Computer-assisted analyses and design of optimization methods: personal summary and perspectives

Adrien Taylor

PEP-talks — 2023

Thanks to the organizers!

François Glineur

Julien Hendrickx

Etienne de Klerk

Ernest Ryu

Carolina Bergeling

Pontus Giselsson

Francis Bach

Jérôme Bolte

Yoel Drori

Alexandre d'Aspremont

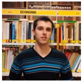
Mathieu Barré

Radu Dragomir

Bryan Van Scoy

Laurent Lessard

Céline Moucer



Baptiste Goujaud

Aymeric Dieuleveut

Shuvomoy Das Gupta

Robert Freund

Andy X. Sun

Eduard Gorbunov

Samuel Horvath

Gauthier Gidel

Manu Upadhyaya

Sebastian Banert

Overview of this talk

- ◇ PEPs: quick recap, problem formulation, notations,

Overview of this talk

- ⋄ PEPs: quick recap, problem formulation, notations,
- ⋄ PEPs: learning outcomes,

Overview of this talk

- ⋄ PEPs: quick recap, problem formulation, notations,
- ⋄ PEPs: learning outcomes,
- ⋄ notions of simplicity (for proofs and worst-case examples),

Overview of this talk

- ◇ PEPs: quick recap, problem formulation, notations,
- ◇ PEPs: learning outcomes,
- ◇ notions of simplicity (for proofs and worst-case examples),
- ◇ creating new methods.

Please contribute!

- ◇ Put your examples/contributions in one of the packages!
  - in Matlab: PESTO,
  - in Python: PEPit.
- ◇ Don't hesitate to use/contribute to "learning PEPs":
  - Learning-Performance-Estimation.
- ◇ We are happy to treat your pull requests!

Genealogy (“my humble, biased, view on...”)

Base methodological developments:

Genealogy ("my humble, biased, view on...")

Base methodological developments:

'14 Drori and Teboulle: upper bounds on worst-case behaviors of FO methods via SDP. Problems scale with number of iterations ($N{\times}N$ SDP matrices).

Genealogy ("my humble, biased, view on...")

Base methodological developments:

'14 Drori and Teboulle: upper bounds on worst-case behaviors of FO methods via SDP. Problems scale with number of iterations ($N \times N$ SDP matrices).

'16 Kim and Fessler: design of an optimized method for smooth convex minimization, using SDPs.

Genealogy ("my humble, biased, view on...")

Base methodological developments:

'14 Drori and Teboulle: upper bounds on worst-case behaviors of FO methods via SDP. Problems scale with number of iterations ($N \times N$ SDP matrices).

'16 Kim and Fessler: design of an optimized method for smooth convex minimization, using SDPs.

'16 Lessard, Recht, Packard: smaller SDPs for linear convergence, via integral quadratic constraints ("IQCs"). Essentially Lyapunov functions.

Genealogy ("my humble, biased, view on...")

Base methodological developments:

'14 Drori and Teboulle: upper bounds on worst-case behaviors of FO methods via SDP. Problems scale with number of iterations ($N \times N$ SDP matrices).

'16 Kim and Fessler: design of an optimized method for smooth convex minimization, using SDPs.

'16 Lessard, Recht, Packard: smaller SDPs for linear convergence, via integral quadratic constraints ("IQCs"). Essentially Lyapunov functions.

This presentation: mainly points of view from

Genealogy ("my humble, biased, view on...")

Base methodological developments:

'14 Drori and Teboulle: upper bounds on worst-case behaviors of FO methods via SDP. Problems scale with number of iterations ($N \times N$ SDP matrices).

'16 Kim and Fessler: design of an optimized method for smooth convex minimization, using SDPs.

'16 Lessard, Recht, Packard: smaller SDPs for linear convergence, via integral quadratic constraints ("IQCs"). Essentially Lyapunov functions.

This presentation: mainly points of view from

'17 T, Hendrickx and Glineur: "principled formulations" + tightness (via interpolation/extensions).

Genealogy ("my humble, biased, view on...")

Base methodological developments:

'14 Drori and Teboulle: upper bounds on worst-case behaviors of FO methods via SDP. Problems scale with number of iterations ($N \times N$ SDP matrices).

'16 Kim and Fessler: design of an optimized method for smooth convex minimization, using SDPs.

'16 Lessard, Recht, Packard: smaller SDPs for linear convergence, via integral quadratic constraints ("IQCs"). Essentially Lyapunov functions.

This presentation: mainly points of view from

'17 T, Hendrickx and Glineur: "principled formulations" + tightness (via interpolation/extensions).

'19 T, Bach: potential functions. Essentially: try to "force" simpler proofs.

Genealogy (“my humble, biased, view on...”)

Base methodological developments:

- ’14 Drori and Teboulle: upper bounds on worst-case behaviors of FO methods via SDP. Problems scale with number of iterations ($N{\times}N$ SDP matrices).
- ’16 Kim and Fessler: design of an optimized method for smooth convex minimization, using SDPs.
- ’16 Lessard, Recht, Packard: smaller SDPs for linear convergence, via integral quadratic constraints (“IQCs”). Essentially Lyapunov functions.

This presentation: mainly points of view from

- ’17 T, Hendrickx and Glineur: “principled formulations” + tightness (via interpolation/extensions).
- ’19 T, Bach: potential functions. Essentially: try to “force” simpler proofs.
- ’20, ’22 Drori, T: Constructive approaches to optimal first-order methods.

Example: analysis of a gradient method

Find $x_\star \in \mathbb{R}^d$ such that

$$f(x_\star) = \min_{x \in \mathbb{R}^d} f(x),$$

with $f \in \mathcal{F}_{\mu,L}$ (L-smooth μ-strongly convex).

Example: analysis of a gradient method

Find $x_\star \in \mathbb{R}^d$ such that

$$f(x_\star) = \min_{x \in \mathbb{R}^d} f(x),$$

with $f \in \mathcal{F}_{\mu,L}$ (L-smooth μ-strongly convex).

(Gradient method) We decide to use: $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$

Example: analysis of a gradient method

Find $x_\star \in \mathbb{R}^d$ such that

$$f(x_\star) = \min_{x \in \mathbb{R}^d} f(x),$$

with $f \in \mathcal{F}_{\mu,L}$ (L-smooth μ-strongly convex).

(Gradient method) We decide to use: $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$

Question: what *a priori* guarantees after N iterations?

Example: analysis of a gradient method

Find $x_\star \in \mathbb{R}^d$ such that

$$f(x_\star) = \min_{x \in \mathbb{R}^d} f(x),$$

with $f \in \mathcal{F}_{\mu,L}$ (L-smooth μ-strongly convex).

(Gradient method) We decide to use: $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$

Question: what *a priori* guarantees after N iterations?

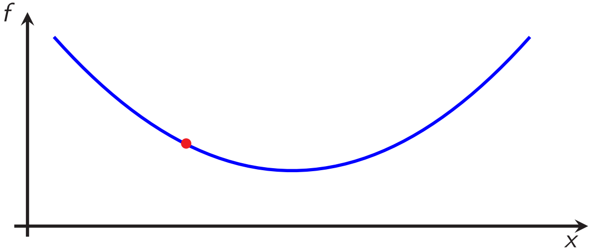
Examples: what about $f(x_N) - f(x_\star)$, $\|\nabla f(x_N)\|$, $\|x_N - x_\star\|$?

About the assumptions

Consider a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, f is (μ-strongly) convex and L-smooth iff $\forall x, y \in \mathbb{R}^d$ we have:

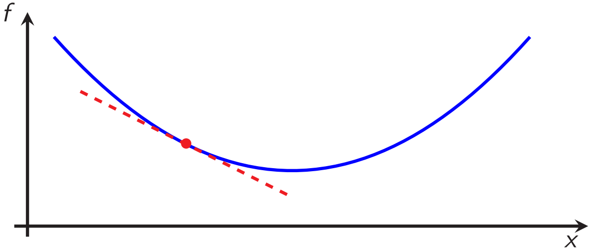
About the assumptions

Consider a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, f is (μ-strongly) convex and L-smooth iff $\forall x, y \in \mathbb{R}^d$ we have:



About the assumptions

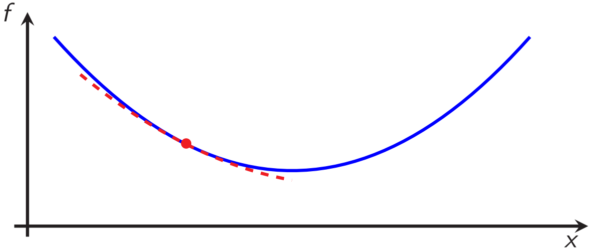
Consider a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, f is (μ-strongly) convex and L-smooth iff $\forall x, y \in \mathbb{R}^d$ we have:



(1) (Convexity) $f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle$,

About the assumptions

Consider a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, f is (μ-strongly) convex and L-smooth iff $\forall x, y \in \mathbb{R}^d$ we have:

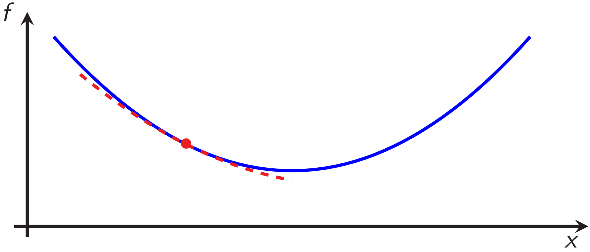


(1) (Convexity) $f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle$,

(1b) (μ-strong convexity) $f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle + \frac{\mu}{2} \|x - y\|^2$,

About the assumptions

Consider a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, f is (μ-strongly) convex and L-smooth iff $\forall x, y \in \mathbb{R}^d$ we have:



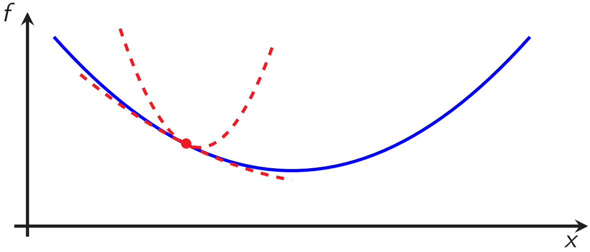
$$(1) \quad \text{(Convexity)} \; f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle,$$

$$(1b) \quad (\mu\text{-strong convexity}) \; f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle + \frac{\mu}{2} \|x - y\|^2,$$

$$(2) \quad \text{(L-smoothness)} \; \|\nabla f(x) - \nabla f(y)\| \leqslant L \|x - y\|,$$

About the assumptions

Consider a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, f is (μ-strongly) convex and L-smooth iff $\forall x, y \in \mathbb{R}^d$ we have:



(1) (Convexity) $f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle$,

(1b) (μ-strong convexity) $f(x) \geqslant f(y) + \langle \nabla f(y), x - y \rangle + \frac{\mu}{2} \|x - y\|^2$,

(2) (L-smoothness) $\|\nabla f(x) - \nabla f(y)\| \leqslant L \|x - y\|$,

(2b) (L-smoothness) $f(x) \leqslant f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2} \|x - y\|^2$.

Convergence rate of a gradient step

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,\mathrm{L}}$),
- x_0, and x_1 generated by gradient step $x_1 = x_0 - \gamma_0 \nabla f(x_0)$,
- $x_\star = \underset{x}{\operatorname{argmin}}\ f(x)$?

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- ◇ L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,\mathrm{L}}$),
- ◇ x_0, and x_1 generated by gradient step $x_1 = x_0 - \gamma_0 \nabla f(x_0)$,
- ◇ $x_\star = \underset{x}{\operatorname{argmin}}\ f(x)$?

First: let's compute τ!

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,\mathrm{L}}$),
- x_0, and x_1 generated by gradient step $x_1 = x_0 - \gamma_0 \nabla f(x_0)$,
- $x_\star = \underset{x}{\operatorname{argmin}}\ f(x)$?

First: let's compute τ!

$$\tau = \max_{f, x_0, x_1, x_\star} \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

s.t.	$f \in \mathcal{F}_{\mu,L}$	Functional class
	$x_1 = x_0 - \gamma_0 \nabla f(x_0)$	Algorithm
	$\nabla f(x_\star) = 0$	Optimality of $x_\star$

Convergence rate of a gradient step

Toy example: What is the smallest τ such that:

$$\|x_1 - x_\star\|^2 \leqslant \tau \|x_0 - x_\star\|^2,$$

for all

- L-smooth and μ-strongly convex function f (notation $f \in \mathcal{F}_{\mu,\mathrm{L}}$),
- x_0, and x_1 generated by gradient step $x_1 = x_0 - \gamma_0 \nabla f(x_0)$,
- $x_\star = \underset{x}{\operatorname{argmin}}\ f(x)$?

First: let's compute τ!

$$\tau = \max_{f, x_0, x_1, x_\star} \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

s.t.	$f \in \mathcal{F}_{\mu,L}$	Functional class
	$x_1 = x_0 - \gamma_0 \nabla f(x_0)$	Algorithm
	$\nabla f(x_\star) = 0$	Optimality of $x_\star$

Variables: f, x_0, x_1, $x_\star$; parameters: μ, L, γ_0.

Sampled version

Sampled version

- Performance estimation problem:

$$\begin{aligned} \max_{f, x_0, x_1, x_\star} \quad & \frac{\|x_1 - x_0\|^2}{\|x_0 - x_\star\|^2} \\ \text{subject to} \quad & f \text{ is } L\text{-smooth and } \mu\text{-strongly convex,} \\ & x_1 = x_0 - \gamma_0 \nabla f(x_0) \\ & \nabla f(x_\star) = 0. \end{aligned}$$

Sampled version

- Performance estimation problem:

$$\begin{aligned} \max_{f, x_0, x_1, x_\star} \quad & \frac{\|x_1 - x_0\|^2}{\|x_0 - x_\star\|^2} \\ \text{subject to} \quad & f \text{ is } L\text{-smooth and } \mu\text{-strongly convex}, \\ & x_1 = x_0 - \gamma_0 \nabla f(x_0) \\ & \nabla f(x_\star) = 0. \end{aligned}$$

- Variables: f, x_0, x_1, $x_\star$.

