Computer-aided worst-case analyses for first-order optimization

Adrien Taylor

One world optimization seminar – June 2020

Disclaimers about this presentation

Overall idea: principled approach to worst-case analyses in first-order optimization.

Disclaimers about this presentation

Overall idea: principled approach to worst-case analyses in first-order optimization.

Based on original ideas by Drori and Teboulle (2014).

Disclaimers about this presentation

Overall idea: principled approach to worst-case analyses in first-order optimization.

Based on original ideas by Drori and Teboulle (2014).

My personal (and informal) view on this topic

based on insights obtained through works with great collaborators.

Disclaimers about this presentation

Overall idea: principled approach to worst-case analyses in first-order optimization.

Based on original ideas by Drori and Teboulle (2014).

My personal (and informal) view on this topic

based on insights obtained through works with great collaborators.

Informal and example-based presentation.

Disclaimers about this presentation

Overall idea: principled approach to worst-case analyses in first-order optimization.

Based on original ideas by Drori and Teboulle (2014).

My personal (and informal) view on this topic

based on insights obtained through works with great collaborators.

Informal and example-based presentation.

If interested, details are provided in references at the end.

Disclaimers about this presentation

Overall idea: principled approach to worst-case analyses in first-order optimization.

- Based on original ideas by Drori and Teboulle (2014).
- My personal (and informal) view on this topic
  - based on insights obtained through works with great collaborators.

Informal and example-based presentation.

- If interested, details are provided in references at the end.
- Complementary material on Francis Bach's blog (also $\pm$ informal)
  - `https://francisbach.com/computer-aided-analyses/`

Disclaimers about this presentation

Overall idea: principled approach to worst-case analyses in first-order optimization.

- Based on original ideas by Drori and Teboulle (2014).
- My personal (and informal) view on this topic
  - based on insights obtained through works with great collaborators.

Informal and example-based presentation.

- If interested, details are provided in references at the end.
- Complementary material on Francis Bach's blog (also $\pm$ informal)
  - `https://francisbach.com/computer-aided-analyses/`
- More examples in toolbox' manual
  - `https://github.com/AdrienTaylor/Performance-Estimation-Toolbox`


François
Glineur


Julien
Hendrickx


Etienne
de Klerk


Ernest
Ryu


Yoel
Drori


Francis
Bach


Jérôme
Bolte


Alexandre
d'Aspremont


Mathieu
Barré


Radu-Alexandru
Dragomir


Bryan
Van Scoy


Laurent
Lessard


Carolina
Bergeling


Pontus
Giselsson

Toy example: gradient descent

A few examples

Simplified proofs?

Concluding remarks and perspectives

Toy example: gradient descent

A few examples

Simplified proofs?

Concluding remarks and perspectives

Analysis of a gradient method

Say we aim to solve

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

under some assumptions on f (it belongs to some class of functions).

Analysis of a gradient method

Say we aim to solve

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

under some assumptions on f (it belongs to some class of functions).

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

Analysis of a gradient method

Say we aim to solve

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

under some assumptions on f (it belongs to some class of functions).

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

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

Analysis of a gradient method

Say we aim to solve

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

under some assumptions on f (it belongs to some class of functions).

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

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

Examples: how small should $f(x_N) - f(x_\star)$, $\|f'(x_N)\|$, $\|x_N - x_\star\|$ be?

Worst-case guarantees

Example: what can we a priori guarantee on $\|f'(x_N)\|$

Worst-case guarantees

Example: what can we a priori guarantee on $\|f'(x_N)\|$

- for all f satisfying some assumptions,

Worst-case guarantees

Example: what can we a priori guarantee on $\|f'(x_N)\|$

- ◇ for all f satisfying some assumptions,
- ◇ for x_N was obtained through gradient descent from x_0?

Worst-case guarantees

Example: what can we a priori guarantee on $\|f'(x_N)\|$

- for all f satisfying some assumptions,
- for x_N was obtained through gradient descent from x_0?

By definition, the "best" such guarantee is

$$\|f'(x_N)\| \leqslant \text{“worst possible value of } \|f'(x_N)\|\text{, given the assumptions”}.$$

Worst-case guarantees

Example: what can we a priori guarantee on $\|f'(x_N)\|$

- $\diamond$ for all f satisfying some assumptions,
- $\diamond$ for x_N was obtained through gradient descent from x_0?

By definition, the "best" such guarantee is

$$\left\|f'(x_N)\right\| \leqslant \text{"worst possible value of } \left\|f'(x_N)\right\|\text{, given the assumptions".}$$

In other words:

$$\begin{aligned} \left\|f'(x_N)\right\| \leqslant \max_{F, y_0, \ldots, y_N} \quad & \left\|F'(y_N)\right\| \\ \text{subject to} \quad & y_1, \ldots, y_N \text{ generated by gradient method from } y_0 \\ & F \text{ satisfies the assumptions on } f \end{aligned}$$

Worst-case guarantees

Example: what can we a priori guarantee on $\|f'(x_N)\|$

- $\diamond$ for all f satisfying some assumptions,
- $\diamond$ for x_N was obtained through gradient descent from x_0?

By definition, the "best" such guarantee is

$$\left\|f'(x_N)\right\| \leqslant \text{"worst possible value of } \left\|f'(x_N)\right\|\text{, given the assumptions".}$$

In other words:

$$\begin{aligned} \left\|f'(x_N)\right\| \leqslant \max_{F, y_0, \ldots, y_N} \quad & \left\|F'(y_N)\right\| \\ \text{subject to} \quad & y_1, \ldots, y_N \text{ generated by gradient method from } y_0 \\ & F \text{ satisfies the assumptions on } f \end{aligned}$$

This problem is typically unbounded (arbitrarily bad starting point are feasible).

Worst-case guarantees

Example: what can we a priori guarantee on $\|f'(x_N)\|$

- $\diamond$ for all f satisfying some assumptions,
- $\diamond$ for x_N was obtained through gradient descent from x_0?

By definition, the "best" such guarantee is

$$\left\|f'(x_N)\right\| \leqslant \text{“worst possible value of } \left\|f'(x_N)\right\|\text{, given the assumptions”.}$$

In other words:

$$\begin{aligned} \left\|f'(x_N)\right\| \leqslant \max_{F, y_0, \ldots, y_N} \quad & \left\|F'(y_N)\right\| \\ \text{subject to} \quad & y_1, \ldots, y_N \text{ generated by gradient method from } y_0 \\ & F \text{ satisfies the assumptions on } f \end{aligned}$$

This problem is typically unbounded (arbitrarily bad starting point are feasible).

Standard workaround: assume something on the starting point,
for example: assume bounded $\|x_0 - x_\star\|^2$, $\|f'(x_0)\|^2$ or $f(x_0) - f(x_\star)$.

Worst-case guarantees

Example: what can we a priori guarantee on $\|f'(x_N)\|$

- for all f **and all x_0** satisfying some assumptions,
- for x_N was obtained through gradient descent from x_0?

By definition, the "best" such guarantee is

$$\left\|f'(x_N)\right\| \leqslant \text{"worst possible value of } \left\|f'(x_N)\right\|\text{, given the assumptions".}$$

In other words:

$$\begin{aligned}\left\|f'(x_N)\right\| \leqslant \max_{F, y_0, \ldots, y_N} \quad & \left\|F'(y_N)\right\| \\ \text{subject to} \quad & y_1, \ldots, y_N \text{ generated by gradient method from } y_0 \\ & F \text{ satisfies the assumptions on } f \\ & \boldsymbol{y_0 \textbf{ not too bad.}}\end{aligned}$$

This problem is typically unbounded (arbitrarily bad starting point are feasible).

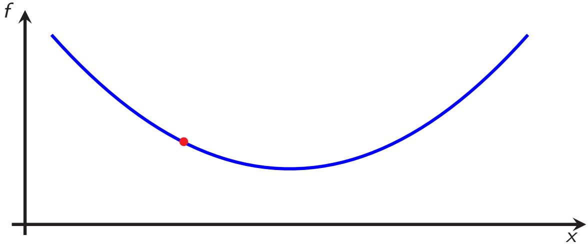
Standard workaround: assume something on the starting point,
for example: assume bounded $\|x_0 - x_\star\|^2$, $\|f'(x_0)\|^2$ or $f(x_0) - f(x_\star)$.

Smooth strongly convex functions

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:

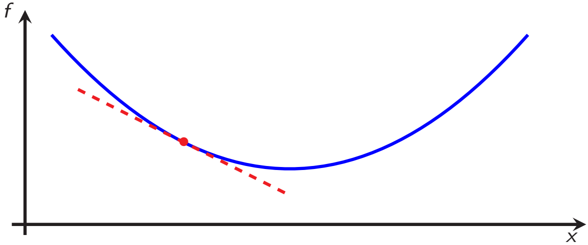
Smooth strongly convex functions

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:



Smooth strongly convex functions

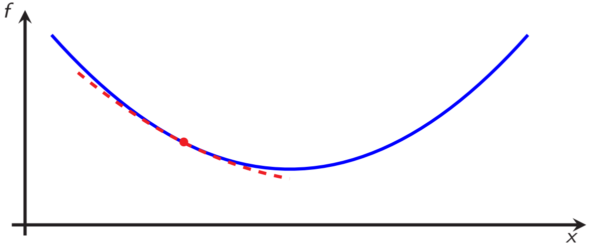
Consider a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, f is (μ-strongly) convex and L-smooth iff $\forall x, y \in \mathbb{R}^d$ we have:



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

Smooth strongly convex functions

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:

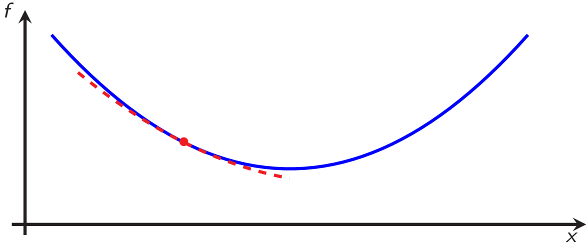


(1) (Convexity) $f(x) \geqslant f(y) + \langle f'(y), x - y \rangle$,

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

Smooth strongly convex functions

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:



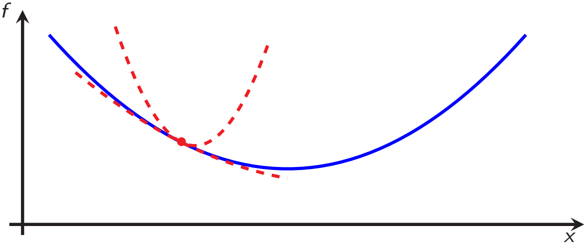
(1) (Convexity) $f(x) \geqslant f(y) + \langle f'(y), x - y \rangle$,

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

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

Smooth strongly convex functions

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) $f(x) \geqslant f(y) + \langle f'(y), x-y \rangle$,

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

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

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

Convergence rate of a gradient step

Convergence rate of a gradient step

Toy example: What can we guarantee on $\|f'(x_1)\|$ given that:

- f is L-smooth and μ-strongly convex (notation $f \in \mathcal{F}_{\mu,\mathsf{L}}$),
- x_1 was generated by gradient descent: $x_1 = x_0 - \gamma f'(x_0)$,
- $\|f'(x_0)\|$ is bounded?

Convergence rate of a gradient step

Toy example: What can we guarantee on $\|f'(x_1)\|$ given that:

- f is L-smooth and μ-strongly convex (notation $f \in \mathcal{F}_{\mu,\mathbf{L}}$),
- x_1 was generated by gradient descent: $x_1 = x_0 - \gamma f'(x_0)$,
- $\|f'(x_0)\|$ is bounded?

$$\max_{f, x_0, x_1} \; \left\|f'(x_1)\right\|^2$$

$$\text{s.t. } f \in \mathcal{F}_{\mu,L} \qquad\qquad \text{Functional class}$$

Convergence rate of a gradient step

Toy example: What can we guarantee on $\|f'(x_1)\|$ given that:

- f is L-smooth and μ-strongly convex (notation $f \in \mathcal{F}_{\mu,\mathbf{L}}$),
- x_1 was generated by gradient descent: $x_1 = x_0 - \gamma f'(x_0)$,
- $\|f'(x_0)\|$ is bounded?

$$\begin{aligned}
\max_{f, x_0, x_1} \quad & \left\|f'(x_1)\right\|^2 & \\
\text{s.t.} \quad & f \in \mathcal{F}_{\mu,L} & \text{Functional class} \\
& x_1 = x_0 - \gamma f'(x_0) & \text{Algorithm}
\end{aligned}$$

Convergence rate of a gradient step

Toy example: What can we guarantee on $\|f'(x_1)\|$ given that:

- f is L-smooth and μ-strongly convex (notation $f \in \mathcal{F}_{\mu,\mathbf{L}}$),
- x_1 was generated by gradient descent: $x_1 = x_0 - \gamma f'(x_0)$,
- $\|f'(x_0)\|$ is bounded?

$$\begin{aligned}
\max_{f, x_0, x_1} \quad & \left\|f'(x_1)\right\|^2 & \\
\text{s.t.} \quad & f \in \mathcal{F}_{\mu,L} & \text{Functional class} \\
& x_1 = x_0 - \gamma f'(x_0) & \text{Algorithm} \\
& \left\|f'(x_0)\right\|^2 = R^2 & \text{Initial condition}
\end{aligned}$$

Convergence rate of a gradient step

Toy example: What can we guarantee on $\|f'(x_1)\|$ given that:

- ◇ f is L-smooth and μ-strongly convex (notation $f \in \mathcal{F}_{\mu,\mathbf{L}}$),
- ◇ x_1 was generated by gradient descent: $x_1 = x_0 - \gamma f'(x_0)$,
- ◇ $\|f'(x_0)\|$ is bounded?

$$
\begin{aligned}
\max_{f, x_0, x_1} \quad & \left\|f'(x_1)\right\|^2 & \\
\text{s.t.} \quad & f \in \mathcal{F}_{\mu,L} & \text{Functional class} \\
& x_1 = x_0 - \gamma f'(x_0) & \text{Algorithm} \\
& \left\|f'(x_0)\right\|^2 = R^2 & \text{Initial condition}
\end{aligned}
$$

<u>Variables</u>: f, x_0, x_1;

Convergence rate of a gradient step

Toy example: What can we guarantee on $\|f'(x_1)\|$ given that:

- ◇ f is L-smooth and μ-strongly convex (notation $f \in \mathcal{F}_{\mu,\mathbf{L}}$),
- ◇ x_1 was generated by gradient descent: $x_1 = x_0 - \gamma f'(x_0)$,
- ◇ $\|f'(x_0)\|$ is bounded?

$$\begin{aligned}
\max_{f, x_0, x_1} \quad & \left\|f'(x_1)\right\|^2 & \\
\text{s.t.} \quad & f \in \mathcal{F}_{\mu,L} & \text{Functional class} \\
& x_1 = x_0 - \gamma f'(x_0) & \text{Algorithm} \\
& \left\|f'(x_0)\right\|^2 = R^2 & \text{Initial condition}
\end{aligned}$$

Variables: f, x_0, x_1; parameters: μ, L, γ, R.

