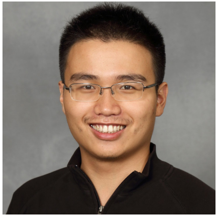
New first-order methods in modern/classical settings

Daniels School of Business Quantitative Methods Seminar

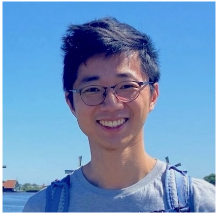
September 2023



Lijun Ding
UW-Madison

Ben Grimmer
Johns Hopkins

Kevin Shu
GA Tech



Alex L. Wang
Purdue University

Part 1

Sharp exact penalty formulations in signal recovery

Joint work with Lijun Ding

Outline

- Motivation: Sparse recovery and low-rank covariance estimation

Outline

- Motivation: Sparse recovery and low-rank covariance estimation
  - $\longrightarrow$ Abstract signal recovery problem

Outline

- Motivation: Sparse recovery and low-rank covariance estimation
  - $\longrightarrow$ Abstract signal recovery problem
- A new formulation of the abstract problem that is sharp

Outline

- Motivation: Sparse recovery and low-rank covariance estimation
  - $\longrightarrow$ Abstract signal recovery problem
- A new formulation of the abstract problem that is sharp
  - Better robustness guarantees, faster algorithms

Outline

- Motivation: Sparse recovery and low-rank covariance estimation
  - $\longrightarrow$ Abstract signal recovery problem
- A new formulation of the abstract problem that is sharp
  - Better robustness guarantees, faster algorithms
- Numerical results

Motivation: Sparse recovery and covariance estimation

Sparse recovery setup

- **Recovery task**: Recover $x^\sharp \in \mathbb{R}^n$ from $A \in \mathbb{R}^{m\times n}, b = Ax^\sharp$

Related: Candes and Tao [2005], Recht et al. [2010], Candès et al. [2013]

Sparse recovery setup

- **Recovery task**: Recover $x^\sharp \in \mathbb{R}^n$ from $A \in \mathbb{R}^{m\times n}, b = Ax^\sharp$
- Suppose A entrywise i.i.d. $N(0, 1/m^2)$

$$\left|\text{supp}(x^\sharp)\right| \le k \ll n \qquad m \asymp k\log(n)$$

Related: Candes and Tao [2005], Recht et al. [2010], Candès et al. [2013]

Sparse recovery setup

- **Recovery task**: Recover $x^\sharp \in \mathbb{R}^n$ from $A \in \mathbb{R}^{m\times n}, b = Ax^\sharp$
- Suppose A entrywise i.i.d. $N(0, 1/m^2)$

$$\left|\mathrm{supp}(x^\sharp)\right| \leq k \ll n \qquad m \asymp k \log(n)$$

- **Conceptual approach:** $\min_{x\in\mathbb{R}^n} \left\{ |\mathrm{supp}(x)| \;:\; Ax = b \right\}$

Related: Candes and Tao [2005], Recht et al. [2010], Candès et al. [2013]

Sparse recovery setup

- **Recovery task**: Recover $x^\sharp \in \mathbb{R}^n$ from $A \in \mathbb{R}^{m \times n}, b = Ax^\sharp$
- Suppose A entrywise i.i.d. $N(0, 1/m^2)$

$$\left|\mathrm{supp}(x^\sharp)\right| \leq k \ll n \qquad m \asymp k \log(n)$$

- **Conceptual approach:** $\min_{x \in \mathbb{R}^n} \left\{ |\mathrm{supp}(x)| \ : \ Ax = b \right\}$
- **Convex optimization approach**: In this regime, $x^\sharp$ is unique minimizer of

$$\min_{x \in \mathbb{R}^n} \left\{ \|x\|_1 \ : \ Ax = b \right\}$$

Related: Candes and Tao [2005], Recht et al. [2010], Candès et al. [2013]

Low-rank covariance estimation

- **Recovery task**: Recover $X^\sharp \in \mathbb{S}^n_+$ with $\mathrm{rank}(X^\sharp) \le k$ from $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$, $b = \mathcal{A}(X^\sharp)$

Related: Recht et al. [2010], Chen et al. [2015]

Low-rank covariance estimation

- **Recovery task**: Recover $X^\sharp \in \mathbb{S}^n_+$ with $\mathrm{rank}(X^\sharp) \le k$ from $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$, $b = \mathcal{A}(X^\sharp)$
- Suppose $\mathcal{A}^*(e_i) = a_i a_i^\intercal$ where $a_i \sim N(0, I_n/m)$ and $m \asymp nk$

Related: Recht et al. [2010], Chen et al. [2015]

Low-rank covariance estimation

- **Recovery task**: Recover $X^\sharp \in \mathbb{S}^n_+$ with $\mathrm{rank}(X^\sharp) \le k$ from $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$, $b = \mathcal{A}(X^\sharp)$
- Suppose $\mathcal{A}^*(e_i) = a_i a_i^\intercal$ where $a_i \sim N(0, I_n/m)$ and $m \asymp nk$
- Known as phase retrieval when $k = 1$