Sampled version

- Performance estimation problem:

$$\begin{aligned} \max_{f, x_0, x_1, x_\star} \quad & \frac{\|x_1 - x_0\|^2}{\|x_0 - x_\star\|^2} \\ \text{subject to} \quad & f \text{ is } L\text{-smooth and } \mu\text{-strongly convex,} \\ & x_1 = x_0 - \gamma_0 \nabla f(x_0) \\ & \nabla f(x_\star) = 0. \end{aligned}$$

- Variables: f, x_0, x_1, $x_\star$.
- Sampled version: f is only used at x_0 and $x_\star$ (no need to sample other points)

Sampled version

- Performance estimation problem:

$$\begin{aligned} \max_{f, x_0, x_1, x_\star} \quad & \frac{\|x_1 - x_0\|^2}{\|x_0 - x_\star\|^2} \\ \text{subject to} \quad & f \text{ is } L\text{-smooth and } \mu\text{-strongly convex,} \\ & x_1 = x_0 - \gamma_0 \nabla f(x_0) \\ & \nabla f(x_\star) = 0. \end{aligned}$$

- Variables: f, x_0, x_1, $x_\star$.
- Sampled version: f is only used at x_0 and $x_\star$ (no need to sample other points)

$$\begin{aligned} \max_{\substack{x_0, x_1, x_\star \\ g_0, g_\star \\ f_0, f_\star}} \quad & \frac{\|x_1 - x_0\|^2}{\|x_0 - x_\star\|^2} \\ \text{subject to} \quad & \exists f \in \mathcal{F}_{\mu,L} \text{ such that } \begin{cases} f_i = f(x_i) & i = 0, \star \\ g_i = \nabla f(x_i) & i = 0, \star \end{cases} \\ & x_1 = x_0 - \gamma_0 g_0 \\ & g_\star = 0. \end{aligned}$$

Sampled version

- Performance estimation problem:

$$\begin{aligned} \max_{f, x_0, x_1, x_\star} \quad & \frac{\|x_1 - x_0\|^2}{\|x_0 - x_\star\|^2} \\ \text{subject to} \quad & f \text{ is } L\text{-smooth and } \mu\text{-strongly convex,} \\ & x_1 = x_0 - \gamma_0 \nabla f(x_0) \\ & \nabla f(x_\star) = 0. \end{aligned}$$

- Variables: f, x_0, x_1, $x_\star$.
- Sampled version: f is only used at x_0 and $x_\star$ (no need to sample other points)

$$\begin{aligned} \max_{\substack{x_0, x_1, x_\star \\ g_0, g_\star \\ f_0, f_\star}} \quad & \frac{\|x_1 - x_0\|^2}{\|x_0 - x_\star\|^2} \\ \text{subject to} \quad & \exists f \in \mathcal{F}_{\mu,L} \text{ such that } \begin{cases} f_i = f(x_i) & i = 0, \star \\ g_i = \nabla f(x_i) & i = 0, \star \end{cases} \\ & x_1 = x_0 - \gamma_0 g_0 \\ & g_\star = 0. \end{aligned}$$

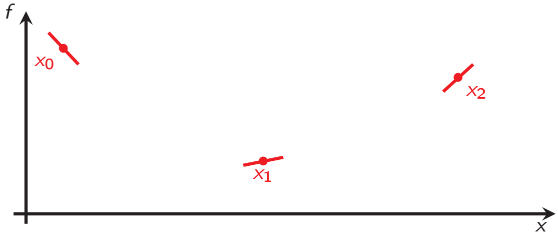
- Variables: x_0, x_1, $x_\star$, g_0, $g_\star$, f_0, $f_\star$.

Smooth strongly convex interpolation (or extension)

Consider an index set S, and its associated values $\{(x_i, g_i, f_i)\}_{i\in S}$ with coordinates x_i, (sub)gradients g_i and function values f_i.

Smooth strongly convex interpolation (or extension)

Consider an index set S, and its associated values $\{(x_i, g_i, f_i)\}_{i\in S}$ with coordinates x_i, (sub)gradients g_i and function values f_i.

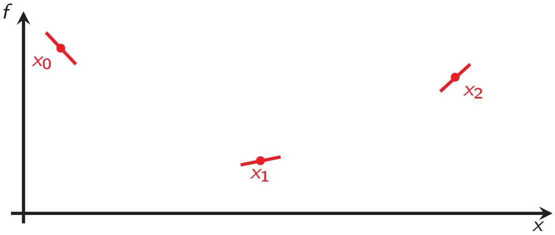


? Possible to find $f \in \mathcal{F}_{\mu,L}$ such that

$$f(x_i) = f_i, \quad \text{and} \quad g_i \in \partial f(x_i), \qquad \forall i \in S.$$

Smooth strongly convex interpolation (or extension)

Consider an index set S, and its associated values $\{(x_i, g_i, f_i)\}_{i\in S}$ with coordinates x_i, (sub)gradients g_i and function values f_i.



? Possible to find $f \in \mathcal{F}_{\mu,L}$ such that

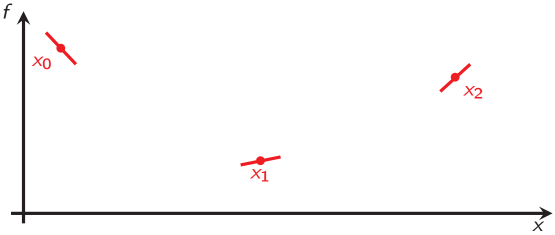
$$f(x_i) = f_i, \quad \text{and} \quad g_i \in \partial f(x_i), \qquad \forall i \in S.$$

- Necessary and sufficient condition: $\forall i, j \in S$

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

Smooth strongly convex interpolation (or extension)

Consider an index set S, and its associated values $\{(x_i, g_i, f_i)\}_{i\in S}$ with coordinates x_i, (sub)gradients g_i and function values f_i.



? Possible to find $f \in \mathcal{F}_{\mu,L}$ such that

$$f(x_i) = f_i, \quad \text{and} \quad g_i \in \partial f(x_i), \qquad \forall i \in S.$$

- Necessary and sufficient condition: $\forall i, j \in S$

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

- Simpler example: pick $\mu = 0$ and $L = \infty$ (just convexity):

$$f_i \geqslant f_j + \langle g_j, x_i - x_j\rangle.$$

Replace constraints

Replace constraints

- Interpolation conditions allow removing red constraints

$$\max_{\substack{x_0, x_1, x_\star \\ g_0, g_\star \\ f_0, f_\star}} \quad \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

$$\text{subject to} \quad \exists f \in \mathcal{F}_{\mu,L} \text{ such that } \begin{cases} f_i = f(x_i) & i = 0, \star \\ g_i = \nabla f(x_i) & i = 0, \star \end{cases}$$

$$x_1 = x_0 - \gamma_0 g_0$$

$$g_\star = 0,$$

Replace constraints

- ◇ Interpolation conditions allow removing red constraints

$$\max_{\substack{x_0, x_1, x_\star \\ g_0, g_\star \\ f_0, f_\star}} \quad \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

$$\text{subject to} \quad \exists f \in \mathcal{F}_{\mu,L} \text{ such that } \begin{cases} f_i = f(x_i) & i = 0, \star \\ g_i = \nabla f(x_i) & i = 0, \star \end{cases}$$

$$x_1 = x_0 - \gamma_0 g_0$$

$$g_\star = 0,$$

- ◇ replacing them by

$$f_\star \geqslant f_0 + \langle g_0, x_\star - x_0 \rangle + \tfrac{1}{2L} \|g_\star - g_0\|^2 + \tfrac{\mu}{2(1-\mu/L)} \left\|x_\star - x_0 - \tfrac{1}{L}(g_\star - g_0)\right\|^2$$

$$f_0 \geqslant f_\star + \langle g_\star, x_0 - x_\star \rangle + \tfrac{1}{2L} \|g_0 - g_\star\|^2 + \tfrac{\mu}{2(1-\mu/L)} \left\|x_0 - x_\star - \tfrac{1}{L}(g_0 - g_\star)\right\|^2.$$

Replace constraints

◇ Interpolation conditions allow removing red constraints

$$\max_{\substack{x_0, x_1, x_\star \\ g_0, g_\star \\ f_0, f_\star}} \quad \frac{\|x_1 - x_\star\|^2}{\|x_0 - x_\star\|^2}$$

$$\text{subject to} \quad \exists f \in \mathcal{F}_{\mu,L} \text{ such that } \begin{cases} f_i = f(x_i) & i = 0, \star \\ g_i = \nabla f(x_i) & i = 0, \star \end{cases}$$

$$x_1 = x_0 - \gamma_0 g_0$$

$$g_\star = 0,$$

◇ replacing them by

$$f_\star \geqslant f_0 + \langle g_0, x_\star - x_0 \rangle + \tfrac{1}{2L}\|g_\star - g_0\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_\star - x_0 - \tfrac{1}{L}(g_\star - g_0)\right\|^2$$

$$f_0 \geqslant f_\star + \langle g_\star, x_0 - x_\star \rangle + \tfrac{1}{2L}\|g_0 - g_\star\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_\star - \tfrac{1}{L}(g_0 - g_\star)\right\|^2.$$

◇ Same optimal value (no relaxation); but still non-convex quadratic problem.

Semidefinite lifting

Semidefinite lifting

- ◇ Using the new variables $G \succcurlyeq 0$ and F

$$G = \begin{bmatrix} \|x_0 - x_\star\|^2 & \langle g_0, x_0 - x_\star\rangle \\ \langle g_0, x_0 - x_\star\rangle & \|g_0\|^2 \end{bmatrix}, \quad F = f_0 - f_\star,$$

Semidefinite lifting

- Using the new variables $G \succcurlyeq 0$ and F

$$G = \begin{bmatrix} \|x_0 - x_\star\|^2 & \langle g_0, x_0 - x_\star\rangle \\ \langle g_0, x_0 - x_\star\rangle & \|g_0\|^2 \end{bmatrix}, \quad F = f_0 - f_\star,$$

- previous problem can be reformulated as a 2×2 SDP

$$\begin{aligned} \max_{G,\,F} \quad & \frac{G_{1,1} + \gamma_0^2 G_{2,2} - 2\gamma_0 G_{1,2}}{G_{1,1}} \\ \text{subject to} \quad & F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{L}{L-\mu} G_{1,2} \leqslant 0 \\ & -F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{\mu}{L-\mu} G_{1,2} \leqslant 0 \\ & G \succcurlyeq 0, \end{aligned}$$

Semidefinite lifting

- Using the new variables $G \succcurlyeq 0$ and F

$$G = \begin{bmatrix} \|x_0 - x_\star\|^2 & \langle g_0, x_0 - x_\star \rangle \\ \langle g_0, x_0 - x_\star \rangle & \|g_0\|^2 \end{bmatrix}, \quad F = f_0 - f_\star,$$

- previous problem can be reformulated as a 2×2 SDP

$$\begin{aligned} \max_{G,\,F} \quad & G_{1,1} + \gamma_0^2 G_{2,2} - 2\gamma_0 G_{1,2} \\ \text{subject to} \quad & F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{L}{L-\mu} G_{1,2} \leqslant 0 \\ & -F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{\mu}{L-\mu} G_{1,2} \leqslant 0 \\ & G_{1,1} = 1 \\ & G \succcurlyeq 0, \end{aligned}$$

(using an an homogeneity argument and substituting x_1 and $g_\star$).

Semidefinite lifting

- Using the new variables $G \succcurlyeq 0$ and F

$$G = \begin{bmatrix} \|x_0 - x_\star\|^2 & \langle g_0, x_0 - x_\star\rangle \\ \langle g_0, x_0 - x_\star\rangle & \|g_0\|^2 \end{bmatrix}, \quad F = f_0 - f_\star,$$

- previous problem can be reformulated as a 2×2 SDP

$$\begin{aligned} \max_{G,\,F} \quad & G_{1,1} + \gamma_0^2 G_{2,2} - 2\gamma_0 G_{1,2} \\ \text{subject to} \quad & F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{L}{L-\mu} G_{1,2} \leqslant 0 \\ & -F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{\mu}{L-\mu} G_{1,2} \leqslant 0 \\ & G_{1,1} = 1 \\ & G \succcurlyeq 0, \end{aligned}$$

(using an an homogeneity argument and substituting x_1 and $g_\star$).

- Assuming $x_0, x_\star, g_0 \in \mathbb{R}^d$ with $d \geqslant 2$, same optimal value as original problem!

Semidefinite lifting

- Using the new variables $G \succcurlyeq 0$ and F

$$G = \begin{bmatrix} \|x_0 - x_\star\|^2 & \langle g_0, x_0 - x_\star \rangle \\ \langle g_0, x_0 - x_\star \rangle & \|g_0\|^2 \end{bmatrix}, \quad F = f_0 - f_\star,$$

- previous problem can be reformulated as a 2×2 SDP

$$\begin{aligned} \max_{G,\,F} \quad & G_{1,1} + \gamma_0^2 G_{2,2} - 2\gamma_0 G_{1,2} \\ \text{subject to} \quad & F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{L}{L-\mu} G_{1,2} \leqslant 0 \\ & -F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{\mu}{L-\mu} G_{1,2} \leqslant 0 \\ & G_{1,1} = 1 \\ & G \succcurlyeq 0, \end{aligned}$$

(using an an homogeneity argument and substituting x_1 and $g_\star$).

- Assuming $x_0, x_\star, g_0 \in \mathbb{R}^d$ with $d \geqslant 2$, same optimal value as original problem!
- For $d = 1$ same as original problem by adding $\text{rank}(G) \leqslant 1$.