From infinite to finite dimensional problems

As it is, the previous problem does not seem very practical...

From infinite to finite dimensional problems

As it is, the previous problem does not seem very practical...

- How to treat the infinite dimensional variable f?

From infinite to finite dimensional problems

As it is, the previous problem does not seem very practical...

- How to treat the infinite dimensional variable f?
- How to cope with the constraint $f \in \mathcal{F}_{\mu,L}$?

From infinite to finite dimensional problems

As it is, the previous problem does not seem very practical...

- How to treat the infinite dimensional variable f?

- How to cope with the constraint $f \in \mathcal{F}_{\mu,L}$?

Idea:

From infinite to finite dimensional problems

As it is, the previous problem does not seem very practical...

- How to treat the infinite dimensional variable f?
- How to cope with the constraint $f \in \mathcal{F}_{\mu,L}$?

Idea:

- replace f by its discrete version:

$$f_i = f(x_i),\ g_i = f'(x_i) \quad \forall i \in \{0,1\}.$$

From infinite to finite dimensional problems

As it is, the previous problem does not seem very practical...

- How to treat the infinite dimensional variable f?
- How to cope with the constraint $f \in \mathcal{F}_{\mu,L}$?

Idea:

- replace f by its discrete version:

$$f_i = f(x_i), \;\; g_i = f'(x_i) \quad \forall i \in \{0, 1\}\,.$$

- Require points (x_i, g_i, f_i) to be interpolable by a function $f \in \mathcal{F}_{\mu,L}$.

From infinite to finite dimensional problems

As it is, the previous problem does not seem very practical...

- How to treat the infinite dimensional variable f?
- How to cope with the constraint $f \in \mathcal{F}_{\mu,L}$?

Idea:

- replace f by its discrete version:

$$f_i = f(x_i),\ g_i = f'(x_i) \quad \forall i \in \{0, 1\}\,.$$

- Require points (x_i, g_i, f_i) to be interpolable by a function $f \in \mathcal{F}_{\mu,L}$.
  The new constraint is:

$$\exists f \in \mathcal{F}_{\mu,L}:\ f_i = f(x_i),\ g_i = f'(x_i), \qquad \forall i \in \{0, 1\}\,.$$

Sampled version

Sampled version

- Performance estimation problem:

$$\begin{aligned} \max_{f, x_0, x_1} \quad & \|f'(x_1)\|^2 \\ \text{subject to} \quad & f \text{ is } L\text{-smooth and } \mu\text{-strongly convex}, \\ & x_1 = x_0 - \gamma f'(x_0), \\ & \|f'(x_0)\|^2 = R^2. \end{aligned}$$

Sampled version

- Performance estimation problem:

$$\begin{aligned} \max_{f, x_0, x_1} \quad & \|f'(x_1)\|^2 \\ \text{subject to} \quad & f \text{ is } L\text{-smooth and } \mu\text{-strongly convex}, \\ & x_1 = x_0 - \gamma f'(x_0), \\ & \|f'(x_0)\|^2 = R^2. \end{aligned}$$

- Variables: f, x_0, x_1.

Sampled version

- Performance estimation problem:

$$\begin{aligned} \max_{f, x_0, x_1} \quad & \|f'(x_1)\|^2 \\ \text{subject to} \quad & f \text{ is } L\text{-smooth and } \mu\text{-strongly convex}, \\ & x_1 = x_0 - \gamma f'(x_0), \\ & \|f'(x_0)\|^2 = R^2. \end{aligned}$$

- Variables: f, x_0, x_1.
- Sampled version:

$$\begin{aligned} \max_{\substack{x_0, x_1, g_0, g_1 \\ f_0, f_1}} \quad & \|g_1\|^2 \\ \text{subject to} \quad & \exists f \in \mathcal{F}_{\mu,L} \text{ such that } \begin{cases} f_i = f(x_i) & i = 0,1 \\ g_i = f'(x_i) & i = 0,1 \end{cases} \\ & x_1 = x_0 - \gamma g_0, \\ & \|g_0\|^2 = R^2. \end{aligned}$$

Sampled version

- Performance estimation problem:

$$\begin{aligned} \max_{f, x_0, x_1} \quad & \|f'(x_1)\|^2 \\ \text{subject to} \quad & f \text{ is } L\text{-smooth and } \mu\text{-strongly convex}, \\ & x_1 = x_0 - \gamma f'(x_0), \\ & \|f'(x_0)\|^2 = R^2. \end{aligned}$$

- Variables: f, x_0, x_1.
- Sampled version:

$$\begin{aligned} \max_{\substack{x_0, x_1, g_0, g_1 \\ f_0, f_1}} \quad & \|g_1\|^2 \\ \text{subject to} \quad & \exists f \in \mathcal{F}_{\mu,L} \text{ such that } \begin{cases} f_i = f(x_i) & i = 0,1 \\ g_i = f'(x_i) & i = 0,1 \end{cases} \\ & x_1 = x_0 - \gamma g_0, \\ & \|g_0\|^2 = R^2. \end{aligned}$$

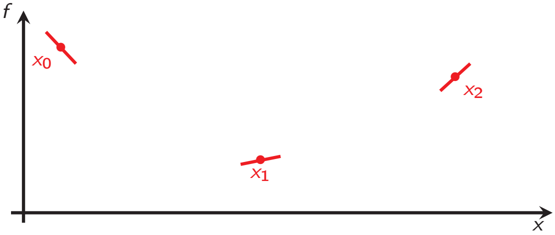
- Variables: x_0, x_1, g_0, g_1, f_0, f_1.

Smooth strongly convex interpolation

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

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.

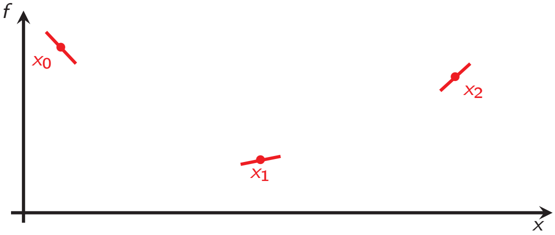


? Possible to find $f \in \mathcal{F}_{\mu,L}$ such that

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

Smooth strongly convex interpolation

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

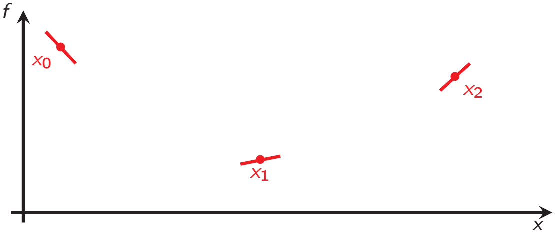
$$f(x_i) = f_i, \quad \text{and} \quad g_i \in \partial f(x_i), \qquad \forall i \in S.$$

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

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

Smooth strongly convex interpolation

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: $\forall i, j \in S$

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

- 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

$$
\begin{aligned}
\max_{\substack{x_0, x_1, g_0, g_1 \\ f_0, f_1}} \quad & \|g_1\|^2 \\
\text{subject to} \quad & \exists f \in \mathcal{F}_{\mu,L} \text{ such that } \begin{cases} f_i = f(x_i) & i = 1,2 \\ g_i = f'(x_i) & i = 1,2 \end{cases} \\
& x_1 = x_0 - \gamma g_0, \\
& \|g_0\|^2 = R^2.
\end{aligned}
$$

Replace constraints

- Interpolation conditions allow removing red constraints

$$\begin{aligned}
\max_{\substack{x_0, x_1, g_0, g_1\\ f_0, f_1}} \quad & \|g_1\|^2 \\
\text{subject to} \quad & \exists f \in \mathcal{F}_{\mu,L} \text{ such that } \begin{cases} f_i = f(x_i) & i = 1,2 \\ g_i = f'(x_i) & i = 1,2 \end{cases} \\
& x_1 = x_0 - \gamma g_0, \\
& \|g_0\|^2 = R^2.
\end{aligned}$$

- replacing them by

$$f_1 \geqslant f_0 + \langle g_0, x_1 - x_0 \rangle + \tfrac{1}{2L}\|g_1 - g_0\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_1 - x_0 - \tfrac{1}{L}(g_1 - g_0)\right\|^2$$

$$f_0 \geqslant f_1 + \langle g_1, x_0 - x_1 \rangle + \tfrac{1}{2L}\|g_0 - g_1\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(g_0 - g_1)\right\|^2.$$

Replace constraints

◇ Interpolation conditions allow removing red constraints

$$
\begin{aligned}
\max_{\substack{x_0, x_1, g_0, g_1 \\ f_0, f_1}} \quad & \|g_1\|^2 \\
\text{subject to} \quad & \exists f \in \mathcal{F}_{\mu,L} \text{ such that } \begin{cases} f_i = f(x_i) & i = 1,2 \\ g_i = f'(x_i) & i = 1,2 \end{cases} \\
& x_1 = x_0 - \gamma g_0, \\
& \|g_0\|^2 = R^2.
\end{aligned}
$$

◇ replacing them by

$$
\begin{aligned}
f_1 &\geqslant f_0 + \langle g_0, x_1 - x_0 \rangle + \tfrac{1}{2L}\|g_1 - g_0\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_1 - x_0 - \tfrac{1}{L}(g_1 - g_0)\right\|^2 \\
f_0 &\geqslant f_1 + \langle g_1, x_0 - x_1 \rangle + \tfrac{1}{2L}\|g_0 - g_1\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(g_0 - g_1)\right\|^2.
\end{aligned}
$$

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

Semidefinite lifting

Semidefinite lifting

◇ Using $x_1 = x_0 - \gamma g_0$, all elements are quadratic in (g_0, g_1), and linear in (f_0, f_1):

$$\begin{aligned}
\max_{\substack{g_0, g_1 \\ f_0, f_1}} \quad & \|g_1\|^2 \\
\text{subject to} \quad & f_1 \geqslant f_0 - \gamma\|g_0\|^2 + \tfrac{1}{2L}\|g_1 - g_0\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|\gamma g_0 + \tfrac{1}{L}(g_1 - g_0)\right\|^2 \\
& f_0 \geqslant f_1 + \gamma\langle g_1, g_0\rangle + \tfrac{1}{2L}\|g_1 - g_0\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|\gamma g_0 + \tfrac{1}{L}(g_1 - g_0)\right\|^2 \\
& \|g_0\|^2 = R^2.
\end{aligned}$$

Semidefinite lifting

- Using $x_1 = x_0 - \gamma g_0$, all elements are quadratic in (g_0, g_1), and linear in (f_0, f_1):

$$\begin{aligned}
\max_{\substack{g_0, g_1 \\ f_0, f_1}} \quad & \|g_1\|^2 \\
\text{subject to} \quad & f_1 \geqslant f_0 - \gamma\|g_0\|^2 + \tfrac{1}{2L}\|g_1 - g_0\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|\gamma g_0 + \tfrac{1}{L}(g_1 - g_0)\right\|^2 \\
& f_0 \geqslant f_1 + \gamma\langle g_1, g_0\rangle + \tfrac{1}{2L}\|g_1 - g_0\|^2 + \tfrac{\mu}{2(1-\mu/L)}\left\|\gamma g_0 + \tfrac{1}{L}(g_1 - g_0)\right\|^2 \\
& \|g_0\|^2 = R^2.
\end{aligned}$$

- They are therefore linear in terms of a Gram matrix G and a vector F, with

$$G = \begin{bmatrix} \|g_0\|^2 & \langle g_0, g_1\rangle \\ \langle g_0, g_1\rangle & \|g_1\|^2 \end{bmatrix} = \begin{bmatrix} g_0 & g_1 \end{bmatrix}^\top \begin{bmatrix} g_0 & g_1 \end{bmatrix}, \quad F = \begin{bmatrix} f_0 & f_1 \end{bmatrix},$$

where $G \succcurlyeq 0$ by construction.

Semidefinite lifting

Semidefinite lifting

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

$$G = \begin{bmatrix} \|g_0\|^2 & \langle g_0, g_1 \rangle \\ \langle g_0, g_1 \rangle & \|g_1\|^2 \end{bmatrix}, \quad F = \begin{bmatrix} f_0 & f_1 \end{bmatrix},$$

Semidefinite lifting

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

$$G = \begin{bmatrix} \|g_0\|^2 & \langle g_0, g_1\rangle \\ \langle g_0, g_1\rangle & \|g_1\|^2 \end{bmatrix}, \quad F = \begin{bmatrix} f_0 & f_1 \end{bmatrix},$$

- previous problem can be reformulated as a 2x2 SDP

$$\begin{aligned} \max_{G,\,F} \quad & G_{2,2} \\ \text{subject to} \quad & F_1 - F_0 + \tfrac{\gamma L(2-\gamma\mu)-1}{2(L-\mu)} G_{1,1} + \tfrac{1-\gamma\mu}{L-\mu} G_{1,2} - \tfrac{1}{2(L-\mu)} G_{2,2} \geqslant 0 \\ & F_0 - F_1 + \tfrac{\gamma\mu(2-\gamma L)-1}{2(L-\mu)} G_{1,1} + \tfrac{1-\gamma L}{L-\mu} G_{1,2} - \tfrac{1}{2(L-\mu)} G_{2,2} \geqslant 0 \\ & G_{1,1} = 1 \\ & G \succcurlyeq 0. \end{aligned}$$

Semidefinite lifting

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

$$G = \begin{bmatrix} \|g_0\|^2 & \langle g_0, g_1\rangle \\ \langle g_0, g_1\rangle & \|g_1\|^2 \end{bmatrix}, \quad F = \begin{bmatrix} f_0 & f_1 \end{bmatrix},$$

◇ previous problem can be reformulated as a 2x2 SDP

$$\begin{aligned} \max_{G,\,F} \quad & G_{2,2} \\ \text{subject to} \quad & F_1 - F_0 + \tfrac{\gamma L(2-\gamma\mu)-1}{2(L-\mu)} G_{1,1} + \tfrac{1-\gamma\mu}{L-\mu} G_{1,2} - \tfrac{1}{2(L-\mu)} G_{2,2} \geqslant 0 \\ & F_0 - F_1 + \tfrac{\gamma\mu(2-\gamma L)-1}{2(L-\mu)} G_{1,1} + \tfrac{1-\gamma L}{L-\mu} G_{1,2} - \tfrac{1}{2(L-\mu)} G_{2,2} \geqslant 0 \\ & G_{1,1} = 1 \\ & G \succcurlyeq 0. \end{aligned}$$

◇ Assuming $g_0, g_1 \in \mathbb{R}^d$ with $d \geqslant 2$, same optimal value as original problem!

