambda = (\lambda_1, \ldots, \lambda_N) \in \Lambda$, $S(\lambda, t) \succeq 0$ is a feasible solution to (DG')

Theorem 3.1. *Let $f \in C_L^{1,1}(\mathbb{R}^d)$ and let $x_0, \ldots, x_N \in \mathbb{R}^d$ be generated by Algorithm GM with $0 < h \leq 1$. Then*

$$f(x_N) - f(x_*) \leq \frac{LR^2}{4Nh+2}. \tag{3.11}$$

Theorem 3.2. *Let $L > 0$, $N \in \mathbb{N}$ and $d \in \mathbb{N}$. Then for every $h > 0$ there exists a convex function $\varphi \in C_L^{1,1}(\mathbb{R}^d)$ and a point $x_0 \in \mathbb{R}^d$ such that after N iterations, Algorithm GM reaches an approximate solution x_N with following absolute inaccuracy*

$$\varphi(x_N) - \varphi^* = \frac{LR^2}{2} \max\left(\frac{1}{2Nh+1}, (1-h)^{2N}\right).$$

Table of Contents

Optimized Method

We have seen that PEP can:

- give lower bound instance (every numerical solution corresponds to a instance),
- induce upper bound proof,
- verify Lyapunov function.

We can obtain methods with faster convergence rate by performing minimax optimization on PEP.

In the following of this section we show example of Optimized Gradient Method (OGM) from [Kim and Fessler, 2016].

General Fixed Step FO Method

Algorithm Class FO

Input: $f \in \mathcal{F}_L(\mathbb{R}^d)$, $\boldsymbol{x}_0 \in \mathbb{R}^d$.

For $i = 0, \ldots, N-1$

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \frac{1}{L} \sum_{k=0}^{i} h_{i+1,k} f'(\boldsymbol{x}_k). \tag{1.1}$$

PEP for FO:

$$\mathcal{B}_{\mathrm{P}}(\boldsymbol{h}, N, d, L, R) \triangleq \max_{f \in \mathcal{F}_L(\mathbb{R}^d)} \max_{\substack{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d, \\ \boldsymbol{x}_* \in X_*(f)}} f(\boldsymbol{x}_N) - f(\boldsymbol{x}_*) \tag{P}$$

$$\text{s.t. } \boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \frac{1}{L}\sum_{k=0}^{i} h_{i+1,k} f'(\boldsymbol{x}_k), \quad i = 0, \ldots, N-1,$$

$$\|\boldsymbol{x}_0 - \boldsymbol{x}_*\| \le R.$$

$$\mathcal{B}_{\mathrm{D}}(\boldsymbol{h}, N, L, R) \triangleq \min_{\substack{\boldsymbol{\lambda} \in \mathbb{R}^N, \\ \boldsymbol{\tau} \in \mathbb{R}^{N+1}, \\ \gamma \in \mathbb{R}}} \left\{ \frac{1}{2} L R^2 \gamma : \begin{pmatrix} S(\boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{\tau}) & \frac{1}{2}\boldsymbol{\tau} \\ \frac{1}{2}\boldsymbol{\tau}^\top & \frac{1}{2}\gamma \end{pmatrix} \succeq 0, \quad (\boldsymbol{\lambda}, \boldsymbol{\tau}) \in \Lambda \right\}, \tag{D}$$

$$\Lambda = \left\{ (\boldsymbol{\lambda}, \boldsymbol{\tau}) \in \mathbb{R}_+^N \times \mathbb{R}_+^{N+1} : \begin{array}{l} \tau_0 = \lambda_1, \ \lambda_N + \tau_N = 1 \\ \lambda_i - \lambda_{i+1} + \tau_i = 0, \ i = 1, \ldots, N-1 \end{array} \right\}, \tag{4.3}$$

Analytical OGM

Following the work [Drori and Teboulle, 2014], to find the optimized steps h can be reduced from finding solution of the primal problem to the dual problem.

$$\hat{\boldsymbol{h}} \triangleq \mathop{\arg\min}_{\boldsymbol{h} \in \mathbb{R}^{N(N+1)/2}} \mathcal{B}_{\mathrm{D}}(\boldsymbol{h}, N, L, R), \tag{HD}$$

Convert to new variables:

$$r_{i,k} = \lambda_i h_{i,k} + \tau_i \sum_{j=k+1}^{i} h_{j,k} \tag{6.1}$$

Then the bilinear minimax problem becomes SDP again.