Dual problem

◇ Dual problem is

$$
\begin{aligned}
\min_{\tau,\lambda_1,\lambda_2\geqslant 0}\ & \tau \\
\text{subject to } S = & \begin{bmatrix} \tau - 1 + \frac{\lambda_1 L\mu}{L-\mu} & \gamma_0 - \frac{\lambda_1(\mu+L)}{2(L-\mu)} \\ \gamma_0 - \frac{\lambda_1(\mu+L)}{2(L-\mu)} & \frac{\lambda_1}{L-\mu} - \gamma_0^2 \end{bmatrix} \succcurlyeq 0 \\
0 = & \lambda_1 - \lambda_2.
\end{aligned}
$$

Dual problem

◇ Dual problem is

$$\min_{\tau,\lambda_1,\lambda_2 \geqslant 0} \tau$$

$$\text{subject to } S = \begin{bmatrix} \tau - 1 + \frac{\lambda_1 L\mu}{L-\mu} & \gamma_0 - \frac{\lambda_1(\mu+L)}{2(L-\mu)} \\ \gamma_0 - \frac{\lambda_1(\mu+L)}{2(L-\mu)} & \frac{\lambda_1}{L-\mu} - \gamma_0^2 \end{bmatrix} \succcurlyeq 0$$

$$0 = \lambda_1 - \lambda_2.$$

◇ Weak duality: any dual feasible point $\equiv$ valid worst-case convergence rate

Dual problem

- Dual problem is

$$\min_{\tau,\lambda_1,\lambda_2\geqslant 0} \tau$$
$$\text{subject to } S = \begin{bmatrix} \tau - 1 + \frac{\lambda_1 L\mu}{L-\mu} & \gamma_0 - \frac{\lambda_1(\mu+L)}{2(L-\mu)} \\ \gamma_0 - \frac{\lambda_1(\mu+L)}{2(L-\mu)} & \frac{\lambda_1}{L-\mu} - \gamma_0^2 \end{bmatrix} \succcurlyeq 0$$
$$0 = \lambda_1 - \lambda_2.$$

- Weak duality: any dual feasible point $\equiv$ valid worst-case convergence rate
- Direct consequence: for any $\tau \geqslant 0$ we have

$$\|x_1 - x_\star\|^2 \leqslant \tau\|x_0 - x_\star\|^2 \text{ for all } f \in \mathcal{F}_{\mu,L}, \text{ all } x_0 \in \mathbb{R}^d, \text{ all } d \in \mathbb{N},$$
$$\text{with } x_1 = x_0 - \gamma_0 \nabla f(x_0).$$
$$\Uparrow$$
$$\exists\lambda \geqslant 0 : \begin{bmatrix} \tau - 1 + \frac{\lambda L\mu}{L-\mu} & \gamma_0 - \frac{\lambda(\mu+L)}{2(L-\mu)} \\ \gamma_0 - \frac{\lambda(\mu+L)}{2(L-\mu)} & \frac{\lambda}{L-\mu} - \gamma_0^2 \end{bmatrix} \succcurlyeq 0$$

Dual problem

- Dual problem is

$$\min_{\tau,\lambda_1,\lambda_2\geqslant 0} \tau$$
$$\text{subject to } S = \begin{bmatrix} \tau - 1 + \frac{\lambda_1 L\mu}{L-\mu} & \gamma_0 - \frac{\lambda_1(\mu+L)}{2(L-\mu)} \\ \gamma_0 - \frac{\lambda_1(\mu+L)}{2(L-\mu)} & \frac{\lambda_1}{L-\mu} - \gamma_0^2 \end{bmatrix} \succcurlyeq 0$$
$$0 = \lambda_1 - \lambda_2.$$

- Weak duality: any dual feasible point $\equiv$ valid worst-case convergence rate ($\Uparrow$).
- Direct consequence: for any $\tau \geqslant 0$ we have

$$\|x_1 - x_\star\|^2 \leqslant \tau\|x_0 - x_\star\|^2 \text{ for all } f \in \mathcal{F}_{\mu,L}, \text{ all } x_0 \in \mathbb{R}^d, \text{ all } d \in \mathbb{N},$$
$$\text{with } x_1 = x_0 - \gamma_0 \nabla f(x_0).$$
$$\Uparrow$$
$$\exists \lambda \geqslant 0 : \begin{bmatrix} \tau - 1 + \frac{\lambda L\mu}{L-\mu} & \gamma_0 - \frac{\lambda(\mu+L)}{2(L-\mu)} \\ \gamma_0 - \frac{\lambda(\mu+L)}{2(L-\mu)} & \frac{\lambda}{L-\mu} - \gamma_0^2 \end{bmatrix} \succcurlyeq 0$$

Dual problem

- Dual problem is

$$\min_{\tau,\lambda_1,\lambda_2\geqslant 0} \tau$$
$$\text{subject to } S = \begin{bmatrix} \tau - 1 + \frac{\lambda_1 L\mu}{L-\mu} & \gamma_0 - \frac{\lambda_1(\mu+L)}{2(L-\mu)} \\ \gamma_0 - \frac{\lambda_1(\mu+L)}{2(L-\mu)} & \frac{\lambda_1}{L-\mu} - \gamma_0^2 \end{bmatrix} \succcurlyeq 0$$
$$0 = \lambda_1 - \lambda_2.$$

- Weak duality: any dual feasible point $\equiv$ valid worst-case convergence rate ($\Uparrow$).
- Direct consequence: for any $\tau \geqslant 0$ we have

$$\|x_1 - x_\star\|^2 \leqslant \tau\|x_0 - x_\star\|^2 \text{ for all } f \in \mathcal{F}_{\mu,L}, \text{ all } x_0 \in \mathbb{R}^d, \text{ all } d \in \mathbb{N},$$
$$\text{with } x_1 = x_0 - \gamma_0 \nabla f(x_0).$$
$$\Uparrow$$
$$\exists\lambda \geqslant 0 : \begin{bmatrix} \tau - 1 + \frac{\lambda L\mu}{L-\mu} & \gamma_0 - \frac{\lambda(\mu+L)}{2(L-\mu)} \\ \gamma_0 - \frac{\lambda(\mu+L)}{2(L-\mu)} & \frac{\lambda}{L-\mu} - \gamma_0^2 \end{bmatrix} \succcurlyeq 0$$

- Strong duality holds (existence of a Slater point): any valid worst-case convergence rate $\equiv$ valid dual feasible point ($\Downarrow$) .

Dual problem

- Dual problem is

$$\min_{\tau,\lambda_1,\lambda_2\geqslant 0} \tau$$

$$\text{subject to } S = \begin{bmatrix} \tau - 1 + \frac{\lambda_1 L\mu}{L-\mu} & \gamma_0 - \frac{\lambda_1(\mu+L)}{2(L-\mu)} \\ \gamma_0 - \frac{\lambda_1(\mu+L)}{2(L-\mu)} & \frac{\lambda_1}{L-\mu} - \gamma_0^2 \end{bmatrix} \succcurlyeq 0$$

$$0 = \lambda_1 - \lambda_2.$$

- Weak duality: any dual feasible point $\equiv$ valid worst-case convergence rate ($\Uparrow$).
- Direct consequence: for any $\tau \geqslant 0$ we have

$$\|x_1 - x_\star\|^2 \leqslant \tau\|x_0 - x_\star\|^2 \text{ for all } f \in \mathcal{F}_{\mu,L}, \text{ all } x_0 \in \mathbb{R}^d, \text{ all } d \in \mathbb{N},$$
$$\text{with } x_1 = x_0 - \gamma_0 \nabla f(x_0).$$

$$\Updownarrow$$

$$\exists\lambda \geqslant 0 : \begin{bmatrix} \tau - 1 + \frac{\lambda L\mu}{L-\mu} & \gamma_0 - \frac{\lambda(\mu+L)}{2(L-\mu)} \\ \gamma_0 - \frac{\lambda(\mu+L)}{2(L-\mu)} & \frac{\lambda}{L-\mu} - \gamma_0^2 \end{bmatrix} \succcurlyeq 0$$

- Strong duality holds (existence of a Slater point): any valid worst-case convergence rate $\equiv$ valid dual feasible point ($\Downarrow$) : hence "$\Updownarrow$".

Translation to worst-case guarantees

◇ Summary: we can compute for the smallest $\tau(\gamma_0)$ such that

$$\|x_1 - x_\star\|^2 \leqslant \tau(\gamma_0)\|x_0 - x_\star\|^2$$

is satisfied for all $x_0 \in \mathbb{R}^d$, $d \in \mathbb{N}$, $f \in \mathcal{F}_{\mu,L}$, and $x_1 = x_0 - \gamma_0 \nabla f(x_0)$.

Translation to worst-case guarantees

- Summary: we can compute for the smallest $\tau(\gamma_0)$ such that

$$\|x_1 - x_\star\|^2 \leqslant \tau(\gamma_0)\|x_0 - x_\star\|^2$$

  is satisfied for all $x_0 \in \mathbb{R}^d$, $d \in \mathbb{N}$, $f \in \mathcal{F}_{\mu,L}$, and $x_1 = x_0 - \gamma_0 \nabla f(x_0)$.
- Feasible points to SDP correspond to lower bounds on $\tau(\gamma_0)$.
- Feasible points to dual SDP correspond to upper bounds on $\tau(\gamma_0)$.

Translation to worst-case guarantees

- Summary: we can compute for the smallest $\tau(\gamma_0)$ such that

$$\|x_1 - x_\star\|^2 \leqslant \tau(\gamma_0)\|x_0 - x_\star\|^2$$

  is satisfied for all $x_0 \in \mathbb{R}^d$, $d \in \mathbb{N}$, $f \in \mathcal{F}_{\mu,L}$, and $x_1 = x_0 - \gamma_0 \nabla f(x_0)$.
- Feasible points to SDP correspond to lower bounds on $\tau(\gamma_0)$.
- Feasible points to dual SDP correspond to upper bounds on $\tau(\gamma_0)$.
- Therefore:

Translation to worst-case guarantees

- Summary: we can compute for the smallest $\tau(\gamma_0)$ such that

$$\|x_1 - x_\star\|^2 \leqslant \tau(\gamma_0)\|x_0 - x_\star\|^2$$

  is satisfied for all $x_0 \in \mathbb{R}^d$, $d \in \mathbb{N}$, $f \in \mathcal{F}_{\mu,\mathsf{L}}$, and $x_1 = x_0 - \gamma_0 \nabla f(x_0)$.
- Feasible points to SDP correspond to lower bounds on $\tau(\gamma_0)$.
- Feasible points to dual SDP correspond to upper bounds on $\tau(\gamma_0)$.
- Therefore:
  - proof via linear combinations of interpolation inequalities (evaluated at the iterates and $x_\star$),
  - proofs can be rewritten as a "sum-of-squares" certificates.

When does it work?

The methodology applies, as is, as soon as:

When does it work?

The methodology applies, as is, as soon as:

- ◇ performance measure and initial condition are linear in G,

When does it work?

The methodology applies, as is, as soon as:

- ◇ performance measure and initial condition are linear in G,
- ◇ interpolation inequalities are linear in G,

When does it work?

The methodology applies, as is, as soon as:

- ◇ performance measure and initial condition are linear in G,
- ◇ interpolation inequalities are linear in G,
- ◇ algorithm can be described linearly in G.

When does it work?

The methodology applies, as is, as soon as:

- ◇ performance measure and initial condition are linear in G,
- ◇ interpolation inequalities are linear in G,
- ◇ algorithm can be described linearly in G.

This applies to a variety of scenarios (as we discuss in the workshop).

When does it work?

The methodology applies, as is, as soon as:

- ◇ performance measure and initial condition are linear in G,
- ◇ interpolation inequalities are linear in G,
- ◇ algorithm can be described linearly in G.

This applies to a variety of scenarios (as we discuss in the workshop).

- ◇ check PEPit and PESTO (currently more than 75 examples);

When does it work?

The methodology applies, as is, as soon as:

- ◇ performance measure and initial condition are linear in G,
- ◇ interpolation inequalities are linear in G,
- ◇ algorithm can be described linearly in G.

This applies to a variety of scenarios (as we discuss in the workshop).

- ◇ check PEPit and PESTO (currently more than 75 examples);
- ◇ add yours ☺.

A few natural questions

A few natural questions

◇ What happens if one ingredient is not "nice" in G?

A few natural questions

- ◇ What happens if one ingredient is not "nice" in G?
  - – we can try convex relaxations,

A few natural questions

◇ What happens if one ingredient is not "nice" in G?

- we can try convex relaxations,
- for instance: no interpolation condition:

A few natural questions

- ◇ What happens if one ingredient is not "nice" in G?
  - – we can try convex relaxations,
  - – for instance: no interpolation condition:

    add all inequalities you are aware of,

A few natural questions

- ⋄ What happens if one ingredient is not "nice" in G?
  - – we can try convex relaxations,
  - – for instance: no interpolation condition:

    add all inequalities you are aware of,
    not necessarily evaluated only at the iterates and $x_\star$.

A few natural questions

- What happens if one ingredient is not "nice" in G?
  - we can try convex relaxations,
  - for instance: no interpolation condition:

    add all inequalities you are aware of,
    not necessarily evaluated only at the iterates and $x_\star$.
- Can we obtain "simple proofs" and worst-case examples?

A few natural questions

- ⋄ What happens if one ingredient is not "nice" in G?
  - – we can try convex relaxations,
  - – for instance: no interpolation condition:

    add all inequalities you are aware of,
    not necessarily evaluated only at the iterates and $x_\star$.
- ⋄ Can we obtain "simple proofs" and worst-case examples?
- ⋄ How to optimize the step sizes?

Recap'

Recap'

- ☺ Worst-case guarantees *cannot be improved*, systematic approach,

Recap'

- ☺ Worst-case guarantees *cannot be improved*, systematic approach,
- ☺ allows reaching proofs that could barely be obtained by hand,

Recap'

- ☺ Worst-case guarantees *cannot be improved*, systematic approach,
- ☺ allows reaching proofs that could barely be obtained by hand,
- ☺ fair amount of scenarios/algorithms (e.g., proximal terms, stochastic, etc.),

Recap'

- ☺ Worst-case guarantees *cannot be improved*, systematic approach,
- ☺ allows reaching proofs that could barely be obtained by hand,
- ☺ fair amount of scenarios/algorithms (e.g., proximal terms, stochastic, etc.),
- ☹ SDPs typically become prohibitively large in a variety of scenarios,

Recap'

- ☺ Worst-case guarantees *cannot be improved*, systematic approach,
- ☺ allows reaching proofs that could barely be obtained by hand,
- ☺ fair amount of scenarios/algorithms (e.g., proximal terms, stochastic, etc.),
- ☹ SDPs typically become prohibitively large in a variety of scenarios,
- ☹ transient behavior VS. asymptotic behavior: might be hard to distinguish with small N,

Recap'

- ☺ Worst-case guarantees *cannot be improved*, systematic approach,
- ☺ allows reaching proofs that could barely be obtained by hand,
- ☺ fair amount of scenarios/algorithms (e.g., proximal terms, stochastic, etc.),
- ☹ SDPs typically become prohibitively large in a variety of scenarios,
- ☹ transient behavior VS. asymptotic behavior: might be hard to distinguish with small N,
- ☹ proofs (may be) quite involved and hard to intuit,

Recap'

- ☺ Worst-case guarantees *cannot be improved*, systematic approach,
- ☺ allows reaching proofs that could barely be obtained by hand,
- ☺ fair amount of scenarios/algorithms (e.g., proximal terms, stochastic, etc.),
- ☹ SDPs typically become prohibitively large in a variety of scenarios,
- ☹ transient behavior VS. asymptotic behavior: might be hard to distinguish with small N,
- ☹ proofs (may be) quite involved and hard to intuit,
- ☹ proofs (may be) hard to generalize.