Semidefinite lifting

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

$$G = \begin{bmatrix} \|g_0\|^2 & \langle g_0, g_1 \rangle \\ \langle g_0, g_1 \rangle & \|g_1\|^2 \end{bmatrix}, \quad F = \begin{bmatrix} f_0 & f_1 \end{bmatrix},$$

- previous problem can be reformulated as a 2x2 SDP

$$\begin{aligned} \max_{G,\,F} \quad & G_{2,2} \\ \text{subject to} \quad & F_1 - F_0 + \tfrac{\gamma L(2-\gamma\mu)-1}{2(L-\mu)} G_{1,1} + \tfrac{1-\gamma\mu}{L-\mu} G_{1,2} - \tfrac{1}{2(L-\mu)} G_{2,2} \geqslant 0 \\ & F_0 - F_1 + \tfrac{\gamma\mu(2-\gamma L)-1}{2(L-\mu)} G_{1,1} + \tfrac{1-\gamma L}{L-\mu} G_{1,2} - \tfrac{1}{2(L-\mu)} G_{2,2} \geqslant 0 \\ & G_{1,1} = 1 \\ & G \succcurlyeq 0. \end{aligned}$$

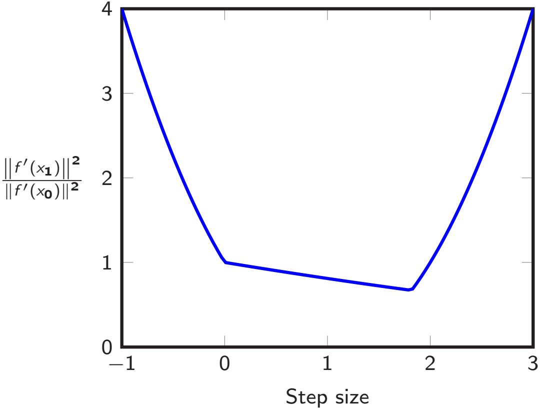
- Assuming $g_0, g_1 \in \mathbb{R}^d$ with $d \geqslant 2$, same optimal value as original problem!
- For $d = 1$ same optimal value by adding $\operatorname{rank}(G) \leqslant 1$.

Solving the SDP...

Fix $L = 1$, $\mu = .1$ and solve the SDP for a few values of γ.

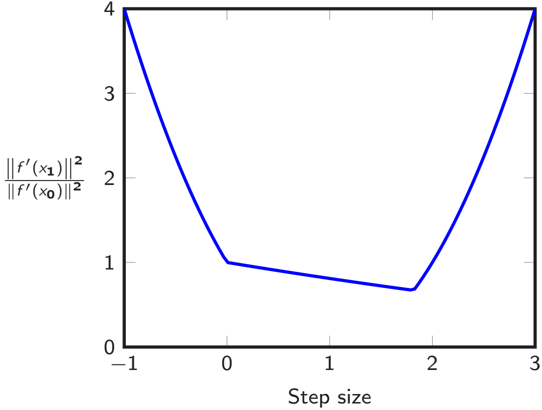
Solving the SDP...

Fix $L = 1$, $\mu = .1$ and solve the SDP for a few values of γ.



Solving the SDP...

Fix $L = 1$, $\mu = .1$ and solve the SDP for a few values of γ.



Observation: numerics match the (expected) $\max\{(1-\gamma L)^2, (1-\gamma\mu)^2\}$.

Translation to worst-case guarantees

- Let us rephrase our target: we look for $\rho(\gamma)$ (hopefully small) such that

$$\|f'(x_1)\| \leq \rho(\gamma)\|f'(x_0)\|$$

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

Translation to worst-case guarantees

- Let us rephrase our target: we look for $\rho(\gamma)$ (hopefully small) such that

$$\|f'(x_1)\| \leq \rho(\gamma)\|f'(x_0)\|$$

  is satisfied for all $x_0 \in \mathbb{R}^d$, $f \in \mathcal{F}_{\mu,L}$, and $x_1 = x_0 - \gamma f'(x_0)$.
- Feasible points to the previous SDP correspond to lower bounds on $\rho(\gamma)$.

Translation to worst-case guarantees

- Let us rephrase our target: we look for $\rho(\gamma)$ (hopefully small) such that

$$\|f'(x_1)\| \leq \rho(\gamma)\|f'(x_0)\|$$

  is satisfied for all $x_0 \in \mathbb{R}^d$, $f \in \mathcal{F}_{\mu,L}$, and $x_1 = x_0 - \gamma f'(x_0)$.
- Feasible points to the previous SDP correspond to lower bounds on $\rho(\gamma)$.
- Traditionally: guarantees on $\rho(\gamma)$ obtained by combining inequalities (due to problem assumptions).

  Exactly what a dual does!

Translation to worst-case guarantees

- Let us rephrase our target: we look for $\rho(\gamma)$ (hopefully small) such that

$$\|f'(x_1)\| \leq \rho(\gamma)\|f'(x_0)\|$$

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

- Feasible points to the previous SDP correspond to lower bounds on $\rho(\gamma)$.
- Traditionally: guarantees on $\rho(\gamma)$ obtained by combining inequalities (due to problem assumptions).

  Exactly what a dual does!

- Any $\rho(\gamma)$ that is valid for all d is a feasible point to the dual SDP.

Dual problem

- Introduce dual variables τ, λ_1 and λ_2,

Dual problem

- Introduce dual variables τ, λ_1 and λ_2,
- dual problem becomes

$$
\begin{aligned}
\underset{\tau,\lambda_1,\lambda_2\geqslant 0}{\text{minimize}}\ & \tau \\
\text{subject to } & S = \begin{bmatrix} -\frac{\lambda_1(\gamma\mu-1)(\gamma L-1)}{L-\mu}-\tau & -\frac{\lambda_1(\gamma(\mu+L)-2)}{2(L-\mu)} \\ -\frac{\lambda_1(\gamma(\mu+L)-2)}{2(L-\mu)} & 1-\frac{\lambda_1}{L-\mu} \end{bmatrix} \preccurlyeq 0 \\
& 0 = \lambda_1 - \lambda_2.
\end{aligned}
$$

Dual problem

- Introduce dual variables τ, λ_1 and λ_2,
- dual problem becomes

$$\begin{aligned} \underset{\tau,\lambda_1,\lambda_2\geqslant 0}{\text{minimize}}\ & \tau \\ \text{subject to } & S = \begin{bmatrix} -\frac{\lambda_1(\gamma\mu-1)(\gamma L-1)}{L-\mu} - \tau & -\frac{\lambda_1(\gamma(\mu+L)-2)}{2(L-\mu)} \\ -\frac{\lambda_1(\gamma(\mu+L)-2)}{2(L-\mu)} & 1 - \frac{\lambda_1}{L-\mu} \end{bmatrix} \preccurlyeq 0 \\ & 0 = \lambda_1 - \lambda_2. \end{aligned}$$

- From any feasible point we get a valid rate $\rho^2(\gamma) = \tau(\gamma)$.

Dual problem

- Introduce dual variables τ, λ_1 and λ_2,
- dual problem becomes

$$
\begin{aligned}
\operatorname*{minimize}_{\tau,\lambda_1,\lambda_2 \geqslant 0} \ & \tau \\
\text{subject to } & S = \begin{bmatrix} -\frac{\lambda_1(\gamma\mu-1)(\gamma L-1)}{L-\mu} - \tau & -\frac{\lambda_1(\gamma(\mu+L)-2)}{2(L-\mu)} \\ -\frac{\lambda_1(\gamma(\mu+L)-2)}{2(L-\mu)} & 1-\frac{\lambda_1}{L-\mu} \end{bmatrix} \preccurlyeq 0 \\
& 0 = \lambda_1 - \lambda_2.
\end{aligned}
$$

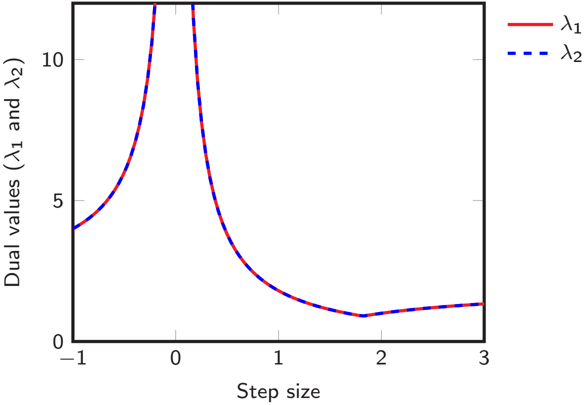
- From any feasible point we get a valid rate $\rho^2(\gamma) = \tau(\gamma)$.
- Strong duality holds (existence of a Slater point).

Solving the dual

Fix $L = 1$, $\mu = .1$ and solve the dual SDP for a few values of γ.

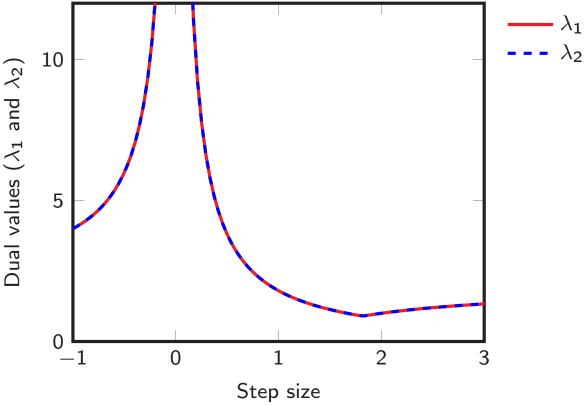
Solving the dual

Fix $L = 1$, $\mu = .1$ and solve the dual SDP for a few values of γ.



Solving the dual

Fix $L = 1$, $\mu = .1$ and solve the dual SDP for a few values of γ.



Note: numerics match $\lambda_1 = \lambda_2 = \frac{2}{|\gamma|}\rho(\gamma)$ with $\rho(\gamma) = \max\{|1 - \gamma L|, |1 - \gamma\mu|\}$.

Recovering a “standard” proof

Gradient with $\gamma = \frac{1}{L}$. Perform weighted sum of two inequalities

Recovering a "standard" proof

Gradient with $\gamma = \frac{1}{L}$. Perform weighted sum of two inequalities

$$\begin{aligned}
f_0 \geqslant f_1 \quad & +\langle f'(x_1), x_0 - x_1 \rangle + \tfrac{1}{2L}\|f'(x_0) - f'(x_1)\|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1))\right\|^2 && : \lambda_1 \\
f_1 \geqslant f_0 \quad & +\langle f'(x_0), x_1 - x_0 \rangle + \tfrac{1}{2L}\|f'(x_0) - f'(x_1)\|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1))\right\|^2 && : \lambda_2
\end{aligned}$$

Recovering a "standard" proof

Gradient with $\gamma = \frac{1}{L}$. Perform weighted sum of two inequalities

$$
\begin{aligned}
f_0 \geqslant f_1 \quad & +\langle f'(x_1), x_0 - x_1\rangle + \tfrac{1}{2L}\|f'(x_0) - f'(x_1)\|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1))\right\|^2 \qquad : \lambda_1 \\
f_1 \geqslant f_0 \quad & +\langle f'(x_0), x_1 - x_0\rangle + \tfrac{1}{2L}\|f'(x_0) - f'(x_1)\|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1))\right\|^2 \qquad : \lambda_2
\end{aligned}
$$

with $\lambda_1, \lambda_2 \geqslant 0$. Weighted sum can be reformulated as

Recovering a "standard" proof

Gradient with $\gamma = \frac{1}{L}$. Perform weighted sum of two inequalities

$$\begin{aligned}
f_0 \geqslant f_1 \quad & +\langle f'(x_1), x_0 - x_1\rangle + \tfrac{1}{2L}\|f'(x_0) - f'(x_1)\|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1))\right\|^2 \qquad : \lambda_1 = \tfrac{2}{\gamma}(1-\mu\gamma) \\
f_1 \geqslant f_0 \quad & +\langle f'(x_0), x_1 - x_0\rangle + \tfrac{1}{2L}\|f'(x_0) - f'(x_1)\|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1))\right\|^2 \qquad : \lambda_2 = \tfrac{2}{\gamma}(1-\mu\gamma)
\end{aligned}$$

with $\lambda_1, \lambda_2 \geqslant 0$. Weighted sum can be reformulated as

Recovering a "standard" proof

Gradient with $\gamma = \frac{1}{L}$. Perform weighted sum of two inequalities

$$\begin{aligned} f_0 \geqslant f_1 \quad & +\langle f'(x_1), x_0 - x_1\rangle + \tfrac{1}{2L}\|f'(x_0) - f'(x_1)\|^2 \\ & + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1))\right\|^2 && : \lambda_1 = \tfrac{2}{\gamma}(1-\mu\gamma) \\ f_1 \geqslant f_0 \quad & +\langle f'(x_0), x_1 - x_0\rangle + \tfrac{1}{2L}\|f'(x_0) - f'(x_1)\|^2 \\ & + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1))\right\|^2 && : \lambda_2 = \tfrac{2}{\gamma}(1-\mu\gamma) \end{aligned}$$

with $\lambda_1, \lambda_2 \geqslant 0$. Weighted sum can be reformulated as

$$\left\|f'(x_1)\right\|^2 \leqslant (1-\gamma\mu)^2 \left\|f'(x_0)\right\|^2 - \underbrace{\tfrac{2-\gamma(L+\mu)}{\gamma(L-\mu)}\left\|(1-\mu\gamma)f'(x_0) - f'(x_1)\right\|^2}_{},$$

Recovering a "standard" proof

Gradient with $\gamma = \frac{1}{L}$. Perform weighted sum of two inequalities

$$\begin{aligned}
f_0 \geqslant f_1 \quad & +\langle f'(x_1), x_0 - x_1\rangle + \tfrac{1}{2L}\|f'(x_0) - f'(x_1)\|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1))\right\|^2 \qquad : \lambda_1 = \tfrac{2}{\gamma}(1-\mu\gamma) \\
f_1 \geqslant f_0 \quad & +\langle f'(x_0), x_1 - x_0\rangle + \tfrac{1}{2L}\|f'(x_0) - f'(x_1)\|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1))\right\|^2 \qquad : \lambda_2 = \tfrac{2}{\gamma}(1-\mu\gamma)
\end{aligned}$$

with $\lambda_1, \lambda_2 \geqslant 0$. Weighted sum can be reformulated as

$$\left\|f'(x_1)\right\|^2 \leqslant (1-\gamma\mu)^2 \left\|f'(x_0)\right\|^2 - \underbrace{\tfrac{2-\gamma(L+\mu)}{\gamma(L-\mu)}\left\|(1-\mu\gamma)f'(x_0) - f'(x_1)\right\|^2}_{\geqslant 0},$$