$$\hat{\boldsymbol{r}} \triangleq \underset{\boldsymbol{r}\in\mathbb{R}^{N(N+1)/2}}{\arg\min} \; \breve{\mathcal{B}}_{\mathrm{D}}(\boldsymbol{r}, N, L, R), \tag{RD}$$

where

$$\breve{\mathcal{B}}_{\mathrm{D}}(\boldsymbol{r}, N, L, R) \triangleq \min_{\substack{\boldsymbol{\lambda}\in\mathbb{R}^N,\\ \boldsymbol{\tau}\in\mathbb{R}^{N+1},\\ \gamma\in\mathbb{R}}} \left\{ \frac{1}{2}LR^2\gamma \colon \begin{pmatrix} \breve{S}(\boldsymbol{r}, \boldsymbol{\lambda}, \boldsymbol{\tau}) & \frac{1}{2}\boldsymbol{\tau} \\ \frac{1}{2}\boldsymbol{\tau}^\top & \frac{1}{2}\gamma \end{pmatrix} \succeq 0,\, (\boldsymbol{\lambda}, \boldsymbol{\tau}) \in \Lambda \right\},$$

$$\begin{aligned} \breve{S}(\boldsymbol{r}, \boldsymbol{\lambda}, \boldsymbol{\tau}) \triangleq & \frac{1}{2}\sum_{i=1}^{N} \lambda_i (\boldsymbol{u}_{i-1} - \boldsymbol{u}_i)(\boldsymbol{u}_{i-1} - \boldsymbol{u}_i)^\top + \frac{1}{2}\sum_{i=0}^{N} \tau_i \boldsymbol{u}_i \boldsymbol{u}_i^\top \\ & + \frac{1}{2}\sum_{i=1}^{N}\sum_{k=0}^{i-1} r_{i,k}\left(\boldsymbol{u}_i\boldsymbol{u}_k^\top + \boldsymbol{u}_k\boldsymbol{u}_i^\top\right). \end{aligned} \tag{6.2}$$

Analytical OGM (RD')

Utilize the fact that $S_{N,N} = \frac{1}{2}$, one can split the last variable in w and optimize it first, thus reducing the problem to:

$$\hat{\boldsymbol{r}} = \underset{\boldsymbol{r}\in\mathbb{R}^{N(N+1)/2}}{\arg\min}\; \breve{\mathcal{B}}_{\mathrm{D1}}(\boldsymbol{r}, N, L, R), \tag{RD1}$$

where

$$\breve{\mathcal{B}}_{\mathrm{D1}}(\boldsymbol{r}, N, L, R) \triangleq \min_{\substack{\boldsymbol{\lambda}\in\mathbb{R}^N,\\ \boldsymbol{\tau}\in\mathbb{R}^{N+1},\\ \gamma\in\mathbb{R}}} \left\{ \frac{1}{2}LR^2\gamma : \begin{pmatrix} \breve{\boldsymbol{Q}} - 2\breve{\boldsymbol{q}}\breve{\boldsymbol{q}}^\top & \frac{1}{2}(\breve{\boldsymbol{\tau}} - 2\breve{\boldsymbol{q}}\tau_N) \\ \frac{1}{2}(\breve{\boldsymbol{\tau}} - 2\breve{\boldsymbol{q}}\tau_N)^\top & \frac{1}{2}(\gamma - \tau_N^2) \end{pmatrix} \succeq 0,\; (\boldsymbol{\lambda}, \boldsymbol{\tau}) \in \Lambda \right\},$$

$$\breve{\boldsymbol{Q}}(\boldsymbol{r}, \boldsymbol{\lambda}, \boldsymbol{\tau}) = \frac{1}{2}\sum_{i=1}^{N-1}\lambda_i(\breve{\boldsymbol{u}}_{i-1} - \breve{\boldsymbol{u}}_i)(\breve{\boldsymbol{u}}_{i-1} - \breve{\boldsymbol{u}}_i)^\top + \frac{1}{2}\lambda_N\breve{\boldsymbol{u}}_{N-1}\breve{\boldsymbol{u}}_{N-1}^\top$$
$$+ \frac{1}{2}\sum_{i=0}^{N-1}\tau_i\breve{\boldsymbol{u}}_i\breve{\boldsymbol{u}}_i^\top + \frac{1}{2}\sum_{i=1}^{N-1}\sum_{k=0}^{i-1} r_{i,k}\left(\breve{\boldsymbol{u}}_i\breve{\boldsymbol{u}}_k^\top + \breve{\boldsymbol{u}}_k\breve{\boldsymbol{u}}_i^\top\right), \tag{6.7}$$

$$\breve{\boldsymbol{q}}(\boldsymbol{r}, \boldsymbol{\lambda}, \boldsymbol{\tau}) = \frac{1}{2}\sum_{k=0}^{N-2} r_{N,k}\breve{\boldsymbol{u}}_k, + \frac{1}{2}(r_{N,N-1} - \lambda_N)\breve{\boldsymbol{u}}_{N-1} \tag{6.8}$$