Reminders

Notions of simplicity

Designing methods

Concluding remarks

Simple counter-examples & proofs?

What is a simple counter-example?

Simple counter-examples & proofs?

What is a simple counter-example?

- ◇ low-dimensional,

Simple counter-examples & proofs?

What is a simple counter-example?

- ◇ low-dimensional,
- ◇ "simple" closed-form?

Simple counter-examples & proofs?

What is a simple counter-example?

- ◇ low-dimensional,
- ◇ “simple” closed-form?

What is a simple proof? Tentative answers:

Simple counter-examples & proofs?

What is a simple counter-example?

- ◇ low-dimensional,
- ◇ "simple" closed-form?

What is a simple proof? Tentative answers:

- ◇ uses few inequalities,

Simple counter-examples & proofs?

What is a simple counter-example?

- ◇ low-dimensional,
- ◇ "simple" closed-form?

What is a simple proof? Tentative answers:

- ◇ uses few inequalities,
- ◇ has few residual term (low-rank dual matrix),

Simple counter-examples & proofs?

What is a simple counter-example?

- ◇ low-dimensional,
- ◇ "simple" closed-form?

What is a simple proof? Tentative answers:

- ◇ uses few inequalities,
- ◇ has few residual term (low-rank dual matrix),
- ◇ has a nice structure (e.g., recursive)?

Low-dimensional examples

Two tricks:

Low-dimensional examples

Two tricks:

- ◇ minimize rank via trace heuristic: minimize $\mathrm{Tr}(G)$,

Low-dimensional examples

Two tricks:

- ◇ minimize rank via trace heuristic: minimize $\mathrm{Tr}(G)$,
- ◇ minimize rank via logdet heuristic: minimize log $\det(G)$.

Low-dimensional examples

Two tricks:

- minimize rank via trace heuristic: minimize $\mathrm{Tr}(G)$,
- minimize rank via logdet heuristic: minimize $\log\det(G)$.

Examples in PEPit!

Nice proof structure: Lyapunov/potential functions

Guarantees for gradient descent when minimizing an L-smooth convex function

$$f_\star = \min_{x \in \mathbb{R}^d} f(x)?$$

Nice proof structure: Lyapunov/potential functions

Guarantees for gradient descent when minimizing an L-smooth convex function

$$f_\star = \min_{x \in \mathbb{R}^d} f(x)?$$

Known that $f(x_N) - f_\star = O(\frac{1}{N})$ with small enough step sizes (e.g., $\frac{1}{L}$).

Nice proof structure: Lyapunov/potential functions

Guarantees for gradient descent when minimizing an L-smooth convex function

$$f_\star = \min_{x \in \mathbb{R}^d} f(x)?$$

Known that $f(x_N) - f_\star = O(\frac{1}{N})$ with small enough step sizes (e.g., $\frac{1}{L}$).

For all L-smooth convex f, $x_k \in \mathbb{R}^d$, and $k \geqslant 0$, easy to show $\phi_{k+1}^f \leqslant \phi_k^f$ with

$$\phi_k^f = k(f(x_k) - f_\star) + \frac{L}{2}\|x_k - x_\star\|^2 \text{ \textit{(potential at iteration k)}},$$

see e.g., (Bansal & Gupta 2017).

Nice proof structure: Lyapunov/potential functions

Guarantees for gradient descent when minimizing an L-smooth convex function

$$f_\star = \min_{x \in \mathbb{R}^d} f(x)?$$

Known that $f(x_N) - f_\star = O(\frac{1}{N})$ with small enough step sizes (e.g., $\frac{1}{L}$).

For all L-smooth convex f, $x_k \in \mathbb{R}^d$, and $k \geqslant 0$, easy to show $\phi_{k+1}^f \leqslant \phi_k^f$ with

$$\phi_k^f = k(f(x_k) - f_\star) + \frac{L}{2}\|x_k - x_\star\|^2 \text{ \textit{(potential at iteration k)}},$$

see e.g., (Bansal & Gupta 2017).

Why is that nice? Very simple resulting proof:

Nice proof structure: Lyapunov/potential functions

Guarantees for gradient descent when minimizing an L-smooth convex function

$$f_\star = \min_{x \in \mathbb{R}^d} f(x)?$$

Known that $f(x_N) - f_\star = O(\frac{1}{N})$ with small enough step sizes (e.g., $\frac{1}{L}$).

For all L-smooth convex f, $x_k \in \mathbb{R}^d$, and $k \geqslant 0$, easy to show $\phi_{k+1}^f \leqslant \phi_k^f$ with

$$\phi_k^f = k(f(x_k) - f_\star) + \frac{L}{2}\|x_k - x_\star\|^2 \text{ (potential at iteration } k\text{)},$$

see e.g., (Bansal & Gupta 2017).

Why is that nice? Very simple resulting proof:

$$\phi_N^f \leqslant \phi_{N-1}^f \leqslant \ldots \leqslant \phi_0^f$$

Nice proof structure: Lyapunov/potential functions

Guarantees for gradient descent when minimizing an L-smooth convex function

$$f_\star = \min_{x \in \mathbb{R}^d} f(x)?$$

Known that $f(x_N) - f_\star = O(\frac{1}{N})$ with small enough step sizes (e.g., $\frac{1}{L}$).

For all L-smooth convex f, $x_k \in \mathbb{R}^d$, and $k \geqslant 0$, easy to show $\phi_{k+1}^f \leqslant \phi_k^f$ with

$$\phi_k^f = k(f(x_k) - f_\star) + \frac{L}{2}\|x_k - x_\star\|^2$$ *(potential at iteration k)*,

see e.g., (Bansal & Gupta 2017).

Why is that nice? Very simple resulting proof:

$$N(f(x_N) - f_\star) \leqslant \phi_N^f \leqslant \phi_{N-1}^f \leqslant \ldots \leqslant \phi_0^f$$

Nice proof structure: Lyapunov/potential functions

Guarantees for gradient descent when minimizing an L-smooth convex function

$$f_\star = \min_{x \in \mathbb{R}^d} f(x)?$$

Known that $f(x_N) - f_\star = O(\frac{1}{N})$ with small enough step sizes (e.g., $\frac{1}{L}$).

For all L-smooth convex f, $x_k \in \mathbb{R}^d$, and $k \geqslant 0$, easy to show $\phi^f_{k+1} \leqslant \phi^f_k$ with

$$\phi^f_k = k(f(x_k) - f_\star) + \tfrac{L}{2}\|x_k - x_\star\|^2 \text{ \textit{(potential at iteration k)}},$$

see e.g., (Bansal & Gupta 2017).

Why is that nice? Very simple resulting proof:

$$N(f(x_N) - f_\star) \leqslant \phi^f_N \leqslant \phi^f_{N-1} \leqslant \ldots \leqslant \phi^f_0 = \tfrac{L}{2}\|x_0 - x_\star\|^2,$$

Nice proof structure: Lyapunov/potential functions

Guarantees for gradient descent when minimizing an L-smooth convex function

$$f_\star = \min_{x\in\mathbb{R}^d} f(x)?$$

Known that $f(x_N) - f_\star = O(\frac{1}{N})$ with small enough step sizes (e.g., $\frac{1}{L}$).

For all L-smooth convex f, $x_k \in \mathbb{R}^d$, and $k \geqslant 0$, easy to show $\phi_{k+1}^f \leqslant \phi_k^f$ with

$$\phi_k^f = k(f(x_k) - f_\star) + \tfrac{L}{2}\|x_k - x_\star\|^2 \text{ \textit{(potential at iteration k)}},$$

see e.g., (Bansal & Gupta 2017).

Why is that nice? Very simple resulting proof:

$$N(f(x_N) - f_\star) \leqslant \phi_N^f \leqslant \phi_{N-1}^f \leqslant \ldots \leqslant \phi_0^f = \tfrac{L}{2}\|x_0 - x_\star\|^2,$$

hence: $f(x_N) - f_\star \leqslant \frac{L\|x_0-x_\star\|^2}{2N}$.

How does it work for the gradient method?

Gradient descent, take II: how to bound $\|\nabla f(x_N)\|^2$ using potentials?

How does it work for the gradient method?

Gradient descent, take II: how to bound $\|\nabla f(x_N)\|^2$ using potentials?

Key idea: forget how x_k was generated and prove $\phi_{k+1}^f \leqslant \phi_k^f$.

- ☺ only need to study one iteration
- ☹ where does this ϕ_k^f comes from!? (structure and dependence on k)

How does it work for the gradient method?

Gradient descent, take II: how to bound $\|\nabla f(x_N)\|^2$ using potentials?

Key idea: forget how x_k was generated and prove $\phi_{k+1}^f \leqslant \phi_k^f$.

- ☺ only need to study one iteration
- ☹ where does this ϕ_k^f comes from!? (structure and dependence on k)

Starting point: candidate quadratic ϕ_k^f with *all the available information* at iteration k

$$\phi_k^f = a_k \|x_k - x_\star\|^2 + b_k \|\nabla f(x_k)\|^2 + 2c_k \langle \nabla f(x_k), x_k - x_\star\rangle + d_k (f(x_k) - f_\star).$$

How does it work for the gradient method?

Gradient descent, take II: how to bound $\|\nabla f(x_N)\|^2$ using potentials?

Key idea: forget how x_k was generated and prove $\phi^f_{k+1} \leqslant \phi^f_k$.

- ☺ only need to study one iteration
- ☹ where does this ϕ^f_k comes from!? (structure and dependence on k)

Starting point: candidate quadratic ϕ^f_k with *all the available information* at iteration k

$$\phi^f_k = a_k \|x_k - x_\star\|^2 + b_k \|\nabla f(x_k)\|^2 + 2c_k \langle \nabla f(x_k), x_k - x_\star \rangle + d_k \left(f(x_k) - f_\star\right).$$

How to choose a_k, b_k, c_k, d_k's?

How does it work for the gradient method?

Gradient descent, take II: how to bound $\|\nabla f(x_N)\|^2$ using potentials?

Key idea: forget how x_k was generated and prove $\phi_{k+1}^f \leqslant \phi_k^f$.

- ☺ only need to study one iteration
- ☹ where does this ϕ_k^f comes from!? (structure and dependence on k)

Starting point: candidate quadratic ϕ_k^f with *all the available information* at iteration k

$$\phi_k^f = a_k \|x_k - x_\star\|^2 + b_k \|\nabla f(x_k)\|^2 + 2c_k \langle \nabla f(x_k), x_k - x_\star \rangle + d_k \left(f(x_k) - f_\star\right).$$

How to choose a_k, b_k, c_k, d_k's?

1. choice should satisfy "$\phi_{k+1}^f \leqslant \phi_k^f$",

How does it work for the gradient method?

Gradient descent, take II: how to bound $\|\nabla f(x_N)\|^2$ using potentials?

Key idea: forget how x_k was generated and prove $\phi_{k+1}^f \leqslant \phi_k^f$.

- ☺ only need to study one iteration
- ☹ where does this ϕ_k^f comes from!? (structure and dependence on k)

Starting point: candidate quadratic ϕ_k^f with *all the available information* at iteration k

$$\phi_k^f = a_k \|x_k - x_\star\|^2 + b_k \|\nabla f(x_k)\|^2 + 2c_k \langle \nabla f(x_k), x_k - x_\star \rangle + d_k (f(x_k) - f_\star).$$

How to choose a_k, b_k, c_k, d_k's?

1. choice should satisfy "$\phi_{k+1}^f \leqslant \phi_k^f$",
2. choice should result in bound on $\|\nabla f(x_N)\|^2$.

How does it work for the gradient method?

Given ϕ^f_{k+1}, ϕ^f_k, *how to verify* that for all L-smooth convex f, $x_k \in \mathbb{R}^d$, and $d \in \mathbb{N}$:

$$\phi^f_{k+1} \leqslant \phi^f_k?$$

How does it work for the gradient method?

Given ϕ^f_{k+1}, ϕ^f_k, *how to verify* that for all L-smooth convex f, $x_k \in \mathbb{R}^d$, and $d \in \mathbb{N}$:

$$\phi^f_{k+1} \leqslant \phi^f_k?$$

(notations: the set of such pairs (ϕ^f_k, ϕ^f_{k+1}) is denoted $\mathcal{V}_k$.)

How does it work for the gradient method?

Given ϕ_{k+1}^f, ϕ_k^f, *how to verify* that for all L-smooth convex f, $x_k \in \mathbb{R}^d$, and $d \in \mathbb{N}$:

$$\phi_{k+1}^f \leqslant \phi_k^f?$$

(notations: the set of such pairs (ϕ_k^f, ϕ_{k+1}^f) is denoted $\mathcal{V}_k$.)

Answer:

$$\phi_{k+1}^f \leqslant \phi_k^f \text{ for all } L\text{-smooth convex } f,\ x_k \in \mathbb{R}^d, \text{ and } d \in \mathbb{N}$$

$$\Leftrightarrow$$

some small-sized *linear matrix inequality (LMI)* is feasible.

How does it work for the gradient method?

Given ϕ^f_{k+1}, ϕ^f_k, *how to verify* that for all L-smooth convex f, $x_k \in \mathbb{R}^d$, and $d \in \mathbb{N}$:

$$\phi^f_{k+1} \leqslant \phi^f_k?$$

(notations: the set of such pairs (ϕ^f_k, ϕ^f_{k+1}) is denoted $\mathcal{V}_k$.)

Answer:

$$\phi^f_{k+1} \leqslant \phi^f_k \text{ for all } L\text{-smooth convex } f,\ x_k \in \mathbb{R}^d, \text{ and } d \in \mathbb{N}$$

$$\Leftrightarrow$$

some small-sized *linear matrix inequality (LMI)* is feasible.

Furthermore: LMI is linear in parameters $\{a_k, b_k, c_k, d_k\}_k$.

How does it work for the gradient method?

Given ϕ^f_{k+1}, ϕ^f_k, *how to verify* that for all L-smooth convex f, $x_k \in \mathbb{R}^d$, and $d \in \mathbb{N}$:

$$\phi^f_{k+1} \leqslant \phi^f_k?$$

(notations: the set of such pairs (ϕ^f_k, ϕ^f_{k+1}) is denoted $\mathcal{V}_k$.)

Answer:

$$\phi^f_{k+1} \leqslant \phi^f_k \text{ for all } L\text{-smooth convex } f,\ x_k \in \mathbb{R}^d, \text{ and } d \in \mathbb{N}$$

$$\Leftrightarrow$$

some small-sized *linear matrix inequality (LMI)* is feasible.