Recovering a "standard" proof

Gradient with $\gamma = \frac{1}{L}$. Perform weighted sum of two inequalities

$$
\begin{aligned}
f_0 \geqslant f_1 \quad & +\langle f'(x_1), x_0 - x_1\rangle + \tfrac{1}{2L}\|f'(x_0) - f'(x_1)\|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1))\right\|^2 && : \lambda_1 = \tfrac{2}{\gamma}(1-\mu\gamma) \\
f_1 \geqslant f_0 \quad & +\langle f'(x_0), x_1 - x_0\rangle + \tfrac{1}{2L}\|f'(x_0) - f'(x_1)\|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1))\right\|^2 && : \lambda_2 = \tfrac{2}{\gamma}(1-\mu\gamma)
\end{aligned}
$$

with $\lambda_1, \lambda_2 \geqslant 0$. Weighted sum can be reformulated as

$$
\begin{aligned}
\left\|f'(x_1)\right\|^2 & \leqslant (1-\gamma\mu)^2\left\|f'(x_0)\right\|^2 - \underbrace{\tfrac{2-\gamma(L+\mu)}{\gamma(L-\mu)}\left\|(1-\mu\gamma)f'(x_0) - f'(x_1)\right\|^2}_{\geqslant 0}, \\
& \leqslant (1-\gamma\mu)^2\left\|f'(x_0)\right\|^2,
\end{aligned}
$$

Recovering a "standard" proof

Gradient with $\gamma = \frac{1}{L}$. Perform weighted sum of two inequalities

$$
\begin{aligned}
f_0 \geqslant f_1 \quad & +\langle f'(x_1), x_0 - x_1\rangle + \tfrac{1}{2L}\|f'(x_0) - f'(x_1)\|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1))\right\|^2 \qquad : \lambda_1 = \tfrac{2}{\gamma}(1-\mu\gamma) \\
f_1 \geqslant f_0 \quad & +\langle f'(x_0), x_1 - x_0\rangle + \tfrac{1}{2L}\|f'(x_0) - f'(x_1)\|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1))\right\|^2 \qquad : \lambda_2 = \tfrac{2}{\gamma}(1-\mu\gamma)
\end{aligned}
$$

with $\lambda_1, \lambda_2 \geqslant 0$. Weighted sum can be reformulated as

$$
\begin{aligned}
\left\|f'(x_1)\right\|^2 & \leqslant (1-\gamma\mu)^2 \left\|f'(x_0)\right\|^2 - \underbrace{\tfrac{2-\gamma(L+\mu)}{\gamma(L-\mu)}\left\|(1-\mu\gamma)f'(x_0) - f'(x_1)\right\|^2}_{\geqslant 0}, \\
& \leqslant (1-\gamma\mu)^2 \left\|f'(x_0)\right\|^2,
\end{aligned}
$$

leading to $\|f'(x_1)\|^2 \leqslant (1-\frac{\mu}{L})^2\|f'(x_0)\|^2$

Recovering a "standard" proof

Gradient with $\gamma = \frac{1}{L}$. Perform weighted sum of two inequalities

$$
\begin{aligned}
f_0 \geqslant f_1 \quad & + \langle f'(x_1), x_0 - x_1 \rangle + \tfrac{1}{2L} \| f'(x_0) - f'(x_1) \|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)} \left\| x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1)) \right\|^2 \qquad : \lambda_1 = \tfrac{2}{\gamma}(1 - \mu\gamma) \\
f_1 \geqslant f_0 \quad & + \langle f'(x_0), x_1 - x_0 \rangle + \tfrac{1}{2L} \| f'(x_0) - f'(x_1) \|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)} \left\| x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1)) \right\|^2 \qquad : \lambda_2 = \tfrac{2}{\gamma}(1 - \mu\gamma)
\end{aligned}
$$

with $\lambda_1, \lambda_2 \geqslant 0$. Weighted sum can be reformulated as

$$
\begin{aligned}
\left\| f'(x_1) \right\|^2 & \leqslant (1 - \gamma\mu)^2 \left\| f'(x_0) \right\|^2 - \underbrace{\tfrac{2-\gamma(L+\mu)}{\gamma(L-\mu)} \left\| (1 - \mu\gamma) f'(x_0) - f'(x_1) \right\|^2}_{\geqslant 0, \text{ or } = 0 \text{ when worst-case is achieved}}, \\
& \leqslant (1 - \gamma\mu)^2 \left\| f'(x_0) \right\|^2,
\end{aligned}
$$

leading to $\|f'(x_1)\|^2 \leqslant (1 - \frac{\mu}{L})^2 \|f'(x_0)\|^2$

Recovering a "standard" proof

Gradient with $\gamma = \frac{1}{L}$. Perform weighted sum of two inequalities

$$
\begin{aligned}
f_0 \geqslant f_1 \quad & +\langle f'(x_1), x_0 - x_1\rangle + \tfrac{1}{2L}\|f'(x_0) - f'(x_1)\|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1))\right\|^2 \qquad : \lambda_1 = \tfrac{2}{\gamma}(1-\mu\gamma) \\
f_1 \geqslant f_0 \quad & +\langle f'(x_0), x_1 - x_0\rangle + \tfrac{1}{2L}\|f'(x_0) - f'(x_1)\|^2 \\
& + \tfrac{\mu}{2(1-\mu/L)}\left\|x_0 - x_1 - \tfrac{1}{L}(f'(x_0) - f'(x_1))\right\|^2 \qquad : \lambda_2 = \tfrac{2}{\gamma}(1-\mu\gamma)
\end{aligned}
$$

with $\lambda_1, \lambda_2 \geqslant 0$. Weighted sum can be reformulated as

$$
\begin{aligned}
\left\|f'(x_1)\right\|^2 & \leqslant (1-\gamma\mu)^2 \left\|f'(x_0)\right\|^2 - \underbrace{\tfrac{2-\gamma(L+\mu)}{\gamma(L-\mu)}\left\|(1-\mu\gamma)f'(x_0) - f'(x_1)\right\|^2}_{\geqslant 0, \text{ or } = 0 \text{ when worst-case is achieved}}, \\
& \leqslant (1-\gamma\mu)^2 \left\|f'(x_0)\right\|^2,
\end{aligned}
$$

leading to $\|f'(x_1)\|^2 \leqslant (1-\frac{\mu}{L})^2\|f'(x_0)\|^2$ (tight).

Remarks

Dual interpretations:

Remarks

Dual interpretations:

- Find smallest convergence rate that can be proved by a linear combination of interpolation inequalities.

Remarks

Dual interpretations:

- Find smallest convergence rate that can be proved by a linear combination of interpolation inequalities.
- From strong duality: in such settings, any (dimension-independent) convergence rate can be proved by linear combination of interpolation inequalities.

Remarks

Dual interpretations:

- ◇ Find smallest convergence rate that can be proved by a linear combination of interpolation inequalities.
- ◇ From strong duality: in such settings, any (dimension-independent) convergence rate can be proved by linear combination of interpolation inequalities.
- ◇ Any dual feasible point can be translated into a "traditional" (SDP-less) proof.

Remarks

Dual interpretations:

- Find smallest convergence rate that can be proved by a linear combination of interpolation inequalities.
- From strong duality: in such settings, any (dimension-independent) convergence rate can be proved by linear combination of interpolation inequalities.
- Any dual feasible point can be translated into a "traditional" (SDP-less) proof.

For finding proofs:

Remarks

Dual interpretations:

- ◇ Find smallest convergence rate that can be proved by a linear combination of interpolation inequalities.
- ◇ From strong duality: in such settings, any (dimension-independent) convergence rate can be proved by linear combination of interpolation inequalities.
- ◇ Any dual feasible point can be translated into a "traditional" (SDP-less) proof.

For finding proofs:

- ◇ the SDP might help by playing with both sides:
  - play with primal (e.g., worst-case functions might be easy to identify),
  - play with dual (e.g., dual variables might be easy to identify).

Remarks

Dual interpretations:

- ◇ Find smallest convergence rate that can be proved by a linear combination of interpolation inequalities.
- ◇ From strong duality: in such settings, any (dimension-independent) convergence rate can be proved by linear combination of interpolation inequalities.
- ◇ Any dual feasible point can be translated into a "traditional" (SDP-less) proof.

For finding proofs:

- ◇ the SDP might help by playing with both sides:
  - play with primal (e.g., worst-case functions might be easy to identify),
  - play with dual (e.g., dual variables might be easy to identify).
- ◇ Standard tricks apply, e.g., trace norm minimization for promoting low-rank solutions (on primal or dual).

When does it work?

Problem setting:

- ◇ pick a method
- ◇ pick a class of functions
- ◇ pick a type of inequality we want to reach

  (e.g., via a convergence measure & an initial condition).

When does it work?

Problem setting:

- ◇ pick a method
- ◇ pick a class of functions
- ◇ pick a type of inequality we want to reach
  (e.g., via a convergence measure & an initial condition).

Why could we solve the previous PEP?

When does it work?

Problem setting:

- ◇ pick a method
- ◇ pick a class of functions
- ◇ pick a type of inequality we want to reach
  (e.g., via a convergence measure & an initial condition).

Why could we solve the previous PEP?

- ◇ Step size γ was "fixed beforehand"; no dependence on $f(.)$ (non-adaptive).

When does it work?

Problem setting:

- pick a method
- pick a class of functions
- pick a type of inequality we want to reach
  (e.g., via a convergence measure & an initial condition).

Why could we solve the previous PEP?

- Step size γ was "fixed beforehand"; no dependence on $f(.)$ (non-adaptive).
- Class of function $\mathcal{F}_{\mu,\mathsf{L}}$ was encoded via linear constraints in G and F.

When does it work?

Problem setting:

- pick a method
- pick a class of functions
- pick a type of inequality we want to reach
  (e.g., via a convergence measure & an initial condition).

Why could we solve the previous PEP?

- Step size γ was "fixed beforehand"; no dependence on $f(.)$ (non-adaptive).
- Class of function $\mathcal{F}_{\mu,\mathsf{L}}$ was encoded via linear constraints in G and F.
- Convergence measure $\|f'(x_1)\|^2$ was linear in terms of G and F.

When does it work?

Problem setting:

- pick a method
- pick a class of functions
- pick a type of inequality we want to reach
  (e.g., via a convergence measure & an initial condition).

Why could we solve the previous PEP?

- Step size γ was "fixed beforehand"; no dependence on $f(.)$ (non-adaptive).
- Class of function $\mathcal{F}_{\mu,\mathrm{L}}$ was encoded via linear constraints in G and F.
- Convergence measure $\|f'(x_1)\|^2$ was linear in terms of G and F.
- Initial condition $\|f'(x_0)\|^2$ was linear in terms of G and F.

When does it work?

Problem setting:

- ⋄ pick a method
- ⋄ pick a class of functions
- ⋄ pick a type of inequality we want to reach

  (e.g., via a convergence measure & an initial condition).

Why could we solve the previous PEP?

- ⋄ Step size γ was "fixed beforehand"; no dependence on $f(.)$ (non-adaptive).
- ⋄ Class of function $\mathcal{F}_{\mu,L}$ was encoded via linear constraints in G and F.
- ⋄ Convergence measure $\|f'(x_1)\|^2$ was linear in terms of G and F.
- ⋄ Initial condition $\|f'(x_0)\|^2$ was linear in terms of G and F.

... such conditions (or slight generalizations) apply in a variety of settings.

When does it work?

Problem setting:

- ◇ pick a method
- ◇ pick a class of functions
- ◇ pick a type of inequality we want to reach
  (e.g., via a convergence measure & an initial condition).

Why could we solve the previous PEP?

- ◇ Step size γ was "fixed beforehand"; no dependence on $f(.)$ (non-adaptive).
- ◇ Class of function $\mathcal{F}_{\mu,\mathrm{L}}$ was encoded via linear constraints in G and F.
- ◇ Convergence measure $\|f'(x_1)\|^2$ was linear in terms of G and F.
- ◇ Initial condition $\|f'(x_0)\|^2$ was linear in terms of G and F.

... such conditions (or slight generalizations) apply in a variety of settings.

In other situations, one might want to relax the PEP for obtaining upper-bounds.

PEP genealogy ("my humble, biased, view on...")

Base methodological developments:

PEP genealogy ("my humble, biased, view on...")

Base methodological developments:

'14 Drori and Teboulle (MP): upper bounds on worst-case behaviors of FO methods via SDPs, idea of using this machinery for designing methods.

PEP genealogy ("my humble, biased, view on...")

Base methodological developments:

'14 Drori and Teboulle (MP): upper bounds on worst-case behaviors of FO methods via SDPs, idea of using this machinery for designing methods.

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

PEP genealogy ("my humble, biased, view on...")

Base methodological developments:

'14 Drori and Teboulle (MP): upper bounds on worst-case behaviors of FO methods via SDPs, idea of using this machinery for designing methods.

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

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

In this presentation:

PEP genealogy ("my humble, biased, view on...")

Base methodological developments:

'14 Drori and Teboulle (MP): upper bounds on worst-case behaviors of FO methods via SDPs, idea of using this machinery for designing methods.

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

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

In this presentation:

'17 T, Hendrickx and Glineur: interpolation (tightness), and primal/dual interpretations of the SDPs, and few generalizations of the approach.

PEP genealogy ("my humble, biased, view on...")

Base methodological developments:

- '14 Drori and Teboulle (MP): upper bounds on worst-case behaviors of FO methods via SDPs, idea of using this machinery for designing methods.
- '16 Kim and Fessler (MP): design of an optimized method for smooth convex minimization, using SDPs.
- '16 Lessard, Recht, Packard (SIOPT): smaller SDPs for linear convergence, via integral quadratic constraints ("IQCs"). Essentially Lyapunov functions.

In this presentation:

- '17 T, Hendrickx and Glineur: interpolation (tightness), and primal/dual interpretations of the SDPs, and few generalizations of the approach.
- — Other examples randomly picked from different works.

PEP genealogy (“my humble, biased, view on...”)

Base methodological developments:

- ’14 Drori and Teboulle (MP): upper bounds on worst-case behaviors of FO methods via SDPs, idea of using this machinery for designing methods.
- ’16 Kim and Fessler (MP): design of an optimized method for smooth convex minimization, using SDPs.
- ’16 Lessard, Recht, Packard (SIOPT): smaller SDPs for linear convergence, via integral quadratic constraints (“IQCs”). Essentially Lyapunov functions.

In this presentation:

- ’17 T, Hendrickx and Glineur: interpolation (tightness), and primal/dual interpretations of the SDPs, and few generalizations of the approach.
- — Other examples randomly picked from different works.
- ’19 T, Bach: PEPs for designing potential functions (impose structure in proofs).

But also:

PEP genealogy ("my humble, biased, view on...")

Base methodological developments:

'14 Drori and Teboulle (MP): upper bounds on worst-case behaviors of FO methods via SDPs, idea of using this machinery for designing methods.

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

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

In this presentation:

'17 T, Hendrickx and Glineur: interpolation (tightness), and primal/dual interpretations of the SDPs, and few generalizations of the approach.

— Other examples randomly picked from different works.

'19 T, Bach: PEPs for designing potential functions (impose structure in proofs).

But also:

◇ Fair amount of algorithmic analyses (and design) originated from SDPs (from different authors, examples below), in different settings.

PEP genealogy ("my humble, biased, view on...")

Base methodological developments:

'14 Drori and Teboulle (MP): upper bounds on worst-case behaviors of FO methods via SDPs, idea of using this machinery for designing methods.