Lemma 2 *A feasible point of both (RD) and (RD1) is* $(\hat{\boldsymbol{r}}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\tau}}, \hat{\gamma})$, *where*

$$\hat{r}_{i,k} = \begin{cases} \frac{4\theta_i\theta_k}{\theta_N^2}, & i = 2, \ldots, N-1, \ k = 0, \ldots, i-2, \\ \frac{4\theta_i\theta_{i-1}}{\theta_N^2} + \frac{2\theta_{i-1}^2}{\theta_N^2}, & i = 1, \ldots, N-1, \ k = i-1, \\ \frac{2\theta_k}{\theta_N}, & i = N, \ k = 0, \ldots, i-2, \\ \frac{2\theta_{N-1}}{\theta_N} + \frac{2\theta_{N-1}^2}{\theta_N^2}, & i = N, \ k = i-1, \end{cases} \tag{6.9}$$

$$\hat{\lambda}_i = \frac{2\theta_{i-1}^2}{\theta_N^2}, \quad i = 1, \ldots, N, \tag{6.10}$$

$$\hat{\tau}_i = \begin{cases} \frac{2\theta_i}{\theta_N^2}, & i = 0, \ldots, N-1, \\ 1 - \frac{2\theta_{N-1}^2}{\theta_N^2} = \frac{1}{\theta_N}, & i = N, \end{cases} \tag{6.11}$$

$$\hat{\gamma} = \frac{1}{\theta_N^2}, \tag{6.12}$$

for

$$\theta_i = \begin{cases} 1, & i = 0, \\ \frac{1+\sqrt{1+4\theta_{i-1}^2}}{2}, & i = 1, \ldots, N-1, \\ \frac{1+\sqrt{1+8\theta_{i-1}^2}}{2} & i = N. \end{cases} \tag{6.13}$$

Analytical OGM

Lemma (lemma 3 [Kim and Fessler, 2016])

The choice of $(r, \lambda, \tau, \gamma)$ *given by lemma 2 is optimal solution to both (RD) and (RD1) as KKT conditions hold.*

Lemma (lemma 4 [Kim and Fessler, 2016])

The choice of h *given* θ *in lemma 2, which is*

$$h_{i+1,k} = \begin{cases} \frac{1}{\theta_{i+1}}(2\theta_k - \sum_{j=k+1}^{i} h_{j,k}), & k = 0, \dots, i-1, \\ 1 + \frac{2\theta_i - 1}{\theta_{i+1}}, & k = i, \end{cases}$$

is an optimal solution of (HD).

OGM Bound

With given feasible $\gamma = \frac{1}{\theta_N^2}$, we have a bound on OGM:

Theorem 2 *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and $C_L^{1,1}$ and let $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d$ be generated by Algorithm FO with $\hat{\boldsymbol{h}}$ (6.16) for a given $N \geq 1$. Then*

$$f(\boldsymbol{x}_N) - f(\boldsymbol{x}_*) \leq \frac{L||\boldsymbol{x}_0 - \boldsymbol{x}_*||^2}{2\theta_N^2} \leq \frac{L||\boldsymbol{x}_0 - \boldsymbol{x}_*||^2}{(N+1)(N+1+\sqrt{2})}, \quad \forall \boldsymbol{x}_* \in X_*(f). \tag{6.17}$$

This bound has constant 1. Two times faster than Nesterov's Fast Gradient Method (FGM).

OGM Lower Bound

Similar to GM, OGM is also lower bounded by the case of piecewise affine-quadratic function.

Theorem 3 *For the following convex functions in $C_L^{1,1}(\mathbb{R}^d)$ for all $d \geq 1$:*

$$\phi(\boldsymbol{x}) = \begin{cases} \frac{LR}{\theta_N^2}||\boldsymbol{x}|| - \frac{LR^2}{2\theta_N^4}, & \text{if } ||\boldsymbol{x}|| \geq \frac{R}{\theta_N^2}, \\ \frac{L}{2}||\boldsymbol{x}||^2, & \text{otherwise}, \end{cases} \tag{8.1}$$

both OGM1 and OGM2 exactly achieve the smallest upper bound in (6.17), i.e.,

$$\phi(\boldsymbol{x}_N) - \phi(\boldsymbol{x}_*) = \frac{L||\boldsymbol{x}_0 - \boldsymbol{x}_*||^2}{2\theta_N^2}.$$

This shows that OGM upper bound is tight.