Furthermore: LMI is linear in parameters $\{a_k, b_k, c_k, d_k\}_k$.

In others words: *efficient (convex) representation of $\mathcal{V}_k$ available*!

How does it work for the gradient method?

Recap: we want to bound $\|\nabla f(x_N)\|^2$; choose

How does it work for the gradient method?

Recap: we want to bound $\|\nabla f(x_N)\|^2$; choose

$$\phi_k^f = a_k \|x_k - x_\star\|^2 + b_k \|\nabla f(x_k)\|^2 + 2c_k \langle \nabla f(x_k), x_k - x_\star \rangle + d_k (f(x_k) - f_\star).$$

How does it work for the gradient method?

Recap: we want to bound $\|\nabla f(x_N)\|^2$; choose

$$\phi_k^f = a_k \|x_k - x_\star\|^2 + b_k \|\nabla f(x_k)\|^2 + 2c_k \langle \nabla f(x_k), x_k - x_\star \rangle + d_k (f(x_k) - f_\star).$$

with $\phi_0^f = L^2 \|x_0 - x_\star\|^2$ and $\phi_N^f = b_N \|\nabla f(x_N)\|^2$.

How does it work for the gradient method?

Recap: we want to bound $\|\nabla f(x_N)\|^2$; choose

$$\phi_k^f = a_k \|x_k - x_\star\|^2 + b_k \|\nabla f(x_k)\|^2 + 2c_k \langle \nabla f(x_k), x_k - x_\star \rangle + d_k (f(x_k) - f_\star).$$

with $\phi_0^f = L^2\|x_0 - x_\star\|^2$ and $\phi_N^f = b_N \|\nabla f(x_N)\|^2$.

Motivation: this structure would result in $\|\nabla f(x_N)\|^2 \leqslant \frac{L^2\|x_0 - x_\star\|^2}{b_N}$.

How does it work for the gradient method?

Recap: we want to bound $\|\nabla f(x_N)\|^2$; choose

$$\phi_k^f = a_k \|x_k - x_\star\|^2 + b_k \|\nabla f(x_k)\|^2 + 2c_k \langle \nabla f(x_k), x_k - x_\star \rangle + d_k (f(x_k) - f_\star).$$

with $\phi_0^f = L^2\|x_0 - x_\star\|^2$ and $\phi_N^f = b_N \|\nabla f(x_N)\|^2$.

Motivation: this structure would result in $\|\nabla f(x_N)\|^2 \leqslant \frac{L^2\|x_0 - x_\star\|^2}{b_N}$.

Question: largest provable b_N using such potentials?

How does it work for the gradient method?

Recap: we want to bound $\|\nabla f(x_N)\|^2$; choose

$$\phi_k^f = a_k \|x_k - x_\star\|^2 + b_k \|\nabla f(x_k)\|^2 + 2c_k \langle \nabla f(x_k), x_k - x_\star \rangle + d_k (f(x_k) - f_\star).$$

with $\phi_0^f = L^2\|x_0 - x_\star\|^2$ and $\phi_N^f = b_N \|\nabla f(x_N)\|^2$.

Motivation: this structure would result in $\|\nabla f(x_N)\|^2 \leqslant \frac{L^2\|x_0 - x_\star\|^2}{b_N}$.

Question: largest provable b_N using such potentials?

$$\max_{\phi_1^f,\ldots,\phi_{N-1}^f, b_N} b_N \text{ such that } (\phi_0^f, \phi_1^f) \in \mathcal{V}_0, \ldots, (\phi_{N-1}^f, \phi_N^f) \in \mathcal{V}_{N-1}$$

How does it work for the gradient method?

Recap: we want to bound $\|\nabla f(x_N)\|^2$; choose

$$\phi_k^f = a_k \|x_k - x_\star\|^2 + b_k \|\nabla f(x_k)\|^2 + 2c_k \langle \nabla f(x_k), x_k - x_\star \rangle + d_k (f(x_k) - f_\star).$$

with $\phi_0^f = L^2\|x_0 - x_\star\|^2$ and $\phi_N^f = b_N \|\nabla f(x_N)\|^2$.

Motivation: this structure would result in $\|\nabla f(x_N)\|^2 \leqslant \frac{L^2\|x_0 - x_\star\|^2}{b_N}$.

Question: largest provable b_N using such potentials?

$$\max_{\phi_1^f,\ldots,\phi_{N-1}^f,b_N} b_N \text{ such that } (\phi_0^f, \phi_1^f) \in \mathcal{V}_0, \ldots, (\phi_{N-1}^f, \phi_N^f) \in \mathcal{V}_{N-1}$$

Let's engineer a worst-case guarantee:

How does it work for the gradient method?

Recap: we want to bound $\|\nabla f(x_N)\|^2$; choose

$$\phi_k^f = a_k \|x_k - x_\star\|^2 + b_k \|\nabla f(x_k)\|^2 + 2c_k \langle \nabla f(x_k), x_k - x_\star \rangle + d_k (f(x_k) - f_\star).$$

with $\phi_0^f = L^2\|x_0 - x_\star\|^2$ and $\phi_N^f = b_N \|\nabla f(x_N)\|^2$.

Motivation: this structure would result in $\|\nabla f(x_N)\|^2 \leqslant \frac{L^2\|x_0 - x_\star\|^2}{b_N}$.

Question: largest provable b_N using such potentials?

$$\max_{\phi_1^f,\ldots,\phi_{N-1}^f, b_N} b_N \text{ such that } (\phi_0^f, \phi_1^f) \in \mathcal{V}_0, \ldots, (\phi_{N-1}^f, \phi_N^f) \in \mathcal{V}_{N-1}$$

Let's engineer a worst-case guarantee:

1. Solve the SDP for some values of N.

How does it work for the gradient method?

Recap: we want to bound $\|\nabla f(x_N)\|^2$; choose

$$\phi_k^f = a_k \|x_k - x_\star\|^2 + b_k \|\nabla f(x_k)\|^2 + 2c_k \langle \nabla f(x_k), x_k - x_\star \rangle + d_k (f(x_k) - f_\star).$$

with $\phi_0^f = L^2\|x_0 - x_\star\|^2$ and $\phi_N^f = b_N \|\nabla f(x_N)\|^2$.

Motivation: this structure would result in $\|\nabla f(x_N)\|^2 \leqslant \frac{L^2\|x_0 - x_\star\|^2}{b_N}$.

Question: largest provable b_N using such potentials?

$$\max_{\phi_1^f,\ldots,\phi_{N-1}^f, b_N} b_N \text{ such that } (\phi_0^f, \phi_1^f) \in \mathcal{V}_0, \ldots, (\phi_{N-1}^f, \phi_N^f) \in \mathcal{V}_{N-1}$$

Let's engineer a worst-case guarantee:

1. Solve the SDP for some values of N.
2. Observe the a_k, b_k, c_k, d_k's for some values of N.

How does it work for the gradient method?

Recap: we want to bound $\|\nabla f(x_N)\|^2$; choose

$$\phi_k^f = a_k \|x_k - x_\star\|^2 + b_k \|\nabla f(x_k)\|^2 + 2c_k \langle \nabla f(x_k), x_k - x_\star\rangle + d_k (f(x_k) - f_\star).$$

with $\phi_0^f = L^2\|x_0 - x_\star\|^2$ and $\phi_N^f = b_N \|\nabla f(x_N)\|^2$.

Motivation: this structure would result in $\|\nabla f(x_N)\|^2 \leqslant \frac{L^2\|x_0-x_\star\|^2}{b_N}$.

Question: largest provable b_N using such potentials?

$$\max_{\phi_1^f,\ldots,\phi_{N-1}^f, b_N} b_N \text{ such that } (\phi_0^f, \phi_1^f) \in \mathcal{V}_0, \ldots, (\phi_{N-1}^f, \phi_N^f) \in \mathcal{V}_{N-1}$$

Let's engineer a worst-case guarantee:

1. Solve the SDP for some values of N.
2. Observe the a_k, b_k, c_k, d_k's for some values of N.
3. Try to simplify the ϕ_k^f's without loosing too much.

How does it work for the gradient method?

Recap: we want to bound $\|\nabla f(x_N)\|^2$; choose

$$\phi_k^f = a_k \|x_k - x_\star\|^2 + b_k \|\nabla f(x_k)\|^2 + 2c_k \langle \nabla f(x_k), x_k - x_\star \rangle + d_k (f(x_k) - f_\star).$$

with $\phi_0^f = L^2\|x_0 - x_\star\|^2$ and $\phi_N^f = b_N \|\nabla f(x_N)\|^2$.

Motivation: this structure would result in $\|\nabla f(x_N)\|^2 \leqslant \frac{L^2\|x_0 - x_\star\|^2}{b_N}$.

Question: largest provable b_N using such potentials?

$$\max_{\phi_1^f,\ldots,\phi_{N-1}^f, b_N} b_N \text{ such that } (\phi_0^f, \phi_1^f) \in \mathcal{V}_0, \ldots, (\phi_{N-1}^f, \phi_N^f) \in \mathcal{V}_{N-1}$$

Let's engineer a worst-case guarantee:

1. Solve the SDP for some values of N.
2. Observe the a_k, b_k, c_k, d_k's for some values of N.
3. Try to simplify the ϕ_k^f's without loosing too much.
4. Prove target result by analytically playing with $\mathcal{V}_k$ (i.e., study single iteration).

How does it work for the gradient method?

1. Solve the SDP for some values of N; recall final guarantee of the form:

$$\|\nabla f(x_N)\|^2 \leqslant \frac{L^2\,\|x_0-x_\star\|^2}{b_N}$$

$$N = $$
$$b_N = $$

How does it work for the gradient method?

1. Solve the SDP for some values of N; recall final guarantee of the form:

$$\|\nabla f(x_N)\|^2 \leqslant \frac{L^2 \|x_0 - x_\star\|^2}{b_N}$$

$$\begin{aligned} N &= \quad 1 \\ b_N &= \end{aligned}$$

How does it work for the gradient method?

1. Solve the SDP for some values of N; recall final guarantee of the form:

$$\|\nabla f(x_N)\|^2 \leqslant \frac{L^2\,\|x_0 - x_\star\|^2}{b_N}$$

$$\begin{aligned} N &= 1 \\ b_N &= 4 \end{aligned}$$

How does it work for the gradient method?

1. Solve the SDP for some values of N; recall final guarantee of the form:

$$\|\nabla f(x_N)\|^2 \leqslant \frac{L^2\,\|x_0 - x_\star\|^2}{b_N}$$

$N =$	1	2
$b_N =$	4	9

How does it work for the gradient method?

1. Solve the SDP for some values of N; recall final guarantee of the form:

$$\|\nabla f(x_N)\|^2 \leqslant \frac{L^2\,\|x_0 - x_\star\|^2}{b_N}$$

$N =$	1	2	3
$b_N =$	4	9	16

How does it work for the gradient method?

1. Solve the SDP for some values of N; recall final guarantee of the form:

$$\|\nabla f(x_N)\|^2 \leqslant \frac{L^2\,\|x_0 - x_\star\|^2}{b_N}$$

$N =$	1	2	3	4	...	100
$b_N =$	4	9	16	25	...	10201

How does it work for the gradient method?

1. Solve the SDP for some values of N; recall final guarantee of the form:

$$\|\nabla f(x_N)\|^2 \leqslant \frac{L^2 \|x_0 - x_\star\|^2}{b_N}$$

$N =$	1	2	3	4	...	100
$b_N =$	4	9	16	25	...	10201

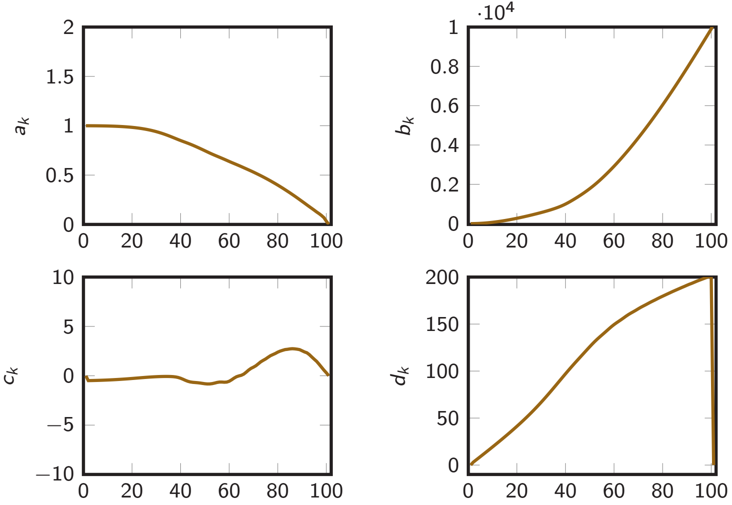
2. Observe the a_k, b_k, c_k, d_k's for some values of N.

Fixed horizon $N = 100$, $L = 1$, and

$$\phi_k^f = a_k \|x_k - x_\star\|^2 + b_k \|\nabla f(x_k)\|^2 + 2c_k \langle \nabla f(x_k), x_k - x_\star \rangle + d_k (f(x_k) - f_\star).$$

Fixed horizon $N = 100$, $L = 1$, and

$$\phi_k^f = a_k \|x_k - x_\star\|^2 + b_k \|\nabla f(x_k)\|^2 + 2c_k \langle \nabla f(x_k), x_k - x_\star \rangle + d_k (f(x_k) - f_\star).$$



How does it work for the gradient method?

1. Solve the SDP for some values of N; recall final guarantee of the form:

$$\|\nabla f(x_N)\|^2 \leqslant \frac{L^2 \, \|x_0 - x_\star\|^2}{b_N}$$

$N =$	1	2	3	4	...	100
$b_N =$	4	9	16	25	...	10201

2. Observe the a_k, b_k, c_k, d_k's for some values of N.

How does it work for the gradient method?