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

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

In this presentation:

'17 T, Hendrickx and Glineur: interpolation (tightness), and primal/dual interpretations of the SDPs, and few generalizations of the approach.

— Other examples randomly picked from different works.

'19 T, Bach: PEPs for designing potential functions (impose structure in proofs).

But also:

- ⋄ Fair amount of algorithmic analyses (and design) originated from SDPs (from different authors, examples below), in different settings.
- ⋄ We try keeping track of related works in the toolbox' manual (see later).

Going further

Going further

- Sublinear rates? Via different types of guarantees, for example:

$$f(x_N) - f(x_\star) \leqslant C_N \|x_0 - x_\star\|^2,$$

for some C_N (hopefully small and decreasing with N). Similar ideas and larger SDPs (typically of order N×N).

Going further

- ◇ Sublinear rates? Via different types of guarantees, for example:

$$f(x_N) - f(x_\star) \leqslant C_N \|x_0 - x_\star\|^2,$$

for some C_N (hopefully small and decreasing with N). Similar ideas and larger SDPs (typically of order N×N).

- ◇ Optimizing/designing methods? For example, consider a gradient-type method

$$x_k = x_0 - \sum_{i=0}^{k-1} \gamma_{k,i} f'(x_i),$$

and try to solve minimax ("minimize (over $\{\gamma_{k,i}\}$) the worst-case").

Going further

◇ Sublinear rates? Via different types of guarantees, for example:

$$f(x_N) - f(x_\star) \leqslant C_N \|x_0 - x_\star\|^2,$$

for some C_N (hopefully small and decreasing with N). Similar ideas and larger SDPs (typically of order N×N).

◇ Optimizing/designing methods? For example, consider a gradient-type method

$$x_k = x_0 - \sum_{i=0}^{k-1} \gamma_{k,i} f'(x_i),$$

and try to solve minimax ("minimize (over $\{\gamma_{k,i}\}$) the worst-case"). For example, see: Drori and Teboulle (2014, 2016), Kim and Fessler (2016, 2018, 2019).

Going further

- Sublinear rates? Via different types of guarantees, for example:

$$f(x_N) - f(x_\star) \leqslant C_N \|x_0 - x_\star\|^2,$$

for some C_N (hopefully small and decreasing with N). Similar ideas and larger SDPs (typically of order NxN).

- Optimizing/designing methods? For example, consider a gradient-type method

$$x_k = x_0 - \sum_{i=0}^{k-1} \gamma_{k,i} f'(x_i),$$

and try to solve minimax ("minimize (over $\{\gamma_{k,i}\}$) the worst-case"). For example, see: Drori and Teboulle (2014, 2016), Kim and Fessler (2016, 2018, 2019).

- Lyapunov functions? E.g., let $V_k = a\|x_k - x_\star\|^2 + b\|f'(x_k)\|^2 + c(f(x_k) - f_\star)$.

Going further

- Sublinear rates? Via different types of guarantees, for example:

$$f(x_N) - f(x_\star) \leqslant C_N \|x_0 - x_\star\|^2,$$

for some C_N (hopefully small and decreasing with N). Similar ideas and larger SDPs (typically of order N×N).

- Optimizing/designing methods? For example, consider a gradient-type method

$$x_k = x_0 - \sum_{i=0}^{k-1} \gamma_{k,i} f'(x_i),$$

and try to solve minimax ("minimize (over $\{\gamma_{k,i}\}$) the worst-case"). For example, see: Drori and Teboulle (2014, 2016), Kim and Fessler (2016, 2018, 2019).

- Lyapunov functions? E.g., let $V_k = a\|x_k - x_\star\|^2 + b\|f'(x_k)\|^2 + c(f(x_k) - f_\star)$. Given ρ, feasibility problem

$$\text{"}?\exists a, b, c \;\text{ s.t. }\; V_{k+1} \leqslant \rho V_k\text{"}$$

is convex.

Toy example: gradient descent

A few examples

Simplified proofs?

Concluding remarks and perspectives

François Glineur

Etienne de Klerk

"On the worst-case complexity of the gradient method with exact line search for smooth strongly convex functions"

Steepest descent with inexact search directions

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

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

Steepest descent with inexact search directions

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

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

Relative error model:

$$\|f'(x_i) - d_i\| \leqslant \varepsilon \|f'(x_i)\| \quad i = 0, 1, \ldots, \tag{1}$$

Steepest descent with inexact search directions

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

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

Relative error model:

$$\|f'(x_i) - d_i\| \leqslant \varepsilon \|f'(x_i)\| \quad i = 0, 1, \dots, \tag{1}$$

Noisy gradient descent method with exact line search

Input: $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, $x_0 \in \mathbb{R}^d$, $0 \leq \varepsilon < 1$.

for $i = 0, 1, \dots$

Select any seach direction d_i that satisfies (1);

$\gamma = \operatorname{argmin}_{\gamma \in \mathbb{R}} f\left(x_i - \gamma d_i\right)$

$x_{i+1} = x_i - \gamma d_i$

Steepest descent with inexact search directions

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

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

Relative error model:

$$\|f'(x_i) - d_i\| \leqslant \varepsilon \|f'(x_i)\| \quad i = 0, 1, \ldots, \tag{1}$$

Noisy gradient descent method with exact line search

Input: $f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)$, $x_0 \in \mathbb{R}^d$, $0 \leq \varepsilon < 1$.

for $i = 0, 1, \ldots$

Select any seach direction d_i that satisfies (1);

$\gamma = \text{argmin}_{\gamma\in\mathbb{R}} f(x_i - \gamma d_i)$

$x_{i+1} = x_i - \gamma d_i$

Worst-case behavior:

$$f(x_{i+1}) - f_* \leqslant \left(\frac{1-\kappa_\varepsilon}{1+\kappa_\varepsilon}\right)^2 (f(x_i) - f_*) \quad i = 0, 1, \ldots$$

where $\kappa_\varepsilon = \frac{\mu}{L}\frac{(1-\varepsilon)}{(1+\varepsilon)}$.

Problem formulation

In the same spirit as in previous slides:

$$\begin{aligned}
\max_{f,x_0,x_1,d_0} \quad & f(x_1) - f(x_\star) \\
\text{s.t.} \quad & f \in \mathcal{F}_{\mu,L} \\
& \left\langle f'(x_1), x_1 - x_0 \right\rangle = 0 \\
& \left\langle f'(x_1), d_0 \right\rangle = 0 \\
& \left\| f'(x_0) - d_0 \right\|^2 \leqslant \varepsilon^2 \left\| f'(x_0) \right\|^2 \\
& f(x_0) - f(x_\star) = 1
\end{aligned}$$

SDP with based on $x_0, x_1, x_\star, g_0, g_1, d_0$, and $g_\star = 0$.

Six interpolation conditions (each pair in set of 3 points) for replacing $f \in \mathcal{F}_{\mu,\mathbf{L}}$.

What does a proof look like?

Aggregate constraints:

What does a proof look like?

Aggregate constraints:

$$
\begin{aligned}
f_0 &\geqslant f_1 + \langle g_1, x_0 - x_1 \rangle + \frac{1}{2L}\|g_0 - g_1\|^2 + \frac{\mu}{2\left(1 - \frac{\mu}{L}\right)}\|x_0 - x_1 - (g_0 - g_1)/L\|^2 \\
f_\star &\geqslant f_0 + \langle g_0, x_\star - x_0 \rangle + \frac{1}{2L}\|g_\star - g_0\|^2 + \frac{\mu}{2\left(1 - \frac{\mu}{L}\right)}\|x_\star - x_0 - (g_\star - g_0)/L\|^2 \\
f_\star &\geqslant f_1 + \langle g_1, x_\star - x_1 \rangle + \frac{1}{2L}\|g_\star - g_1\|^2 + \frac{\mu}{2\left(1 - \frac{\mu}{L}\right)}\|x_\star - x_1 - (g_\star - g_1)/L\|^2 \\
0 &= \langle g_1, d_0 \rangle \\
0 &= \langle g_1, x_1 - x_0 \rangle \\
\varepsilon^2\|g_0\|^2 &\geqslant \|g_0 - d_0\|^2
\end{aligned}
$$

What does a proof look like?

Aggregate constraints:

$$
\begin{aligned}
f_0 &\geqslant f_1 + \langle g_1, x_0 - x_1 \rangle + \frac{1}{2L}\|g_0 - g_1\|^2 + \frac{\mu}{2\left(1-\frac{\mu}{L}\right)}\|x_0 - x_1 - (g_0 - g_1)/L\|^2 \\
f_\star &\geqslant f_0 + \langle g_0, x_\star - x_0 \rangle + \frac{1}{2L}\|g_\star - g_0\|^2 + \frac{\mu}{2\left(1-\frac{\mu}{L}\right)}\|x_\star - x_0 - (g_\star - g_0)/L\|^2 \\
f_\star &\geqslant f_1 + \langle g_1, x_\star - x_1 \rangle + \frac{1}{2L}\|g_\star - g_1\|^2 + \frac{\mu}{2\left(1-\frac{\mu}{L}\right)}\|x_\star - x_1 - (g_\star - g_1)/L\|^2 \\
0 &= \langle g_1, d_0 \rangle \\
0 &= \langle g_1, x_1 - x_0 \rangle \\
\varepsilon^2\|g_0\|^2 &\geqslant \|g_0 - d_0\|^2
\end{aligned}
$$

with multipliers

What does a proof look like?

Aggregate constraints:

$$
\begin{aligned}
f_0 &\geqslant f_1 + \langle g_1, x_0 - x_1 \rangle + \frac{1}{2L}\|g_0 - g_1\|^2 + \frac{\mu}{2\left(1-\frac{\mu}{L}\right)}\|x_0 - x_1 - (g_0 - g_1)/L\|^2 \\
f_\star &\geqslant f_0 + \langle g_0, x_\star - x_0 \rangle + \frac{1}{2L}\|g_\star - g_0\|^2 + \frac{\mu}{2\left(1-\frac{\mu}{L}\right)}\|x_\star - x_0 - (g_\star - g_0)/L\|^2 \\
f_\star &\geqslant f_1 + \langle g_1, x_\star - x_1 \rangle + \frac{1}{2L}\|g_\star - g_1\|^2 + \frac{\mu}{2\left(1-\frac{\mu}{L}\right)}\|x_\star - x_1 - (g_\star - g_1)/L\|^2 \\
0 &= \langle g_1, d_0 \rangle \\
0 &= \langle g_1, x_1 - x_0 \rangle \\
\varepsilon^2\|g_0\|^2 &\geqslant \|g_0 - d_0\|^2
\end{aligned}
$$

with multipliers

$$
y_1 = \tfrac{1-\kappa_\varepsilon}{1+\kappa_\varepsilon}, \quad y_2 = \tfrac{2\kappa_\varepsilon(1-\kappa_\varepsilon)}{(1+\kappa_\varepsilon)^2}, \quad y_3 = \tfrac{2\kappa_\varepsilon}{1+\kappa_\varepsilon}, \quad y_4 = \tfrac{2}{L_\varepsilon+\mu_\varepsilon}, \quad y_5 = 1, \quad y_6 = \tfrac{1-\kappa_\varepsilon}{\varepsilon L_\varepsilon(1+\kappa_\varepsilon)^2},
$$

where we used $L_\varepsilon = L(1+\varepsilon)$, $\mu_\varepsilon = \mu(1-\varepsilon)$, and $\kappa_\varepsilon = \mu_\varepsilon/L_\varepsilon$.

What does the proof look like?

Resulting inequality:

$$
\begin{aligned}
f_1 - f_\star \quad \leqslant \quad & \left(\tfrac{1-\kappa_\varepsilon}{1+\kappa_\varepsilon}\right)^2 (f_0 - f_\star) \\
& - \tfrac{L\mu(L_\varepsilon-\mu_\varepsilon)(L_\varepsilon+3\mu_\varepsilon)}{2(L-\mu)(L_\varepsilon+\mu_\varepsilon)^2} \left\| x_0 + \alpha_1 x_1 - (1+\alpha_1)x_\star + \alpha_2 g_0 - \alpha_3 g_1 + \alpha_4 d_0 \right\|^2 \\
& - \tfrac{2L\mu\mu_\varepsilon}{(L-\mu)(L_\varepsilon+3\mu_\varepsilon)} \left\| x_1 - x_\star + \alpha_5 g_0 + \alpha_6 g_1 + \alpha_7 d_0 \right\|^2 \\
& - \tfrac{\varepsilon}{L_\varepsilon+\mu_\varepsilon} \left\| g_1 + \alpha_8 g_0 + \alpha_9 d_0 \right\|^2,
\end{aligned}
$$

for some $\alpha_1, \ldots, \alpha_9$.

What does the proof look like?

Resulting inequality:

$$
\begin{aligned}
f_1 - f_\star \quad \leqslant \quad & \left(\tfrac{1-\kappa_\varepsilon}{1+\kappa_\varepsilon}\right)^2 (f_0 - f_\star) \\
& - \tfrac{L\mu(L_\varepsilon-\mu_\varepsilon)(L_\varepsilon+3\mu_\varepsilon)}{2(L-\mu)(L_\varepsilon+\mu_\varepsilon)^2} \|x_0 + \alpha_1 x_1 - (1+\alpha_1)x_\star + \alpha_2 g_0 - \alpha_3 g_1 + \alpha_4 d_0\|^2 \\
& - \tfrac{2L\mu\mu_\varepsilon}{(L-\mu)(L_\varepsilon+3\mu_\varepsilon)} \|x_1 - x_\star + \alpha_5 g_0 + \alpha_6 g_1 + \alpha_7 d_0\|^2 \\
& - \tfrac{\varepsilon}{L_\varepsilon+\mu_\varepsilon} \|g_1 + \alpha_8 g_0 + \alpha_9 d_0\|^2,
\end{aligned}
$$

for some $\alpha_1, \ldots, \alpha_9$. Last three terms nonpositive, so

$$f_1 - f_\star \leqslant \left(\frac{1-\kappa_\varepsilon}{1+\kappa_\varepsilon}\right)^2 (f_0 - f_\star).$$

What does the proof look like?