OGM Method

Collect the pieces we get the practical OGM method:

Algorithm OGM1

Input: $f \in C_L^{1,1}(\mathbb{R}^d)$ convex, $\boldsymbol{x}_0 \in \mathbb{R}^d$, $\boldsymbol{y}_0 = \boldsymbol{x}_0$, $\theta_0 = 1$.

For $i = 0, \ldots, N-1$

$$\boldsymbol{y}_{i+1} = \boldsymbol{x}_i - \frac{1}{L} f'(\boldsymbol{x}_i)$$

$$\theta_{i+1} = \begin{cases} \frac{1+\sqrt{1+4\theta_i^2}}{2}, & i \le N-2 \\ \frac{1+\sqrt{1+8\theta_i^2}}{2}, & i = N-1 \end{cases}$$

$$\boldsymbol{x}_{i+1} = \boldsymbol{y}_{i+1} + \frac{\theta_i - 1}{\theta_{i+1}}(\boldsymbol{y}_{i+1} - \boldsymbol{y}_i) + \frac{\theta_i}{\theta_{i+1}}(\boldsymbol{y}_{i+1} - \boldsymbol{x}_i)$$

Optimized Method on Other Metrics

There are some other work that gives optimized method which is optimal not wih $f(x_N) - f_*$, but with $\|x_N - x_*\|^2$ or $\|\nabla f(x_N)\|$.
See [d'Aspremont et al., 2021] Chap 4 for these advanced optimized methods.

Bibliography I

Barr'e, M., Taylor, A. B., and d'Aspremont, A. (2020).
Complexity guarantees for polyak steps with momentum.
In *COLT*.

Colla, S. and Hendrickx, J. M. (2021).
Automated worst-case performance analysis of decentralized gradient descent.
ArXiv, abs/2103.14396.

d'Aspremont, A., Scieur, D., and Taylor, A. B. (2021).
Acceleration methods.
ArXiv, abs/2101.09545.

Dragomir, R., Taylor, A. B., d'Aspremont, A., and Bolte, J. (2019).
Optimal complexity and certification of bregman first-order methods.
ArXiv, abs/1911.08510.

Bibliography II

Drori, Y. and Teboulle, M. (2014).
Performance of first-order methods for smooth convex minimization: a novel approach.
Mathematical Programming, 145:451–482.

Kim, D. and Fessler, J. (2016).
Optimized first-order methods for smooth convex minimization.
Mathematical Programming, 159:81–107.

Kim, D. and Fessler, J. (2017).
On the convergence analysis of the optimized gradient method.
Journal of Optimization Theory and Applications, 172:187–205.

Bibliography III

Kim, D. and Fessler, J. (2018).
Another look at the fast iterative shrinkage/thresholding algorithm (fista).
SIAM journal on optimization : a publication of the Society for Industrial and Applied Mathematics, 28 1:223–250.

Lieder, F. (2021).
On the convergence rate of the halpern-iteration.
Optimization Letters, pages 1–14.

Ryu, E. K., Taylor, A. B., Bergeling, C., and Giselsson, P. (2020).
Operator splitting performance estimation: Tight contraction factors and optimal parameter selection.
SIAM J. Optim., 30:2251–2271.

Bibliography IV

Taylor, A. B. and Bach, F. R. (2019).
Stochastic first-order methods: non-asymptotic and computer-aided analyses via potential functions.
In *COLT*.

Taylor, A. B., Hendrickx, J., and Glineur, F. (2017a).
Smooth strongly convex interpolation and exact worst-case performance of first-order methods.
Mathematical Programming, 161:307–345.

Taylor, A. B., Hendrickx, J. M., and Glineur, F. (2017b).
Performance estimation toolbox (pesto): Automated worst-case analysis of first-order optimization methods.
2017 IEEE 56th Annual Conference on Decision and Control (CDC), pages 1278–1283.

Bibliography V

Taylor, A. B., Hendrickx, J. M., and Glineur, F. (2018a).
Exact worst-case convergence rates of the proximal gradient method for composite convex minimization.
Journal of Optimization Theory and Applications, 178:455–476.

Taylor, A. B., Scoy, B. V., and Lessard, L. (2018b).
Lyapunov functions for first-order methods: Tight automated convergence guarantees.
In *ICML*.