1. Solve the SDP for some values of N; recall final guarantee of the form:

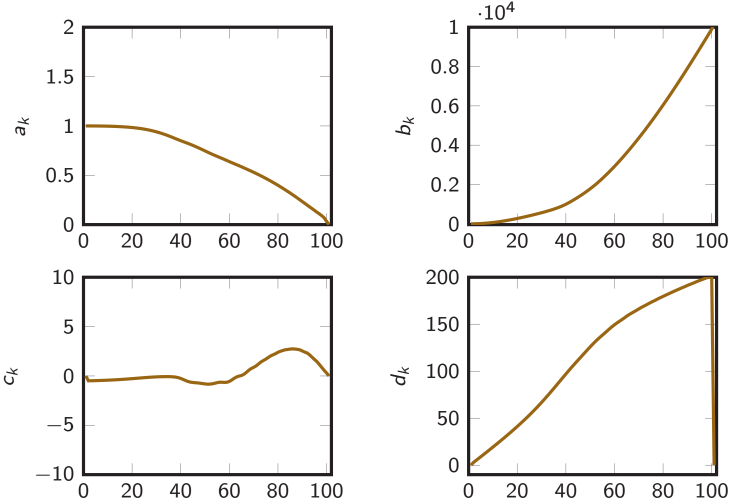
$$\|\nabla f(x_N)\|^2 \leqslant \frac{L^2 \|x_0 - x_\star\|^2}{b_N}$$

$N =$	1	2	3	4	...	100
$b_N =$	4	9	16	25	...	10201

2. Observe the a_k, b_k, c_k, d_k's for some values of N.
3. Try to simplify the ϕ_k^f's without loosing too much.

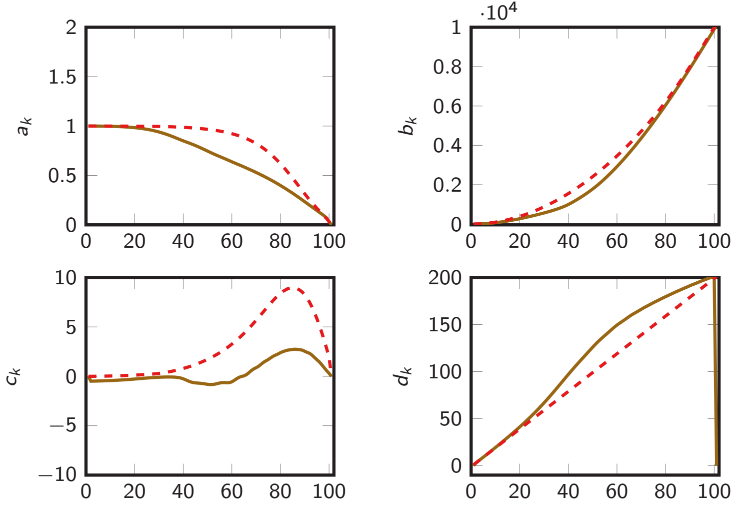
Fixed horizon $N = 100$ and

$$V_k = \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix}^\top \left[\begin{pmatrix} a_k & c_k \\ c_k & b_k \end{pmatrix} \otimes I_d \right] \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix} + d_k \left(f(x_k) - f_\star \right)$$



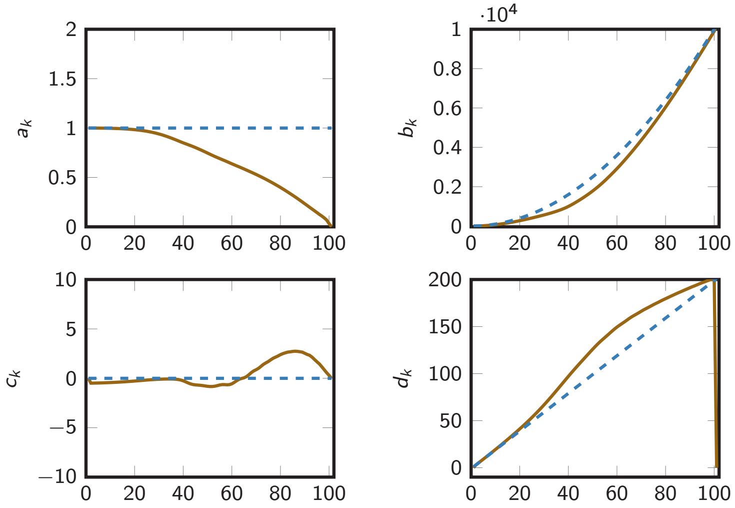
Fixed horizon $N = 100$ and

$$V_k = \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix}^\top \left[\begin{pmatrix} a_k & c_k \\ c_k & b_k \end{pmatrix} \otimes I_d \right] \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix} + (2k+1)L\left(f(x_k) - f_\star\right)$$



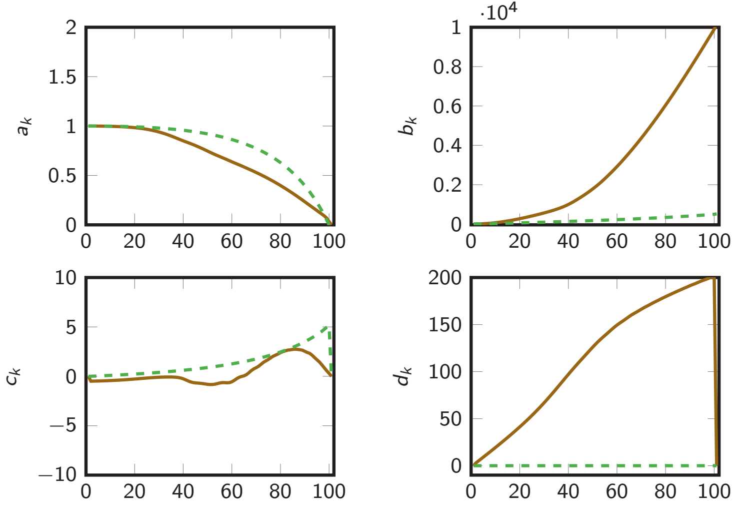
Fixed horizon $N = 100$ and

$$V_k = \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix}^\top \left[\begin{pmatrix} L^2 & 0 \\ 0 & b_k \end{pmatrix} \otimes I_d \right] \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix} + (2k+1)L\left(f(x_k) - f_\star\right)$$



Fixed horizon $N = 100$ and

$$V_k = \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix}^\top \left[\begin{pmatrix} a_k & c_k \\ c_k & b_k \end{pmatrix} \otimes I_d \right] \begin{pmatrix} x_k - x_\star \\ \nabla f(x_k) \end{pmatrix} + 0\,(f(x_k) - f_\star)$$



How does it work for the gradient method?

1. Solve the SDP for some values of N; recall final guarantee of the form:

$$\|\nabla f(x_N)\|^2 \leqslant \frac{L^2\,\|x_0 - x_\star\|^2}{b_N}$$

$$\begin{array}{rcccccc} N = & 1 & 2 & 3 & 4 & \ldots & 100 \\ b_N = & 4 & 9 & 16 & 25 & \ldots & 10201 \end{array}$$

2. Observe the a_k, b_k, c_k, d_k's for some values of N.
3. Try to simplify the ϕ_k^f's without loosing too much.
   Simplification attempt #1: $d_k = (2k+1)L$
   Simplification attempt #2: $a_k = L^2$ and $c_k = 0$
   Simplification attempt #3: $d_k = 0$

How does it work for the gradient method?

1. Solve the SDP for some values of N; recall final guarantee of the form:

$$\|\nabla f(x_N)\|^2 \leqslant \frac{L^2 \|x_0 - x_\star\|^2}{b_N}$$

$$\begin{array}{rcccccc} N = & 1 & 2 & 3 & 4 & \dots & 100 \\ b_N = & 4 & 9 & 16 & 25 & \dots & 10201 \end{array}$$

2. Observe the a_k, b_k, c_k, d_k's for some values of N.
3. Try to simplify the ϕ_k^f's without loosing too much.
   Simplification attempt #1: $d_k = (2k+1)L$
   Simplification attempt #2: $a_k = L^2$ and $c_k = 0$
   Simplification attempt #3: $d_k = 0$
4. Prove target result by analytically playing with $\mathcal{V}_k$:

$$\phi_k^f(x_k) = (2k+1)L(f(x_k) - f_\star) + k(k+2)\|\nabla f(x_k)\|^2 + L^2\|x_k - x_\star\|^2,$$

hence $f(x_N) - f_\star = O(N^{-1})$ and $\|\nabla f(x_N)\|^2 = O(N^{-2})$.

Lyapunov/potential functions

Allows studying more "complicated" methods:

- stochastic structures,
- randomized structures.

Lyapunov/potential functions

Allows studying more “complicated” methods:

- ⋄ stochastic structures,
- ⋄ randomized structures.

Allows gaining intuitions, examples:

- ⋄ optimized gradient method,
- ⋄ triple momentum method,
- ⋄ information-theoretic exact method.

Informal link with “full” PEPs?

How does this strategy compare to regular “N-iteration” PEPs?

Informal link with "full" PEPs?

How does this strategy compare to regular "N-iteration" PEPs?

Example: matrix of dual variables $[\lambda_{i,j}]$:

Informal link with “full” PEPs?

How does this strategy compare to regular “N-iteration” PEPs?

Example: matrix of dual variables $[\lambda_{i,j}]$:

- ◇ 1-smooth convex minimization, gradient descent with $\gamma = 1$,

Informal link with "full" PEPs?

How does this strategy compare to regular "N-iteration" PEPs?

Example: matrix of dual variables $[\lambda_{i,j}]$:

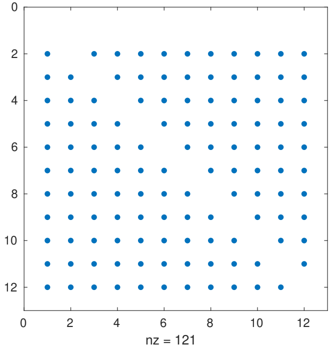
- $\diamond$ 1-smooth convex minimization, gradient descent with $\gamma = 1$,
- $\diamond$ worst-case of $\frac{f(x_N) - f_\star}{\|x_0 - x_\star\|^2}$.

Informal link with "full" PEPs?

How does this strategy compare to regular "N-iteration" PEPs?

Example: matrix of dual variables $[\lambda_{i,j}]$:

- ◇ 1-smooth convex minimization, gradient descent with $\gamma = 1$,
- ◇ worst-case of $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$.



Informal link with "full" PEPs?

How does this strategy compare to regular "N-iteration" PEPs?

Example: matrix of dual variables $[\lambda_{i,j}]$:

Informal link with "full" PEPs?

How does this strategy compare to regular "N-iteration" PEPs?

Example: matrix of dual variables $[\lambda_{i,j}]$:

- ◇ 1-smooth convex minimization, **optimized gradient descent**,

Informal link with "full" PEPs?

How does this strategy compare to regular "N-iteration" PEPs?

Example: matrix of dual variables $[\lambda_{i,j}]$:

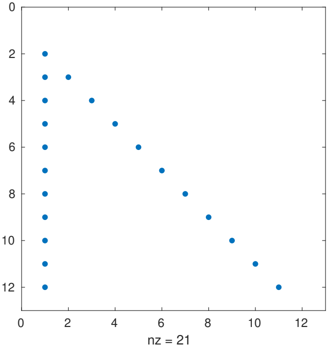
- ◇ 1-smooth convex minimization, **optimized gradient descent**,
- ◇ worst-case of $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$.

Informal link with "full" PEPs?

How does this strategy compare to regular "N-iteration" PEPs?

Example: matrix of dual variables $[\lambda_{i,j}]$:

- ◇ 1-smooth convex minimization, **optimized gradient descent**,
- ◇ worst-case of $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$.



More about Lyapunov approaches

"Tight Lyapunov function existence analysis for first-order methods"

More about Lyapunov approaches

"Tight Lyapunov function existence analysis for first-order methods"

Manu
Upadhyaya

Sebastian
Banert

Pontus
Giselsson

More about Lyapunov approaches

"Tight Lyapunov function existence analysis for first-order methods"

Manu Upadhyaya

Sebastian Banert

Pontus Giselsson

... tomorrow!

Reminders

Notions of simplicity

Designing methods

Concluding remarks

Designing methods

Two main PEP-related techniques:

Designing methods

Two main PEP-related techniques:

- minimax

Designing methods

Two main PEP-related techniques:

- minimax
- subspace search elimination.

Creating new algorithms via minimax approach

Smooth (strongly) convex minimization with more than gradient descent?

Creating new algorithms via minimax approach

Smooth (strongly) convex minimization with more than gradient descent?

$$x_1 = x_0 - h_{1,0}\nabla f(x_0)$$

Creating new algorithms via minimax approach

Smooth (strongly) convex minimization with more than gradient descent?

$$x_1 = x_0 - h_{1,0}\nabla f(x_0)$$
$$x_2 = x_1 - h_{2,0}\nabla f(x_0) - h_{2,1}\nabla f(x_1)$$

Creating new algorithms via minimax approach

Smooth (strongly) convex minimization with more than gradient descent?

$$
\begin{aligned}
x_1 &= x_0 - h_{1,0}\nabla f(x_0)\\
x_2 &= x_1 - h_{2,0}\nabla f(x_0) - h_{2,1}\nabla f(x_1)\\
&\vdots
\end{aligned}
$$

Creating new algorithms via minimax approach

Smooth (strongly) convex minimization with more than gradient descent?

$$
\begin{aligned}
x_1 &= x_0 - h_{1,0}\nabla f(x_0) \\
x_2 &= x_1 - h_{2,0}\nabla f(x_0) - h_{2,1}\nabla f(x_1) \\
&\vdots \\
x_N &= x_{N-1} - h_{N,0}\nabla f(x_0) - \ldots - h_{N,N-1}\nabla f(x_{N-1})
\end{aligned}
$$

Creating new algorithms via minimax approach

Smooth (strongly) convex minimization with more than gradient descent?

$$
\begin{aligned}
x_1 &= x_0 - h_{1,0}\nabla f(x_0)\\
x_2 &= x_1 - h_{2,0}\nabla f(x_0) - h_{2,1}\nabla f(x_1)\\
&\vdots\\
x_N &= x_{N-1} - h_{N,0}\nabla f(x_0) - \ldots - h_{N,N-1}\nabla f(x_{N-1})
\end{aligned}
$$

How to choose $\{h_{i,j}\}$?

Creating new algorithms via minimax approach

Smooth (strongly) convex minimization with more than gradient descent?

$$
\begin{aligned}
x_1 &= x_0 - h_{1,0}\nabla f(x_0)\\
x_2 &= x_1 - h_{2,0}\nabla f(x_0) - h_{2,1}\nabla f(x_1)\\
&\vdots\\
x_N &= x_{N-1} - h_{N,0}\nabla f(x_0) - \ldots - h_{N,N-1}\nabla f(x_{N-1})
\end{aligned}
$$

How to choose $\{h_{i,j}\}$?