Resulting inequality:

$$\begin{aligned} f_1 - f_\star \quad \leqslant \quad & \left(\tfrac{1-\kappa_\varepsilon}{1+\kappa_\varepsilon}\right)^2 (f_0 - f_\star) \\ & -\tfrac{L\mu(L_\varepsilon-\mu_\varepsilon)(L_\varepsilon+3\mu_\varepsilon)}{2(L-\mu)(L_\varepsilon+\mu_\varepsilon)^2}\left\|x_0+\alpha_1 x_1-(1+\alpha_1)x_\star+\alpha_2 g_0-\alpha_3 g_1+\alpha_4 d_0\right\|^2 \\ & -\tfrac{2L\mu\mu_\varepsilon}{(L-\mu)(L_\varepsilon+3\mu_\varepsilon)}\left\|x_1-x_\star+\alpha_5 g_0+\alpha_6 g_1+\alpha_7 d_0\right\|^2 \\ & -\tfrac{\varepsilon}{L_\varepsilon+\mu_\varepsilon}\left\|g_1+\alpha_8 g_0+\alpha_9 d_0\right\|^2, \end{aligned}$$

for some $\alpha_1, \ldots, \alpha_9$. Last three terms nonpositive, so

$$f_1 - f_\star \leqslant \left(\frac{1-\kappa_\varepsilon}{1+\kappa_\varepsilon}\right)^2 (f_0 - f_\star).$$

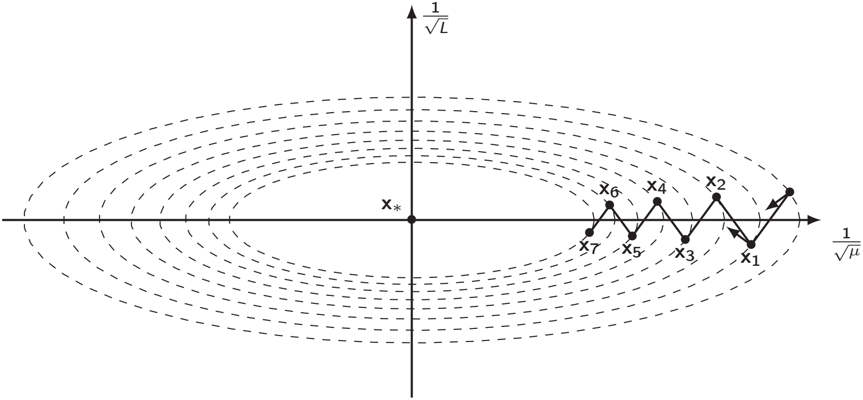
One actually has equality at optimality, due to a quadratic example.

What does a worst-case look like?

Quadratic worst-case function $f(x) = \frac{1}{2} x^\top \begin{pmatrix} \mu & 0 \\ 0 & L \end{pmatrix} x$:

What does a worst-case look like?

Quadratic worst-case function $f(x) = \frac{1}{2} x^\top \begin{pmatrix} \mu & 0 \\ 0 & L \end{pmatrix} x$:



Yoel Drori

"Efficient first-order methods for convex minimization:
a constructive approach"

Optimized gradient methods

Smooth convex minimization setting:

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

with f being L-smooth and convex, with black-box oracle $f'(.)$ available.

Optimized gradient methods

Smooth convex minimization setting:

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

with f being L-smooth and convex, with black-box oracle $f'(.)$ available.

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:

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

Optimized gradient methods

Smooth convex minimization setting:

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

with f being L-smooth and convex, with black-box oracle $f'(.)$ available.

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

$$f(x_N) - f(x_\star) \geqslant \frac{L\|x_0 - x_\star\|^2}{2\theta_N^2} = O(1/N^2),$$

with $\theta_0 = 1$, and:

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

Optimized gradient methods

Smooth convex minimization setting:

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

with f being L-smooth and convex, with black-box oracle $f'(.)$ available.

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

$$f(x_N) - f(x_\star) \geqslant \frac{L\|x_0 - x_\star\|^2}{2\theta_N^2} = O(1/N^2),$$

with $\theta_0 = 1$, and:

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

Coherent with historical lower bounds (Nemirovski & Yudin 1983) and optimal methods (Nemirovski 1982), (Nesterov 1983).

Optimized gradient methods

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

Greedy First-order Method (GFOM)

Inputs: f, x_0, N.

For $i = 1, 2, \ldots$

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

Worst-case guarantee:

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

Optimized gradient methods

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

Optimized gradient method with exact line-search

Inputs: f, x_0, N.

For $i = 1, \ldots, N$

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

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

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

$$x_i = y_i + \alpha d_i$$

Worst-case guarantee:

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

Optimized gradient methods

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

Optimized gradient method

Inputs: f, x_0, N.

For $i = 1, \ldots, N$

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

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

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

Worst-case guarantee:

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

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

Proof

Combine

- interpolation conditions for $i, j \in \{\star, 0, \ldots, N\}$

$$f(x_i) \geqslant f(x_j) + \langle f'(x_j), x_i - x_j \rangle + \frac{1}{2L} \| f'(x_i) - f'(x_j) \|^2$$

- optimality conditions for span searches

$$\begin{aligned} \langle f'(x_i), f'(x_j) \rangle &= 0 \qquad 0 \leqslant j < i \leqslant N \\ \langle f'(x_i), x_j - x_i \rangle &= 0 \qquad 1 \leqslant j \leqslant i \leqslant N \end{aligned}$$

with appropriate weights.

Proof

Combine

- ⋄ interpolation conditions for $i, j \in \{\star, 0, \ldots, N\}$

$$f(x_i) \geqslant f(x_j) + \langle f'(x_j), x_i - x_j \rangle + \tfrac{1}{2L} \|f'(x_i) - f'(x_j)\|^2$$

- ⋄ optimality conditions for span searches

$$\begin{aligned} \langle f'(x_i), f'(x_j) \rangle &= 0 \qquad 0 \leqslant j < i \leqslant N \\ \langle f'(x_i), x_j - x_i \rangle &= 0 \qquad 1 \leqslant j \leqslant i \leqslant N \end{aligned}$$

with appropriate weights. Weighted sum can be rewritten exactly as:

$$f(x_N) - f(x_\star) \leqslant \frac{L\|x_0 - x_\star\|^2}{2\theta_N^2} - \frac{L}{2\theta_N^2} \left\| x_0 - x_\star - \frac{\theta_N}{L} f'(x_N) - \frac{2}{L} \sum_{i=0}^{N-1} \theta_i f'(x_i) \right\|^2$$

Proof

Combine

- ◇ interpolation conditions for $i, j \in \{\star, 0, \ldots, N\}$

$$f(x_i) \geqslant f(x_j) + \left\langle f'(x_j), x_i - x_j \right\rangle + \tfrac{1}{2L} \left\| f'(x_i) - f'(x_j) \right\|^2$$

- ◇ optimality conditions for span searches

$$\begin{aligned} \left\langle f'(x_i), f'(x_j) \right\rangle &= 0 \qquad 0 \leqslant j < i \leqslant N \\ \left\langle f'(x_i), x_j - x_i \right\rangle &= 0 \qquad 1 \leqslant j \leqslant i \leqslant N \end{aligned}$$

with appropriate weights. Weighted sum can be rewritten exactly as:

$$f(x_N) - f(x_\star) \leqslant \frac{L\|x_0 - x_\star\|^2}{2\theta_N^2} - \frac{L}{2\theta_N^2} \left\| x_0 - x_\star - \frac{\theta_N}{L} f'(x_N) - \frac{2}{L} \sum_{i=0}^{N-1} \theta_i f'(x_i) \right\|^2$$

Proof for GFOM actually valid for a family of methods, that includes OGM.

Avoiding semidefinite programming modeling steps?

Avoiding semidefinite programming modeling steps?

François Glineur

Julien Hendrickx

"Performance Estimation Toolbox (PESTO): automated worst-case analysis of first-order optimization methods"

PESTO example: an inexact fast gradient method

Minimize L-smooth convex function $f(x)$:

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

PESTO example: an inexact fast gradient method

Minimize L-smooth convex function $f(x)$:

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

Fast Gradient Method (FGM)

Input: $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$, $x_0 = y_0 \in \mathbb{R}^d$.

For $i = 0 : N-1$

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

$$y_{i+1} = x_{i+1} + \frac{i-1}{i+2}(x_{i+1} - x_i)$$

PESTO example: an inexact fast gradient method

Minimize L-smooth convex function $f(x)$:

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

Fast Gradient Method (FGM)

Input: $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$, $x_0 = y_0 \in \mathbb{R}^d$.

For $i = 0 : N - 1$

$$x_{i+1} = y_i - \frac{1}{L}\nabla f(y_i)$$
$$y_{i+1} = x_{i+1} + \frac{i-1}{i+2}(x_{i+1} - x_i)$$

What if inexact gradient used instead? Relative inaccuracy model:

$$\|\tilde{\nabla} f(y_i) - \nabla f(y_i)\| \leq \varepsilon \|\nabla f(y_i)\|.$$

PESTO example: an inexact fast gradient method

Minimize L-smooth convex function $f(x)$:

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

Fast Gradient Method (FGM)

Input: $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$, $x_0 = y_0 \in \mathbb{R}^d$.

For $i = 0 : N-1$

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

$$y_{i+1} = x_{i+1} + \frac{i-1}{i+2}(x_{i+1} - x_i)$$

What if inexact gradient used instead? Relative inaccuracy model:

$$\|\tilde{\nabla} \mathbf{f}(\mathbf{y_i}) - \nabla f(y_i)\| \leq \varepsilon \|\nabla f(y_i)\|.$$

PESTO example: an inexact fast gradient method

Minimize L-smooth convex function $f(x)$:

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

Fast Gradient Method (FGM)

Input: $f \in \mathcal{F}_{0,L}(\mathbb{R}^d)$, $x_0 = y_0 \in \mathbb{R}^d$.

For $i = 0 : N - 1$

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

$$y_{i+1} = x_{i+1} + \frac{i-1}{i+2}(x_{i+1} - x_i)$$

What if inexact gradient used instead? Relative inaccuracy model:

$$\|\tilde{\nabla} \mathbf{f}(\mathbf{y_i}) - \nabla f(y_i)\| \leq \varepsilon \|\nabla f(y_i)\|.$$

PESTO example: an inexact fast gradient method

```
% (0) Initialize an empty PEP
P = pep();

% (1) Set up the objective function
param.mu = 0;      % strong convexity parameter
param.L  = 1;      % Smoothness parameter

F=P.DeclareFunction('SmoothStronglyConvex',param); % F is the objective function

% (2) Set up the starting point and initial condition
x0       = P.StartingPoint();        % x0 is some starting point
[xs, fs] = F.OptimalPoint();         % xs is an optimal point, and fs=F(xs)
P.InitialCondition((x0-xs)^2 <= 1);  % Add an initial condition ||x0-xs||^2<= 1

% (3) Algorithm
N = 7; % number of iterations

x    = cell(N+1,1); % we store the iterates in a cell for convenience
x{1} = x0;
y    = x0;
eps  = .1;
for i = 1:N
    d      = inexactsubgradient(y, F, eps);
    x{i+1} = y - 1/param.L * d;
    y      = x{i+1} + (i-1)/(i+2) * (x{i+1} - x{i});
end

% (4) Set up the performance measure
[g, f] = F.oracle(x{N+1});    % g=grad F(x), f=F(x)
P.PerformanceMetric(f - fs); % Worst-case evaluated as F(x)-F(xs)

% (5) Solve the PEP
P.solve()

% (6) Evaluate the output
double(f - fs)   % worst-case objective function accuracy
```

PESTO example: an inexact fast gradient method

```
% (0) Initialize an empty PEP
P = pep();

% (1) Set up the objective function
param.mu = 0;      % strong convexity parameter
param.L  = 1;      % Smoothness parameter

F=P.DeclareFunction('SmoothStronglyConvex',param); % F is the objective function

% (2) Set up the starting point and initial condition
x0      = P.StartingPoint();       % x0 is some starting point
[xs, fs] = F.OptimalPoint();       % xs is an optimal point, and fs=F(xs)
                                                                  = 1
```

```
x{1} = x0;
y    = x0;
eps  = .1;
for i = 1:N
    d       = inexactsubgradient(y, F, eps);
    x{i+1} = y - 1/param.L * d;
    y       = x{i+1} + (i-1)/(i+2) * (x{i+1} - x{i});
end
```

```
    y      = x{i+1} + (i-1)/(i+2) * (x{i+1} - x{i});
end

% (4) Set up the performance measure
[g, f] = F.oracle(x{N+1});    % g=grad F(x), f=F(x)
P.PerformanceMetric(f - fs); % Worst-case evaluated as F(x)-F(xs)

% (5) Solve the PEP
P.solve()

% (6) Evaluate the output
double(f - fs)   % worst-case objective function accuracy
```

PESTO example: an inexact fast gradient method

```
% (0) Initialize an empty PEP
P = pep();

% (1) Set up the objective function
param.mu = 0;      % strong convexity parameter
param.L  = 1;      % Smoothness parameter

F=P.DeclareFunction('SmoothStronglyConvex',param); % F is the objective function

% (2) Set up the starting point and initial condition
x0       = P.StartingPoint();       % x0 is some starting point
[xs, fs] = F.OptimalPoint();        % xs is an optimal point, and fs=F(xs)
```

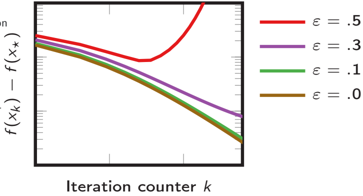
```
x{1} = x0;
y    = x0;
eps  = .1;
for i = 1:N
    d       = inexactsubgradient(y, F, eps);
    x{i+1} = y - 1/param.L * d;
    y       = x{i+1} + (i-1)/(i+2) * (x{i+1} - x{i});
end
```

```
    y       = x{i+1} + (i-1)/(i+2) * (x{i+1} - x{i});
end

% (4) Set up the performance measure
[g, f] = F.oracle(x{N+1});    % g=grad F(x), f=F(x)
P.PerformanceMetric(f - fs); % Worst-case evaluated as F(x)-F(xs)

% (5) Solve the PEP
P.solve()

% (6) Evaluate the output
double(f - fs)   % worst-case objective function accuracy
```



Current examples within PESTO

Current examples within PESTO

Includes...

- ◇ subgradient, gradient, heavy-ball, fast gradient, optimized gradient methods,
- ◇ projected and proximal variants, and accelerated/momentum versions,
- ◇ steepest descent, greedy/conjugate gradient methods,
- ◇ Douglas-Rachford/three operator splitting,
- ◇ Frank-Wolfe/conditional gradient,
- ◇ inexact versions of gradient/fast gradient,
- ◇ Krasnoselskii-Mann and Halpern fixed-point iterations,
- ◇ mirror descent/Bregman gradient/NoLips,
- ◇ stochastic methods: SAG, SAGA, SGD, and some variants.

Current examples within PESTO

Includes...

- ◇ subgradient, gradient, heavy-ball, fast gradient, optimized gradient methods,
- ◇ projected and proximal variants, and accelerated/momentum versions,
- ◇ steepest descent, greedy/conjugate gradient methods,
- ◇ Douglas-Rachford/three operator splitting,
- ◇ Frank-Wolfe/conditional gradient,
- ◇ inexact versions of gradient/fast gradient,
- ◇ Krasnoselskii-Mann and Halpern fixed-point iterations,
- ◇ mirror descent/Bregman gradient/NoLips,
- ◇ stochastic methods: SAG, SAGA, SGD, and some variants.

PESTO contains most of recent PEP-related advances (including techniques by other groups). Clean updated references in user manual.

Current examples within PESTO

Includes...

- ◇ subgradient, gradient, heavy-ball, fast gradient, optimized gradient methods,
- ◇ projected and proximal variants, and accelerated/momentum versions,
- ◇ steepest descent, greedy/conjugate gradient methods,
- ◇ Douglas-Rachford/three operator splitting,
- ◇ Frank-Wolfe/conditional gradient,
- ◇ inexact versions of gradient/fast gradient,
- ◇ Krasnoselskii-Mann and Halpern fixed-point iterations,
- ◇ mirror descent/Bregman gradient/NoLips,
- ◇ stochastic methods: SAG, SAGA, SGD, and some variants.

PESTO contains most of recent PEP-related advances (including techniques by other groups). Clean updated references in user manual.

Among others, see works by Drori, Teboulle, Kim, Fessler, Lieder, Lessard, Recht, Packard, Van Scoy, Hu, Cyrus, Gu, Yang, etc.

Current examples within PESTO

Includes...

- ◇ subgradient, gradient, heavy-ball, fast gradient, optimized gradient methods,
- ◇ projected and proximal variants, and accelerated/momentum versions,
- ◇ steepest descent, greedy/conjugate gradient methods,
- ◇ Douglas-Rachford/three operator splitting,
- ◇ Frank-Wolfe/conditional gradient,
- ◇ inexact versions of gradient/fast gradient,
- ◇ Krasnoselskii-Mann and Halpern fixed-point iterations,
- ◇ mirror descent/Bregman gradient/NoLips,
- ◇ stochastic methods: SAG, SAGA, SGD, and some variants.

PESTO contains most of recent PEP-related advances (including techniques by other groups). Clean updated references in user manual.

Among others, see works by Drori, Teboulle, Kim, Fessler, Lieder, Lessard, Recht, Packard, Van Scoy, Hu, Cyrus, Gu, Yang, etc.

If you have additional examples, we would be glad to add them!

Toy example: gradient descent

A few examples

Simplified proofs?

Concluding remarks and perspectives

Francis Bach

"Stochastic first-order methods: non-asymptotic and computer-aided analyses via potential functions"

Some opinions on PEPs

Pros/cons of PEPs

Some opinions on PEPs

Pros/cons of PEPs

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

Some opinions on PEPs

Pros/cons of PEPs

- ☺ Worst-case guarantees *cannot be improved*,
- ☺ fair amount of generalizations (finite sums, constraints, prox, etc.),

Some opinions on PEPs

Pros/cons of PEPs

- ☺ Worst-case guarantees *cannot be improved*,
- ☺ fair amount of generalizations (finite sums, constraints, prox, etc.),
- ☺ allows reaching proofs that could barely be obtained (or intuited) by hand,

Some opinions on PEPs

Pros/cons of PEPs

- ☺ Worst-case guarantees *cannot be improved*,
- ☺ fair amount of generalizations (finite sums, constraints, prox, etc.),
- ☺ allows reaching proofs that could barely be obtained (or intuited) by hand,
- ☹ SDPs typically become prohibitively large (with N and generalizations),

Some opinions on PEPs

Pros/cons of PEPs

- ☺ Worst-case guarantees *cannot be improved*,
- ☺ fair amount of generalizations (finite sums, constraints, prox, etc.),
- ☺ allows reaching proofs that could barely be obtained (or intuited) by hand,
- ☹ SDPs typically become prohibitively large (with N and generalizations),
- ☹ proofs (may be) quite involved and hard to intuit,

Some opinions on PEPs

Pros/cons of PEPs

- ☺ Worst-case guarantees *cannot be improved*,
- ☺ fair amount of generalizations (finite sums, constraints, prox, etc.),
- ☺ allows reaching proofs that could barely be obtained (or intuited) by hand,
- ☹ SDPs typically become prohibitively large (with N and generalizations),
- ☹ proofs (may be) quite involved and hard to intuit,
- ☹ proofs (may be) hard to generalize (e.g., to handle projections, backtracking),

Some opinions on PEPs

Pros/cons of PEPs

- ☺ Worst-case guarantees *cannot be improved*,
- ☺ fair amount of generalizations (finite sums, constraints, prox, etc.),
- ☺ allows reaching proofs that could barely be obtained (or intuited) by hand,
- ☹ SDPs typically become prohibitively large (with N and generalizations),
- ☹ proofs (may be) quite involved and hard to intuit,
- ☹ proofs (may be) hard to generalize (e.g., to handle projections, backtracking),
- ☺ possible to "force" simple proofs (typically at some cost: e.g., loosing tightness).

Potential functions

What guarantees for gradient descent when minimizing a L-smooth convex function

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

Potential functions

What guarantees for gradient descent when minimizing a L-smooth convex function

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

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

Potential functions

What guarantees for gradient descent when minimizing a L-smooth convex function

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

It is 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

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

see e.g., (Bansal & Gupta 2019).

Potential functions

What guarantees for gradient descent when minimizing a L-smooth convex function

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

It is 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

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

see e.g., (Bansal & Gupta 2019).

Why is that nice? Very simple resulting proof:

Potential functions

What guarantees for gradient descent when minimizing a L-smooth convex function

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

It is 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

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

see e.g., (Bansal & Gupta 2019).

Why is that nice? Very simple resulting proof:

$$\phi^f_N \leqslant \phi^f_{N-1} \leqslant \ldots \leqslant \phi^f_0$$

Potential functions

What guarantees for gradient descent when minimizing a L-smooth convex function

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

It is 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

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

see e.g., (Bansal & Gupta 2019).

Why is that nice? Very simple resulting proof:

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

Potential functions

What guarantees for gradient descent when minimizing a L-smooth convex function

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

It is 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 2019).

Why is that nice? Very simple resulting proof:

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

Potential functions

What guarantees for gradient descent when minimizing a L-smooth convex function

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

It is 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 2019).

Why is that nice? Very simple resulting proof:

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

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

Potential functions

What guarantees for gradient descent when minimizing a L-smooth convex function

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

It is 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 2019).

Why is that nice? Very simple resulting proof:

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

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

Potentials are not new; see e.g., Nesterov (1983), Beck & Teboulle (2009), Hu & Lessard (2017), Bansal & Gupta (2019).

How does it work for the gradient method?

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

How does it work for the gradient method?

Gradient descent, take II: how to bound $\|f'(x_N)\|^2$ 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 $\|f'(x_N)\|^2$ 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

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

How does it work for the gradient method?

Gradient descent, take II: how to bound $\|f'(x_N)\|^2$ 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

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

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 $\|f'(x_N)\|^2$ 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

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

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

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

How does it work for the gradient method?

Gradient descent, take II: how to bound $\|f'(x_N)\|^2$ 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

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

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

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

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 $d \in \mathbb{N}$:

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

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^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.

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.

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?

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*!
- idea: apply previous reformulation tricks to reformulate:

$$0 \geqslant \max_f \phi^f_{k+1} - \phi^f_k.$$

  Dual is a feasibility problem, linear in $\{a_k, b_k, c_k, d_k\}_k$.

How does it work for the gradient method?

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

How does it work for the gradient method?

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

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

How does it work for the gradient method?

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

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

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

How does it work for the gradient method?

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

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

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

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

How does it work for the gradient method?

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

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

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

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

Question: largest provable b_N using such potentials?

How does it work for the gradient method?

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

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

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

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

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 $\|f'(x_N)\|^2$; choose

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

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

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

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 $\|f'(x_N)\|^2$; choose

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

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

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

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 $\|f'(x_N)\|^2$; choose

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

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

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

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 $\|f'(x_N)\|^2$; choose

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

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

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

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 $\|f'(x_N)\|^2$; choose

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

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

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

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:

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

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

How does it work for the gradient method?

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

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

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

How does it work for the gradient method?

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

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

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

How does it work for the gradient method?

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

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

$N =$	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:

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

$$\begin{array}{rccc} N = & 1 & 2 & 3 \\ b_N = & 4 & 9 & 16 \end{array}$$

How does it work for the gradient method?

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

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

$N =$	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:

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

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

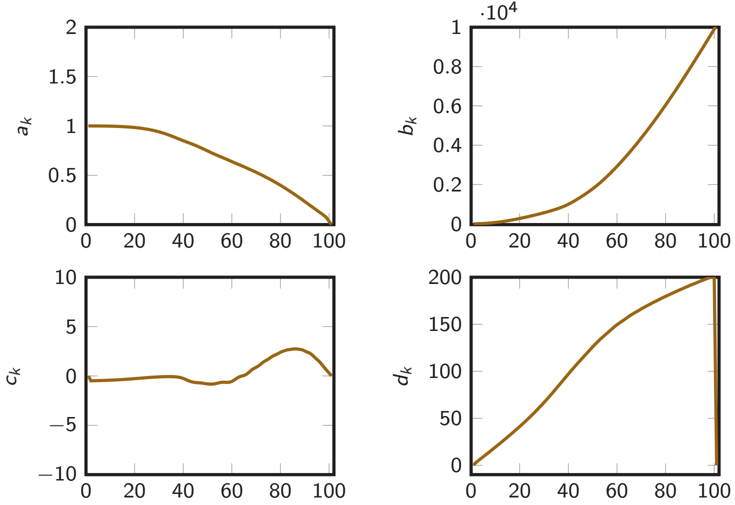
2. Observe the a_k, b_k, c_k, d_k's for some values of N.

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

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

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

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



How does it work for the gradient method?

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

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

$N =$	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?

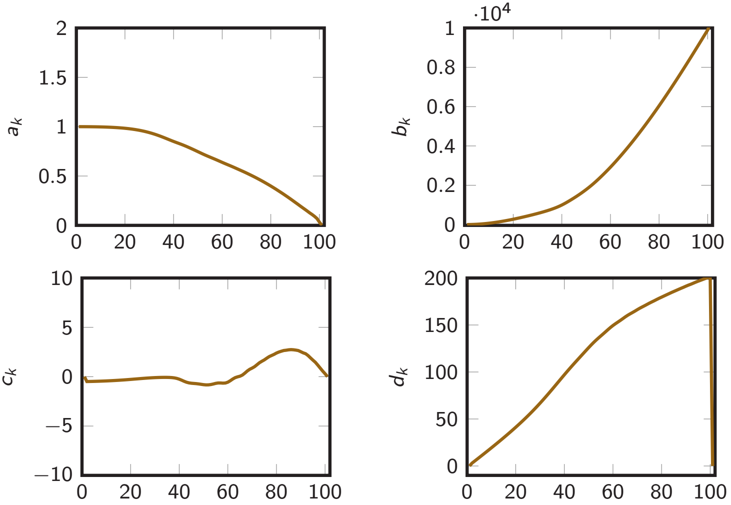
1. Solve the SDP for some values of N; recall final guarantee of the form:

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

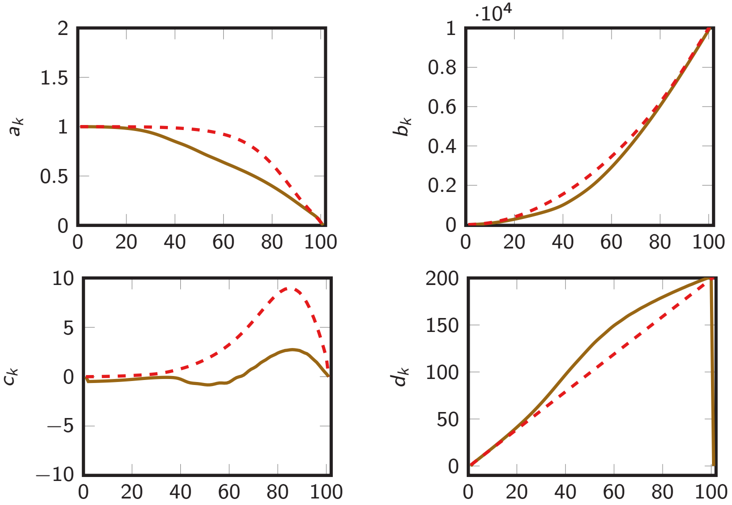
$$\begin{array}{rccccc} N = & 1 & 2 & 3 & 4 & \ldots & 100 \\ b_N = & 4 & 9 & 16 & 25 & \ldots & 10201 \end{array}$$

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.
   Tentative simplification #1: $d_k = (2k+1)L$
   Tentative simplification #2: $a_k = L^2$, $c_k = 0$
   Tentative simplification #3: $d_k = 0$

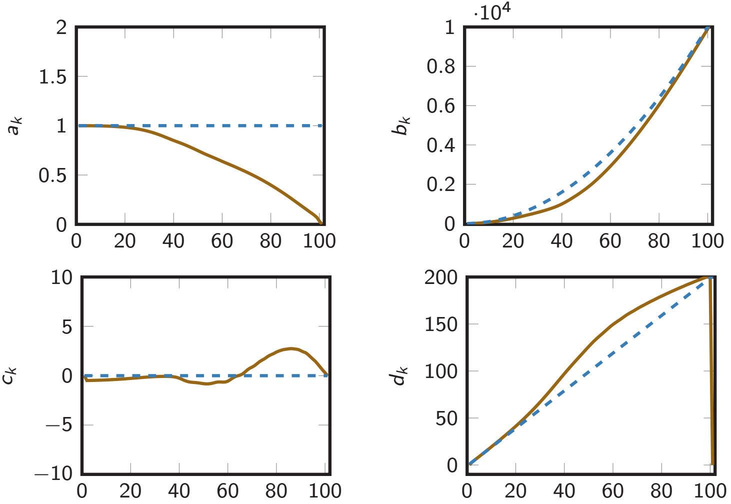
$$V_k = \begin{pmatrix} x_k - x_\star \\ f'(x_k) \end{pmatrix}^\top \left[\begin{pmatrix} a_k & c_k \\ c_k & b_k \end{pmatrix} \otimes I_d \right] \begin{pmatrix} x_k - x_\star \\ f'(x_k) \end{pmatrix} + d_k \left(f(x_k) - f(x_\star) \right)$$



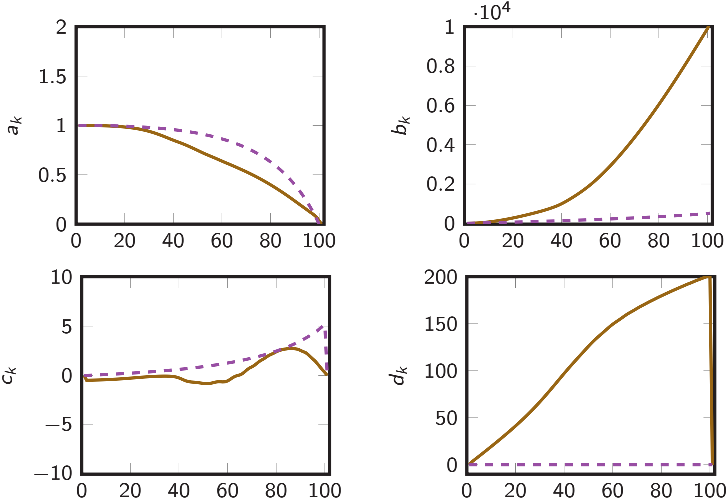
$$V_k = \begin{pmatrix} x_k - x_\star \\ f'(x_k) \end{pmatrix}^\top \left[\begin{pmatrix} a_k & c_k \\ c_k & b_k \end{pmatrix} \otimes I_d \right] \begin{pmatrix} x_k - x_\star \\ f'(x_k) \end{pmatrix} + (2k+1)L\left(f(x_k) - f(x_\star)\right)$$



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



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



How does it work for the gradient method?