- pick a performance criterion, for instance

$$\frac{\|x_N - x_\star\|^2}{\|x_0 - x_\star\|^2},$$

Creating new algorithms via minimax approach

Smooth (strongly) convex minimization with more than gradient descent?

$$
\begin{aligned}
x_1 &= x_0 - h_{1,0}\nabla f(x_0) \\
x_2 &= x_1 - h_{2,0}\nabla f(x_0) - h_{2,1}\nabla f(x_1) \\
&\vdots \\
x_N &= x_{N-1} - h_{N,0}\nabla f(x_0) - \ldots - h_{N,N-1}\nabla f(x_{N-1})
\end{aligned}
$$

How to choose $\{h_{i,j}\}$?

- ◇ pick a performance criterion, for instance

$$\frac{\|x_N - x_\star\|^2}{\|x_0 - x_\star\|^2},$$

- ◇ solve the minimax:

$$\min_{\{h_{i,j}\}_{i,j}} \max_{f\in\mathcal{F},\{x_i\}} \frac{\|x_N - x_\star\|^2}{\|x_0 - x_\star\|^2}.$$

Creating new algorithms via minimax approach

Situation seems quite involved in general, apart from a few cases

Creating new algorithms via minimax approach

Situation seems quite involved in general, apart from a few cases

◇ $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu = 0$: optimized gradient method (OGM, Kim & Fessler 2016),

Creating new algorithms via minimax approach

Situation seems quite involved in general, apart from a few cases

- $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu=0$: optimized gradient method (OGM, Kim & Fessler 2016),
- $\frac{\|x_N-x_\star\|^2}{\|x_0-x_\star\|^2}$: information-theoretic exact method (ITEM, T & Drori 2021),

Creating new algorithms via minimax approach

Situation seems quite involved in general, apart from a few cases

- ◇ $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu = 0$: optimized gradient method (OGM, Kim & Fessler 2016),
- ◇ $\frac{\|x_N-x_\star\|^2}{\|x_0-x_\star\|^2}$: information-theoretic exact method (ITEM, T & Drori 2021),
- ◇ $\frac{\|\nabla f(x_N)\|^2}{f(x_0)-f_\star}$ with $\mu = 0$: OGM for gradient (OGM-G, Kim & Fessler 2021).

Creating new algorithms via minimax approach

Situation seems quite involved in general, apart from a few cases

- $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu=0$: optimized gradient method (OGM, Kim & Fessler 2016),
- $\frac{\|x_N-x_\star\|^2}{\|x_0-x_\star\|^2}$: information-theoretic exact method (ITEM, T & Drori 2021),
- $\frac{\|\nabla f(x_N)\|^2}{f(x_0)-f_\star}$ with $\mu=0$: OGM for gradient (OGM-G, Kim & Fessler 2021).

Relation to quadratics? When specifying f to be quadratic, similar known methods

Creating new algorithms via minimax approach

Situation seems quite involved in general, apart from a few cases

- $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu = 0$: optimized gradient method (OGM, Kim & Fessler 2016),
- $\frac{\|x_N-x_\star\|^2}{\|x_0-x_\star\|^2}$: information-theoretic exact method (ITEM, T & Drori 2021),
- $\frac{\|\nabla f(x_N)\|^2}{f(x_0)-f_\star}$ with $\mu = 0$: OGM for gradient (OGM-G, Kim & Fessler 2021).

Relation to quadratics? When specifying f to be quadratic, similar known methods

- $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu = 0$ (via Chebyshev polynomials),

Creating new algorithms via minimax approach

Situation seems quite involved in general, apart from a few cases

- $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu=0$: optimized gradient method (OGM, Kim & Fessler 2016),
- $\frac{\|x_N-x_\star\|^2}{\|x_0-x_\star\|^2}$: information-theoretic exact method (ITEM, T & Drori 2021),
- $\frac{\|\nabla f(x_N)\|^2}{f(x_0)-f_\star}$ with $\mu=0$: OGM for gradient (OGM-G, Kim & Fessler 2021).

Relation to quadratics? When specifying f to be quadratic, similar known methods

- $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu=0$ (via Chebyshev polynomials),
- $\frac{\|x_N-x_\star\|^2}{\|x_0-x_\star\|^2}$ (via Chebyshev polynomials), asymptotically Polyak's Heavy-Ball

Creating new algorithms via minimax approach

Situation seems quite involved in general, apart from a few cases

- $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu=0$: optimized gradient method (OGM, Kim & Fessler 2016),
- $\frac{\|x_N-x_\star\|^2}{\|x_0-x_\star\|^2}$: information-theoretic exact method (ITEM, T & Drori 2021),
- $\frac{\|\nabla f(x_N)\|^2}{f(x_0)-f_\star}$ with $\mu=0$: OGM for gradient (OGM-G, Kim & Fessler 2021).

Relation to quadratics? When specifying f to be quadratic, similar known methods

- $\frac{f(x_N)-f_\star}{\|x_0-x_\star\|^2}$ with $\mu=0$ (via Chebyshev polynomials),
- $\frac{\|x_N-x_\star\|^2}{\|x_0-x_\star\|^2}$ (via Chebyshev polynomials), asymptotically Polyak's Heavy-Ball
- see e.g.: A. Nemirovsky's "Information-based complexity of convex programming." (lecture notes, 1995)

Creating new algorithms via minimax approach

Other examples of methods constructed using the minimax approach:

- ◇ Kim (2021). "Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions".

Creating new algorithms via minimax approach

Other examples of methods constructed using the minimax approach:

- ◇ Kim (2021). "Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions".
- ◇ Park, Ryu (2022). "Exact optimal accelerated complexity for fixed-point iterations".

Creating new algorithms via minimax approach

Other examples of methods constructed using the minimax approach:

- Kim (2021). "Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions".
- Park, Ryu (2022). "Exact optimal accelerated complexity for fixed-point iterations".

New methodology:

- Das Gupta, Van Parijs, Ryu (2022). "Branch-and-Bound Performance Estimation Programming: A Unified Methodology for Constructing Optimal Optimization Methods".

Subspace search elimination

For choosing step sizes $\{h_{i,j}\}$, study greedy method:

Subspace search elimination

For choosing step sizes $\{h_{i,j}\}$, study greedy method:

Greedy First-order Method (GFOM)

Inputs: f, x_0.

For $i = 1, 2, \ldots$

$$x_i \in \operatorname*{argmin}_{x \in \mathbb{R}^d} \{f(x) : \ x \in x_0 + \mathrm{span}\{\nabla f(x_0), \ldots, \nabla f(x_{i-1})\}\} .$$

Subspace search elimination

For choosing step sizes $\{h_{i,j}\}$, study greedy method:

Gradient method with exact line search

Inputs: f, x_0.

For $i = 1, 2, \ldots$

$$x_i \in \operatorname*{argmin}_{x \in \mathbb{R}^d} \{f(x) : \ x \in x_{i-1} + \mathrm{span}\{\nabla f(x_{i-1})\}\}.$$

Subspace search elimination

For choosing step sizes $\{h_{i,j}\}$, study greedy method:

Gradient method with exact line search

Inputs: f, x_0.

For $i = 1, 2, \ldots$

$$x_i \in \operatorname*{argmin}_{x \in \mathbb{R}^d} \{ f(x) : \ x \in x_{i-1} + \operatorname{span}\{\nabla f(x_{i-1})\}\} .$$

Running example: solve

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

with $f \in \mathcal{F}_{\mu,L}$ (L-smooth μ-strongly convex).

Subspace search elimination

For choosing step sizes $\{h_{i,j}\}$, study greedy method:

Gradient method with exact line search

Inputs: f, x_0.

For $i = 1, 2, \ldots$

$$x_i \in \operatorname*{argmin}_{x\in\mathbb{R}^d} \{f(x) : \ x \in x_{i-1} + \operatorname{span}\{\nabla f(x_{i-1})\}\}.$$

Running example: solve

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

with $f \in \mathcal{F}_{\mu,L}$ (L-smooth μ-strongly convex).

Exact line-search or optimal fixed step size?

The convergence rate can be written as

Exact line-search or optimal fixed step size?

The convergence rate can be written as

$$\rho \overset{\text{(def)}}{=} \max_{x_0, x_1, f \in \mathcal{F}_{\mu,L}} \left\{ \frac{f_1 - f_\star}{f_0 - f_\star} \text{ st } \langle \nabla f(x_1), \nabla f(x_0) \rangle = 0,\ \langle \nabla f(x_1), x_1 - x_0 \rangle = 0 \right\},$$

Exact line-search or optimal fixed step size?

The convergence rate can be written as

$$\rho \overset{\text{(def)}}{=} \max_{x_0, x_1, f \in \mathcal{F}_{\mu,L}} \left\{ \frac{f_1 - f_\star}{f_0 - f_\star} \text{ st } \langle \nabla f(x_1), \nabla f(x_0) \rangle = 0, \; \langle \nabla f(x_1), x_1 - x_0 \rangle = 0 \right\},$$

it can be upper bounded using a Lagrangian relaxation with $\lambda_1, \lambda_2 \in \mathbb{R}$:

$$\rho_\leqslant \bar{\rho}(\lambda_1, \lambda_2) \overset{\text{(def)}}{=} \max_{x_0, x_1, f \in \mathcal{F}_{\mu,L}} \left\{ \frac{f_1 - f_\star}{f_0 - f_\star} + \lambda_1 \langle \nabla f(x_1), \nabla f(x_0) \rangle + \lambda_2 \langle \nabla f(x_1), x_1 - x_0 \rangle \right\}.$$

Exact line-search or optimal fixed step size?

The convergence rate can be written as

$$\rho \overset{(\text{def})}{=} \max_{x_0, x_1, f \in \mathcal{F}_{\mu,L}} \left\{ \frac{f_1 - f_\star}{f_0 - f_\star} \text{ st } \langle \nabla f(x_1), \nabla f(x_0) \rangle = 0,\ \langle \nabla f(x_1), x_1 - x_0 \rangle = 0 \right\},$$

it can be upper bounded using a Lagrangian relaxation with $\lambda_1, \lambda_2 \in \mathbb{R}$:

$$\rho \leqslant \bar{\rho}(\lambda_1, \lambda_2) \overset{(\text{def})}{=} \max_{x_0, x_1, f \in \mathcal{F}_{\mu,L}} \left\{ \frac{f_1 - f_\star}{f_0 - f_\star} + \lambda_1 \langle \nabla f(x_1), \nabla f(x_0) \rangle + \lambda_2 \langle \nabla f(x_1), x_1 - x_0 \rangle \right\}.$$

We can also create an intermediary problem

$$\rho \leqslant \qquad \leqslant \bar{\rho}(\lambda_1, \lambda_2).$$

Exact line-search or optimal fixed step size?

The convergence rate can be written as

$$\rho \overset{\text{(def)}}{=} \max_{x_0, x_1, f \in \mathcal{F}_{\mu,L}} \left\{ \frac{f_1 - f_\star}{f_0 - f_\star} \text{ st } \langle \nabla f(x_1), \nabla f(x_0) \rangle = 0, \ \langle \nabla f(x_1), x_1 - x_0 \rangle = 0 \right\},$$

it can be upper bounded using a Lagrangian relaxation with $\lambda_1, \lambda_2 \in \mathbb{R}$:

$$\rho_{\leqslant} \bar{\rho}(\lambda_1, \lambda_2) \overset{\text{(def)}}{=} \max_{x_0, x_1, f \in \mathcal{F}_{\mu,L}} \left\{ \frac{f_1 - f_\star}{f_0 - f_\star} + \lambda_1 \langle \nabla f(x_1), \nabla f(x_0) \rangle + \lambda_2 \langle \nabla f(x_1), x_1 - x_0 \rangle \right\}.$$

We can also create an intermediary problem

$$\rho \leqslant \max_{x_0, x_1, f \in \mathcal{F}_{\mu,L}} \left\{ \frac{f_1 - f_\star}{f_0 - f_\star} \text{ st } \lambda_1 \langle \nabla f(x_1), \nabla f(x_0) \rangle + \lambda_2 \langle \nabla f(x_1), x_1 - x_0 \rangle = 0 \right\} \leqslant \bar{\rho}(\lambda_1, \lambda_2).$$

Exact line-search or optimal fixed step size?

The convergence rate can be written as

$$\rho \overset{\text{(def)}}{=} \max_{x_0, x_1, f \in \mathcal{F}_{\mu,L}} \left\{ \frac{f_1 - f_\star}{f_0 - f_\star} \text{ st } \langle \nabla f(x_1), \nabla f(x_0) \rangle = 0,\ \langle \nabla f(x_1), x_1 - x_0 \rangle = 0 \right\},$$

it can be upper bounded using a Lagrangian relaxation with $\lambda_1, \lambda_2 \in \mathbb{R}$:

$$\rho \leqslant \bar{\rho}(\lambda_1, \lambda_2) \overset{\text{(def)}}{=} \max_{x_0, x_1, f \in \mathcal{F}_{\mu,L}} \left\{ \frac{f_1 - f_\star}{f_0 - f_\star} + \lambda_1 \langle \nabla f(x_1), \nabla f(x_0) \rangle + \lambda_2 \langle \nabla f(x_1), x_1 - x_0 \rangle \right\}.$$

We can also create an intermediary problem

$$\rho \leqslant \max_{x_0, x_1, f \in \mathcal{F}_{\mu,L}} \left\{ \frac{f_1 - f_\star}{f_0 - f_\star} \text{ st } \lambda_1 \langle \nabla f(x_1), \nabla f(x_0) \rangle + \lambda_2 \langle \nabla f(x_1), x_1 - x_0 \rangle = 0 \right\} \leqslant \bar{\rho}(\lambda_1, \lambda_2).$$

So: worst-case rate $\bar{\rho}(\lambda_1, \lambda_2)$ applies to all methods described by:

$$\langle \nabla f(x_1), \lambda_1 \nabla f(x_0) + \lambda_2 (x_1 - x_0) \rangle = 0.$$

Exact line-search or optimal fixed step size?