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

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

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

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.

   Tentative simplification #1: $d_k = (2k+1)L$
   Tentative simplification #2: $a_k = L^2$, $c_k = 0$
   Tentative simplification #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:

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

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

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.
   Tentative simplification #1: $d_k = (2k+1)L$ [success]
   Tentative simplification #2: $a_k = L^2$, $c_k = 0$ [success]
   Tentative simplification #3: $d_k = 0$ [fail]

How does it work for the gradient method?

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

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

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

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.

   Tentative simplification #1: $d_k = (2k+1)L$ [success]
   Tentative simplification #2: $a_k = L^2$, $c_k = 0$ [success]
   Tentative simplification #3: $d_k = 0$ [fail]
4. Prove target result by analytically playing with $\mathcal{V}_k$:

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

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

Potential functions

Simpler proof structures:

Potential functions

Simpler proof structures:

- ⋄ allow keeping SDP formulations more tractable,

Potential functions

Simpler proof structures:

- allow keeping SDP formulations more tractable,
- hence usable with more complex settings (e.g., randomizations, stochasticity).

Potential functions

Simpler proof structures:

- allow keeping SDP formulations more tractable,
- hence usable with more complex settings (e.g., randomizations, stochasticity).

More examples:

Potential functions

Simpler proof structures:

- allow keeping SDP formulations more tractable,
- hence usable with more complex settings (e.g., randomizations, stochasticity).

More examples:

- all previous variants (everything that fits into regular PEPs)

Potential functions

Simpler proof structures:

- ◇ allow keeping SDP formulations more tractable,
- ◇ hence usable with more complex settings (e.g., randomizations, stochasticity).

More examples:

- ◇ all previous variants (everything that fits into regular PEPs)
- ◇ stochastic variants (e.g., finite sum, bounded variance, over-parametrization),

Potential functions

Simpler proof structures:

- allow keeping SDP formulations more tractable,
- hence usable with more complex settings (e.g., randomizations, stochasticity).

More examples:

- all previous variants (everything that fits into regular PEPs)
- stochastic variants (e.g., finite sum, bounded variance, over-parametrization),
- randomized block-coordinate variants,

Potential functions

Simpler proof structures:

- allow keeping SDP formulations more tractable,
- hence usable with more complex settings (e.g., randomizations, stochasticity).

More examples:

- all previous variants (everything that fits into regular PEPs)
- stochastic variants (e.g., finite sum, bounded variance, over-parametrization),
- randomized block-coordinate variants,

... but also for designing methods!

Toy example: gradient descent

A few examples

Simplified proofs?

Concluding remarks and perspectives

Take-home message

Finding a worst-case $\equiv$ solving an optimization problem

Take-home message

Finding a worst-case $\equiv$ solving an optimization problem

Duality between worst-case scenarios & combinations of inequalities!

Take-home message

Finding a worst-case $\equiv$ solving an optimization problem

Duality between worst-case scenarios & combinations of inequalities!

PEP: a way to "brute-force" & "benchmark" such proofs.

Concluding remarks

Performance estimation:

Concluding remarks

Performance estimation:

- numerically allows obtaining tight bounds (rigorous baselines),

Concluding remarks

Performance estimation:

- numerically allows obtaining tight bounds (rigorous baselines),
- results can only be improved by changing algorithm and/or assumptions,

Concluding remarks

Performance estimation:

- ◇ numerically allows obtaining tight bounds (rigorous baselines),
- ◇ results can only be improved by changing algorithm and/or assumptions,
- ◇ helps designing analytical proofs (reduces to linear combinations of inequalities),

  proofs can be engineered using numerics & symbolic computations!

Concluding remarks

Performance estimation:

- ◇ numerically allows obtaining tight bounds (rigorous baselines),
- ◇ results can only be improved by changing algorithm and/or assumptions,
- ◇ helps designing analytical proofs (reduces to linear combinations of inequalities),

  proofs can be engineered using numerics & symbolic computations!
- ◇ fast prototyping:

Concluding remarks

Performance estimation:

- ◇ numerically allows obtaining tight bounds (rigorous baselines),
- ◇ results can only be improved by changing algorithm and/or assumptions,
- ◇ helps designing analytical proofs (reduces to linear combinations of inequalities),

  proofs can be engineered using numerics & symbolic computations!
- ◇ fast prototyping:

  before trying to prove your new FO method works; give PEP a try!

Concluding remarks

Performance estimation:

- numerically allows obtaining tight bounds (rigorous baselines),
- results can only be improved by changing algorithm and/or assumptions,
- helps designing analytical proofs (reduces to linear combinations of inequalities),

  proofs can be engineered using numerics & symbolic computations!
- fast prototyping:

  before trying to prove your new FO method works; give PEP a try!
- step forward to "reproducible theory".

Concluding remarks

Performance estimation:

- numerically allows obtaining tight bounds (rigorous baselines),
- results can only be improved by changing algorithm and/or assumptions,
- helps designing analytical proofs (reduces to linear combinations of inequalities),

  proofs can be engineered using numerics & symbolic computations!
- fast prototyping:

  before trying to prove your new FO method works; give PEP a try!
- step forward to “reproducible theory”.

Difficulties:

Concluding remarks

Performance estimation:

- numerically allows obtaining tight bounds (rigorous baselines),
- results can only be improved by changing algorithm and/or assumptions,
- helps designing analytical proofs (reduces to linear combinations of inequalities),
  proofs can be engineered using numerics & symbolic computations!
- fast prototyping:
  before trying to prove your new FO method works; give PEP a try!
- step forward to "reproducible theory".

Difficulties:

- suffers from standard caveats of worst-case analyses,
  key is to find good assumptions/parametrization

Concluding remarks

Performance estimation:

- ◇ numerically allows obtaining tight bounds (rigorous baselines),
- ◇ results can only be improved by changing algorithm and/or assumptions,
- ◇ helps designing analytical proofs (reduces to linear combinations of inequalities),

  proofs can be engineered using numerics & symbolic computations!
- ◇ fast prototyping:

  before trying to prove your new FO method works; give PEP a try!
- ◇ step forward to "reproducible theory".

Difficulties:

- ◇ suffers from standard caveats of worst-case analyses,

  key is to find good assumptions/parametrization
- ◇ closed-form solutions might be involved (if we care about tightness).

Concluding remarks

Performance estimation:

- numerically allows obtaining tight bounds (rigorous baselines),
- results can only be improved by changing algorithm and/or assumptions,
- helps designing analytical proofs (reduces to linear combinations of inequalities),
  proofs can be engineered using numerics & symbolic computations!
- fast prototyping:
  before trying to prove your new FO method works; give PEP a try!
- step forward to "reproducible theory".

Difficulties:

- suffers from standard caveats of worst-case analyses,
  key is to find good assumptions/parametrization
- closed-form solutions might be involved (if we care about tightness).

Ongoing research directions, open questions:

Concluding remarks

Performance estimation:

- numerically allows obtaining tight bounds (rigorous baselines),
- results can only be improved by changing algorithm and/or assumptions,
- helps designing analytical proofs (reduces to linear combinations of inequalities),
  proofs can be engineered using numerics & symbolic computations!
- fast prototyping:
  before trying to prove your new FO method works; give PEP a try!
- step forward to "reproducible theory".

Difficulties:

- suffers from standard caveats of worst-case analyses,
  key is to find good assumptions/parametrization
- closed-form solutions might be involved (if we care about tightness).

Ongoing research directions, open questions:

- computer-assisted algorithmic design,

Concluding remarks

Performance estimation:

- numerically allows obtaining tight bounds (rigorous baselines),
- results can only be improved by changing algorithm and/or assumptions,
- helps designing analytical proofs (reduces to linear combinations of inequalities),
  proofs can be engineered using numerics & symbolic computations!
- fast prototyping:
  before trying to prove your new FO method works; give PEP a try!
- step forward to "reproducible theory".

Difficulties:

- suffers from standard caveats of worst-case analyses,
  key is to find good assumptions/parametrization
- closed-form solutions might be involved (if we care about tightness).

Ongoing research directions, open questions:

- computer-assisted algorithmic design,
- adaptive & structure-exploiting methods,

Concluding remarks

Performance estimation:

- numerically allows obtaining tight bounds (rigorous baselines),
- results can only be improved by changing algorithm and/or assumptions,
- helps designing analytical proofs (reduces to linear combinations of inequalities),

  proofs can be engineered using numerics & symbolic computations!
- fast prototyping:

  before trying to prove your new FO method works; give PEP a try!
- step forward to "reproducible theory".

Difficulties:

- suffers from standard caveats of worst-case analyses,

  key is to find good assumptions/parametrization
- closed-form solutions might be involved (if we care about tightness).

Ongoing research directions, open questions:

- computer-assisted algorithmic design,
- adaptive & structure-exploiting methods,
- non-convex & non-Euclidean settings?

Concluding remarks

Performance estimation:

- ◇ numerically allows obtaining tight bounds (rigorous baselines),
- ◇ results can only be improved by changing algorithm and/or assumptions,
- ◇ helps designing analytical proofs (reduces to linear combinations of inequalities),
  proofs can be engineered using numerics & symbolic computations!
- ◇ fast prototyping:
  before trying to prove your new FO method works; give PEP a try!
- ◇ step forward to "reproducible theory".

Difficulties:

- ◇ suffers from standard caveats of worst-case analyses,
  key is to find good assumptions/parametrization
- ◇ closed-form solutions might be involved (if we care about tightness).

Ongoing research directions, open questions:

- ◇ computer-assisted algorithmic design,
- ◇ adaptive & structure-exploiting methods,
- ◇ non-convex & non-Euclidean settings?
- ◇ Higher order methods?

Concluding remarks

Concluding remarks

A few other recent directions (on my webpage):

- Simplified proofs (Lyapunov functions and potentials),

Concluding remarks

A few other recent directions (on my webpage):

- ◇ Simplified proofs (Lyapunov functions and potentials),
- ◇ Stochastic/randomized methods,

Concluding remarks

A few other recent directions (on my webpage):

- ◇ Simplified proofs (Lyapunov functions and potentials),
- ◇ Stochastic/randomized methods,

Concluding remarks

A few other recent directions (on my webpage):

- ⋄ Simplified proofs (Lyapunov functions and potentials),
- ⋄ Stochastic/randomized methods,
- ⋄ Mirror descent/Bregman gradient/NoLips/...

Concluding remarks

A few other recent directions (on my webpage):

- ◇ Simplified proofs (Lyapunov functions and potentials),
- ◇ Stochastic/randomized methods,
- ◇ Mirror descent/Bregman gradient/NoLips/...
- ◇ Monotone inclusions, splitting methods,

Concluding remarks

A few other recent directions (on my webpage):

- Simplified proofs (Lyapunov functions and potentials),
- Stochastic/randomized methods,
- Mirror descent/Bregman gradient/NoLips/...
- Monotone inclusions, splitting methods,
- Our first attempts to the analysis of adaptive methods (Polyak step sizes & line-searches).

Concluding remarks

A few other recent directions (on my webpage):

- ◇ Simplified proofs (Lyapunov functions and potentials),
- ◇ Stochastic/randomized methods,
- ◇ Mirror descent/Bregman gradient/NoLips/...
- ◇ Monotone inclusions, splitting methods,
- ◇ Our first attempts to the analysis of adaptive methods (Polyak step sizes & line-searches).

Shameless advertisement:

- ◇ Radu-Alexandru Dragomir, T, Alexandre d'Aspremont, Jérôme Bolte. "Optimal complexity and certification of Bregman first-order methods". Preprint 2019.
- ◇ Mathieu Barré, T, Alexandre d'Aspremont. "Complexity Guarantees for Polyak Steps with Momentum". COLT 2020 (to appear).
- ◇ Ernest Ryu, T, Carolina Bergeling, Pontus Giselsson. "Operator splitting performance estimation: Tight contraction factors and optimal parameter selection". Siopt 2020 (to appear).

Main references

References more thoroughly treated in the papers. Explicitly mentioned in this presentation:

- ◇ Yurii Nesterov. "A method for solving the convex programming problem with convergence rate $O(1/k^2)$. Soviet Mathematics Doklady, 1983.
- ◇ Arkadi Nemirovskii, David Yudin. "Problem complexity and method efficiency in optimization". Wiley-Interscience, 1983.
- ◇ Amir Beck, Marc Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". SIAM Journal on Imaging Sciences, 2009.
- ◇ Yoel Drori, Marc Teboulle. "Performance of first-order methods for smooth convex minimization: a novel approach". Mathematical Programming, 2014.
- ◇ Donghwan Kim, Jeffrey Fessler. "Optimized first-order methods for smooth convex minimization". Mathematical Programming, 2016.
- ◇ Yoel Drori, Marc Teboulle. "An optimal variant of Kelley's cutting-plane method". Mathematical Programming, 2016.
- ◇ Laurent Lessard, Benjamin Recht, Andrew Packard. "Analysis and design of optimization algorithms via integral quadratic constraints". SIAM Journal on Optimization, 2016.
- ◇ Yoel Drori. "The exact information-based complexity of smooth convex minimization". Journal of Complexity, 2017.
- ◇ Bin Hu, Laurent Lessard. "Dissipativity Theory for Nesterov's Accelerated Method". ICML, 2017.
- ◇ Donghwan Kim, Jeffrey Fessler. "Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions". Preprint, 2018.
- ◇ Donghwan Kim. "Accelerated Proximal Point method for Maximally Monotone Operators". Preprint, 2019.
- ◇ Nikhil Bansal, Anupam Gupta. "Potential-function proofs for first-order methods". Theory of Computing, 2019.

Thanks! Questions?

www.di.ens.fr/∼ataylor/

AdrienTaylor/Performance-Estimation-Toolbox on Github

Presentation mainly based on:

- T, François Glineur, Julien Hendrickx. "Smooth strongly convex interpolation and exact worst-case performance of first-order methods". Mathematical Programming, 2017.
- Etienne de Klerk, François Glineur, T. "On the worst-case complexity of the gradient method with exact line search for smooth strongly convex functions". Optimization Letters, 2017.
- T, François Glineur, Julien Hendrickx. "Performance Estimation Toolbox (PESTO): automated worst-case analysis of first-order optimization methods" CDC, 2017.
- Yoel Drori, T. "Efficient first-order methods for convex minimization: a constructive approach". Mathematical Programming, 2019.
- T, Francis Bach. "Stochastic first-order methods: non-asymptotic and computer-aided analyses via potential functions". COLT, 2019.