The convergence rate can be written as

$$\rho \overset{(\text{def})}{=} \max_{x_0, x_1, f \in \mathcal{F}_{\mu,L}} \left\{ \frac{f_1 - f_\star}{f_0 - f_\star} \text{ st } \langle \nabla f(x_1), \nabla f(x_0) \rangle = 0,\ \langle \nabla f(x_1), x_1 - x_0 \rangle = 0 \right\},$$

it can be upper bounded using a Lagrangian relaxation with $\lambda_1, \lambda_2 \in \mathbb{R}$:

$$\rho \leqslant \bar{\rho}(\lambda_1, \lambda_2) \overset{(\text{def})}{=} \max_{x_0, x_1, f \in \mathcal{F}_{\mu,L}} \left\{ \frac{f_1 - f_\star}{f_0 - f_\star} + \lambda_1 \langle \nabla f(x_1), \nabla f(x_0) \rangle + \lambda_2 \langle \nabla f(x_1), x_1 - x_0 \rangle \right\}.$$

We can also create an intermediary problem

$$\rho \leqslant \max_{x_0, x_1, f \in \mathcal{F}_{\mu,L}} \left\{ \frac{f_1 - f_\star}{f_0 - f_\star} \text{ st } \lambda_1 \langle \nabla f(x_1), \nabla f(x_0) \rangle + \lambda_2 \langle \nabla f(x_1), x_1 - x_0 \rangle = 0 \right\} \leqslant \bar{\rho}(\lambda_1, \lambda_2).$$

So: worst-case rate $\bar{\rho}(\lambda_1, \lambda_2)$ applies to all methods described by:

$$\langle \nabla f(x_1), \lambda_1 \nabla f(x_0) + \lambda_2 (x_1 - x_0) \rangle = 0.$$

If there exists $\lambda_1^\star, \lambda_2^\star \neq 0$ such that $\rho = \bar{\rho}(\lambda_1^\star, \lambda_2^\star)$, an optimal step size is given by $\frac{\lambda_1^\star}{\lambda_2^\star}$.

Example: non-smooth convex minimization

Non-smooth convex minimization setting:

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

with f convex and $\|g\| \leqslant M$ for any $g \in \partial f(x)$ for some $x \in \mathbb{R}$.

Example: non-smooth convex minimization

Non-smooth convex minimization setting:

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

with f convex and $\|g\| \leqslant M$ for any $g \in \partial f(x)$ for some $x \in \mathbb{R}$.

Lower bound for large-scale setting ($d \geqslant N + 2$):

$$f(x_N) - f(x_\star) \geqslant \frac{M\|x_0 - x_\star\|^2}{\sqrt{N+1}}.$$

Example: non-smooth convex minimization

- Let $\{x_i\}_{i\geqslant 0}$ be a sequence generated by GFOM from f and x_0, and let x_0 be such that $R = \|x_0 - x_\star\|$ for some $x_\star$; then for all $N \in \mathbb{N}$

$$f(x_N) - f(x_\star) \leqslant \frac{MR}{\sqrt{N+1}}.$$

Example: non-smooth convex minimization

- Let $\{x_i\}_{i\geqslant 0}$ be a sequence generated by GFOM from f and x_0, and let x_0 be such that $R = \|x_0 - x_\star\|$ for some $x_\star$; then for all $N \in \mathbb{N}$

$$f(x_N) - f(x_\star) \leqslant \frac{MR}{\sqrt{N+1}}.$$

- For any sequence $x_1, \ldots, x_N$ that satisfies

$$\left\langle \nabla f(x_i), x_i - \left[\frac{i}{i+1} x_{i-1} + \frac{1}{i+1} x_0 - \frac{1}{i+1} \frac{R}{M\sqrt{N+1}} \sum_{j=0}^{i-1} \nabla f(x_j) \right] \right\rangle = 0$$

for all $i = 1, \ldots, N$, we have

$$f(x_N) - f(x_\star) \leqslant \frac{MR}{\sqrt{N+1}}.$$

Example: non-smooth convex minimization

Three methods with the same (optimal) worst-case behavior

Greedy First-order Method (GFOM)

Inputs: f, x_0, N.

For $i = 1, \ldots, N$

$$x_i = \underset{x \in \mathbb{R}^d}{\operatorname{argmin}} \{f(x) : \; x \in x_0 + \operatorname{span}\{\nabla f(x_0), \ldots, \nabla f(x_{i-1})\}\}.$$

Worst-case guarantee:

$$f(x_N) - f(x_\star) \leqslant \frac{M\|x_0 - x_\star\|^2}{\sqrt{N+1}}.$$

Example: non-smooth convex minimization

Three methods with the same (optimal) worst-case behavior

Optimized subgradient method with exact line-search

Inputs: f, x_0, N.

For $i = 1, \ldots, N$

$$y_i = \frac{i}{i+1} x_{i-1} + \frac{1}{i+1} x_0$$

$$d_i = \sum_{j=0}^{i-1} \nabla f(x_j)$$

$$\alpha = \operatorname*{argmin}_{\alpha \in \mathbb{R}} f(y_i + \alpha d_i)$$

$$x_i = y_i + \alpha d_i$$

Worst-case guarantee:

$$f(x_N) - f(x_\star) \leqslant \frac{M \|x_0 - x_\star\|^2}{\sqrt{N+1}}.$$

Example: non-smooth convex minimization

Three methods with the same (optimal) worst-case behavior

Optimized subgradient method

Inputs: f, x_0, N.

For $i = 1, \ldots, N$

$$y_i = x_0 - \frac{1}{\sqrt{N+1}} \frac{R}{M} \sum_{j=0}^{i-1} \nabla f(x_j)$$

$$x_i = \frac{i}{i+1} x_{i-1} + \frac{1}{i+1} y_i$$

Worst-case guarantee:

$$f(x_N) - f(x_\star) \leqslant \frac{M\|x_0 - x_\star\|^2}{\sqrt{N+1}}.$$

Example: smooth convex minimization

Smooth convex minimization setting:

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

with f being L-smooth and convex.

Example: smooth convex minimization

Smooth convex minimization setting:

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

with f being L-smooth and convex.

Lower bound for large-scale setting ($d \geqslant N+2$) by Drori (2017):

$$f(x_N) - f(x_\star) \geqslant \frac{L\|x_0 - x_\star\|^2}{2\theta_N^2},$$

with $\theta_0 = 1$, and:

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

Example: smooth convex minimization

Three methods with the same (optimal) worst-case behavior

Greedy First-order Method (GFOM)

Inputs: f, x_0, N.

For $i = 1, 2, \ldots$

$$x_i = \operatorname*{argmin}_{x \in \mathbb{R}^d} \left\{ f(x) : \; x \in x_0 + \mathrm{span}\{\nabla f(x_0), \ldots, \nabla f(x_{i-1})\} \right\}.$$

Worst-case guarantee:

$$f(x_N) - f(x_\star) \leqslant \frac{L\|x_0 - x_\star\|^2}{2\theta_N^2}.$$

Example: smooth convex minimization

Three methods with the same (optimal) worst-case behavior

Optimized gradient method with exact line-search

Inputs: f, x_0, N.

For $i = 1, \ldots, N$

$$y_i = \left(1 - \frac{1}{\theta_i}\right) x_{i-1} + \frac{1}{\theta_i} x_0$$

$$d_i = \left(1 - \frac{1}{\theta_i}\right) \nabla f(x_{i-1}) + \frac{1}{\theta_i} \left(2 \sum_{j=0}^{i-1} \theta_j \nabla f(x_j)\right)$$

$$\alpha = \operatorname*{argmin}_{\alpha \in \mathbb{R}} f(y_i + \alpha d_i)$$

$$x_i = y_i + \alpha d_i$$

Worst-case guarantee:

$$f(x_N) - f(x_\star) \leqslant \frac{L\|x_0 - x_\star\|^2}{2\theta_N^2}.$$

Example: smooth convex minimization

Three methods with the same (optimal) worst-case behavior

Optimized gradient method

Inputs: f, x_0, N.

For $i = 1, \ldots, N$

$$y_i = x_{i-1} - \frac{1}{L}\nabla f(x_{i-1})$$

$$z_i = x_0 - \frac{2}{L}\sum_{j=0}^{i-1}\theta_j \nabla f(x_j)$$

$$x_i = \left(1 - \frac{1}{\theta_i}\right) y_i + \frac{1}{\theta_i} z_i$$

Worst-case guarantee:

$$f(x_N) - f(x_\star) \leqslant \frac{L\|x_0 - x_\star\|^2}{2\theta_N^2}.$$

See Drori and Teboulle (2014) and Kim and Fessler (2016).

Creating new algorithms via subspace search elimination

Methods & methodology:

- ◇ de Klerk, Glineur, T (2017). "On the worst-case complexity of the gradient method with exact line search for smooth strongly convex functions".

Creating new algorithms via subspace search elimination

Methods & methodology:

- ◇ de Klerk, Glineur, T (2017). "On the worst-case complexity of the gradient method with exact line search for smooth strongly convex functions".
- ◇ Drori, T (2020). "Efficient first-order methods for convex minimization: a constructive approach".

Reminders

Notions of simplicity

Designing methods

Concluding remarks

Perspectives on PEPs

- Systematic access on complexity analyses,

Perspectives on PEPs

- ⋄ Systematic access on complexity analyses,
- ⋄ obtain natural proofs/wc examples,

Perspectives on PEPs

- ◇ Systematic access on complexity analyses,
- ◇ obtain natural proofs/wc examples,
- ◇ identify minimal assumptions,

Perspectives on PEPs

- ⋄ Systematic access on complexity analyses,
- ⋄ obtain natural proofs/wc examples,
- ⋄ identify minimal assumptions,
- ⋄ use convex relaxations (tightness is comfortable, but not required),

Perspectives on PEPs

- Systematic access on complexity analyses,
- obtain natural proofs/wc examples,
- identify minimal assumptions,
- use convex relaxations (tightness is comfortable, but not required),
- study/develop methods beyond traditional comfort zones, for instance:
  - non-Euclidean setups,
  - adaptive methods,
  - higher-order methods.

A few other instructive examples

Worst-case analysis for fixed-point iterations:

- ◇ Lieder (2020). "On the convergence of the Halpern-iteration".

A few other instructive examples

Worst-case analysis for fixed-point iterations:

- Lieder (2020). "On the convergence of the Halpern-iteration".

Analysis of the proximal-point algorithm for monotone inclusions:

- Gu, Yang (2019). "Optimal nonergodic sublinear convergence rate of the proximal point algorithm for maximal monotone inclusion problems".

A few other instructive examples

Worst-case analysis for fixed-point iterations:

- Lieder (2020). "On the convergence of the Halpern-iteration".

Analysis of the proximal-point algorithm for monotone inclusions:

- Gu, Yang (2019). "Optimal nonergodic sublinear convergence rate of the proximal point algorithm for maximal monotone inclusion problems".

Application to designing first-order methods:

- Van Scoy, Freeman, Lynch (2017). "The fastest known globally convergent first-order method for minimizing strongly convex functions".

A few other instructive examples

Worst-case analysis for fixed-point iterations:

- Lieder (2020). "On the convergence of the Halpern-iteration".

Analysis of the proximal-point algorithm for monotone inclusions:

- Gu, Yang (2019). "Optimal nonergodic sublinear convergence rate of the proximal point algorithm for maximal monotone inclusion problems".

Application to designing first-order methods:

- Van Scoy, Freeman, Lynch (2017). "The fastest known globally convergent first-order method for minimizing strongly convex functions".

Application to nonconvex optimization:

- Abbaszadehpeivasti, de Klerk, Zamani (2021). "The exact worst-case convergence rate of the gradient method with fixed step lengths for L-smooth functions".
- Rotaru, Glineur, Patrinos (2022). "Tight convergence rates of the gradient method on hypoconvex functions".

Application to distributed optimization:

- Colla, Hendrickx (2021). "Automated Worst-Case Performance Analysis of Decentralized Gradient Descent".

Shameless advertisement

Application to Bregman methods:

- ◇ Dragomir, T, d'Aspremont, Bolte (2021). "Optimal complexity and certification of Bregman first-order methods".

Shameless advertisement

Application to Bregman methods:

- ◇ Dragomir, T, d'Aspremont, Bolte (2021). "Optimal complexity and certification of Bregman first-order methods".

Continuous-time PEPs:

- ◇ Moucer, T, Bach (2022). "A systematic approach to Lyapunov analyses of continuous-time models in convex optimization".

Shameless advertisement

Application to Bregman methods:

- ◇ Dragomir, T, d'Aspremont, Bolte (2021). "Optimal complexity and certification of Bregman first-order methods".

Continuous-time PEPs:

- ◇ Moucer, T, Bach (2022). "A systematic approach to Lyapunov analyses of continuous-time models in convex optimization".

Application to finding minimal working assumptions:

- ◇ Goujaud, T, Dieuleveut (2022). 'Optimal first-order methods for convex functions with a quadratic upper bound".

Shameless advertisement

Application to Bregman methods:

- ◇ Dragomir, T, d'Aspremont, Bolte (2021). "Optimal complexity and certification of Bregman first-order methods".

Continuous-time PEPs:

- ◇ Moucer, T, Bach (2022). "A systematic approach to Lyapunov analyses of continuous-time models in convex optimization".

Application to finding minimal working assumptions:

- ◇ Goujaud, T, Dieuleveut (2022). 'Optimal first-order methods for convex functions with a quadratic upper bound".

Application to extragradient-type methods:

- ◇ Gorbunov, T, Gidel. "Last-iterate convergence of optimistic gradient method for monotone variational inequalities".

Shameless advertisement

Application to Bregman methods:

- Dragomir, T, d'Aspremont, Bolte (2021). "Optimal complexity and certification of Bregman first-order methods".

Continuous-time PEPs:

- Moucer, T, Bach (2022). "A systematic approach to Lyapunov analyses of continuous-time models in convex optimization".

Application to finding minimal working assumptions:

- Goujaud, T, Dieuleveut (2022). 'Optimal first-order methods for convex functions with a quadratic upper bound".

Application to extragradient-type methods:

- Gorbunov, T, Gidel. "Last-iterate convergence of optimistic gradient method for monotone variational inequalities".

Application to adaptive first-order methods:

- Barré, T, Aspremont (2020). "Complexity Guarantees for Polyak Steps with Momentum".
- Das Gupta, Freund, Sun, T (2023). "Nonlinear conjugate gradient methods: worst-case convergence rates via computer-assisted analyses".

Main references

- T, Hendrickx, Glineur (2017). "Smooth strongly convex interpolation and exact worst-case performance of first-order methods".
- T, Bach (2019). "Stochastic first-order methods: non-asymptotic and computer-aided analyses via potential functions".
- Drori, T (2020). "Efficient first-order methods for convex minimization: a constructive approach".

Packages:

- T, Hendrickx, Glineur (2017). "Performance estimation toolbox (PESTO): Automated worst-case analysis of first-order optimization methods".
- Goujaud et al (2022). "PEPit: computer-assisted worst-case analyses of first-order optimization methods in Python".

Thanks! Questions?

On Github:

PerformanceEstimation/Performance-Estimation-Toolbox

PerformanceEstimation/PEPit