Related: Recht et al. [2010], Chen et al. [2015]

- **Recovery task**: Recover $X^\sharp \in \mathbb{S}^n_+$ with $\mathrm{rank}(X^\sharp) \le k$ from $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$, $b = \mathcal{A}(X^\sharp)$
- Suppose $\mathcal{A}^*(e_i) = a_i a_i^\intercal$ where $a_i \sim N(0, I_n/m)$ and $m \asymp nk$
- Known as phase retrieval when $k = 1$
- **Conceptual approach:** $\min\limits_{X \in \mathbb{S}^n} \left\{ \mathrm{rank}(X) : \begin{array}{l} \mathcal{A}(X) = b \\ X \succeq 0 \end{array} \right\}$

Related: Recht et al. [2010], Chen et al. [2015]

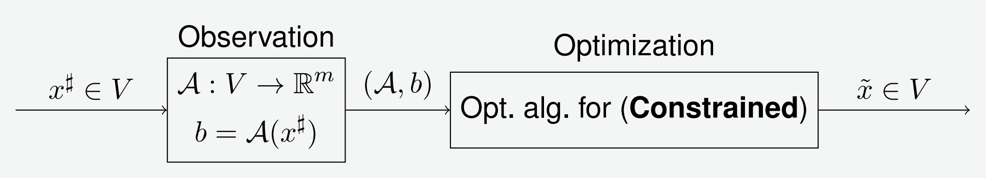
Low-rank covariance estimation

- **Recovery task**: Recover $X^\sharp \in \mathbb{S}^n_+$ with $\mathrm{rank}(X^\sharp) \leq k$ from $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$, $b = \mathcal{A}(X^\sharp)$
- Suppose $\mathcal{A}^*(e_i) = a_i a_i^\intercal$ where $a_i \sim N(0, I_n/m)$ and $m \asymp nk$
- Known as phase retrieval when $k = 1$
- **Conceptual approach:** $\min_{X \in \mathbb{S}^n} \left\{ \mathrm{rank}(X) : \begin{array}{l} \mathcal{A}(X) = b \\ X \succeq 0 \end{array} \right\}$
- **Convex optimization approach**: $X^\sharp$ is unique minimizer of

$$\min_{X \in \mathbb{S}^n} \left\{ \mathrm{tr}(X) : \begin{array}{l} \mathcal{A}(X) = b \\ X \succeq 0 \end{array} \right\}$$

Related: Recht et al. [2010], Chen et al. [2015]

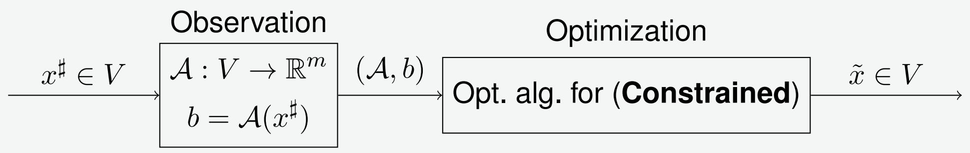
Abstract signal recovery problem and questions



(Constrained)

$$\min_{x \in V} \left\{ f(x) : \begin{array}{l} \mathcal{A}(x) = b \\ x \in K \end{array} \right\}$$

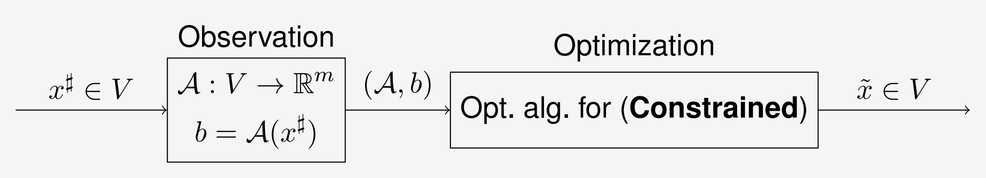
Abstract signal recovery problem and questions



(Constrained) $$\min_{x \in V} \left\{ f(x) : \begin{array}{l} \mathcal{A}(x) = b \\ x \in K \end{array} \right\}$$

- If no noise in sensing process and no error in optimization algorithm, $\tilde{x} = x^\sharp$

Abstract signal recovery problem and questions

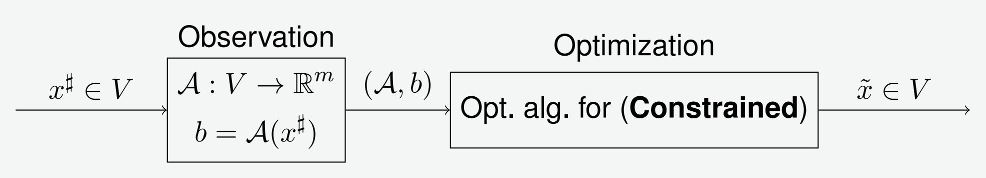


(Constrained)

$$\min_{x \in V} \left\{ f(x) : \begin{array}{l} \mathcal{A}(x) = b \\ x \in K \end{array} \right\}$$

- If no noise in sensing process and no error in optimization algorithm, $\tilde{x} = x^\sharp$
- **Questions**:

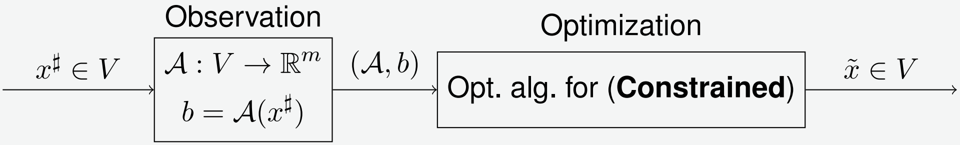
Abstract signal recovery problem and questions



(Constrained) $\displaystyle \min_{x \in V} \left\{ f(x) : \begin{array}{l} \mathcal{A}(x) = b \\ x \in K \end{array} \right\}$

- If no noise in sensing process and no error in optimization algorithm, $\tilde{x} = x^\sharp$
- **Questions**:
  - What if the algorithm receives $\tilde{b} = \mathcal{A}(x^\sharp) + \delta$?

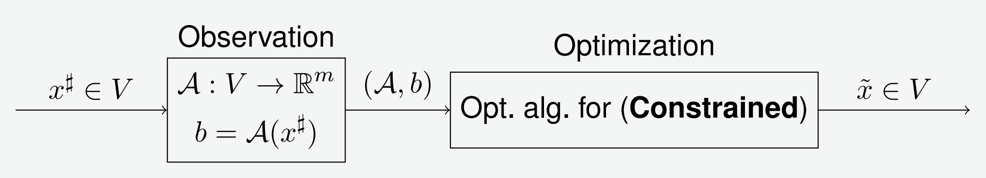
Abstract signal recovery problem and questions



(Constrained) $$\min_{x \in V} \left\{ f(x) : \begin{array}{l} \mathcal{A}(x) = b \\ x \in K \end{array} \right\}$$

- If no noise in sensing process and no error in optimization algorithm, $\tilde{x} = x^\sharp$
- **Questions**:
  - What if the algorithm receives $\tilde{b} = \mathcal{A}(x^\sharp) + \delta$?
  - What if algorithm only produces a ϵ-optimal and ϵ-feasible solution?

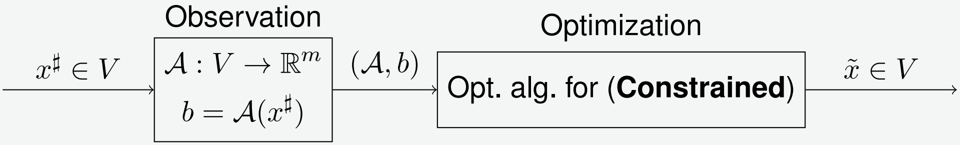
Abstract signal recovery problem and questions



(Constrained) $$\min_{x \in V} \left\{ f(x) : \begin{array}{l} \mathcal{A}(x) = b \\ x \in K \end{array} \right\}$$

- If no noise in sensing process and no error in optimization algorithm, $\tilde{x} = x^\sharp$
- **Questions**:
  - What if the algorithm receives $\tilde{b} = \mathcal{A}(x^\sharp) + \delta$?
  - What if algorithm only produces a ϵ-optimal and ϵ-feasible solution?
  - What algorithm?

Abstract signal recovery problem and questions



(Constrained)
$$\min_{x \in V} \left\{ f(x) : \begin{array}{l} \mathcal{A}(x) = b \\ x \in K \end{array} \right\}$$

- If no noise in sensing process and no error in optimization algorithm, $\tilde{x} = x^\sharp$
- **Questions**:
  - What if the algorithm receives $\tilde{b} = \mathcal{A}(x^\sharp) + \delta$?
  - What if algorithm only produces a ϵ-optimal and ϵ-feasible solution?
  - What algorithm?
  - Another convex problem?

A sharp penalty formulation

A penalty formulation

- **(Constrained)** $$\min_{x \in V} \left\{ f(x) : \begin{array}{l} \mathcal{A}(x) = b \\ x \in K \end{array} \right\}$$

Related: Beck and Teboulle [2009], Tibshirani [1996]

A penalty formulation

- **(Constrained)** $$\min_{x \in V} \left\{ f(x) : \begin{array}{l} \mathcal{A}(x) = b \\ x \in K \end{array} \right\}$$
- Penalty formulation: let $r \asymp \sqrt{k}$ be a penalty parameter

$$F(x) := f(x) + r \left\| \mathcal{A}(x) - b \right\|_1 + 2 \operatorname{dist}_1(x, K)$$

Related: Beck and Teboulle [2009], Tibshirani [1996]

A penalty formulation

- **(Constrained)** $$\min_{x\in V}\left\{ f(x) : \begin{array}{l} \mathcal{A}(x) = b \\ x \in K \end{array} \right\}$$
- Penalty formulation: let $r \asymp \sqrt{k}$ be a penalty parameter
$$F(x) := f(x) + r\,\|\mathcal{A}(x) - b\|_1 + 2\,\mathrm{dist}_1(x, K)$$
- Compare: Lasso $\|Ax - b\|_2^2$ vs $\|Ax - b\|_1$

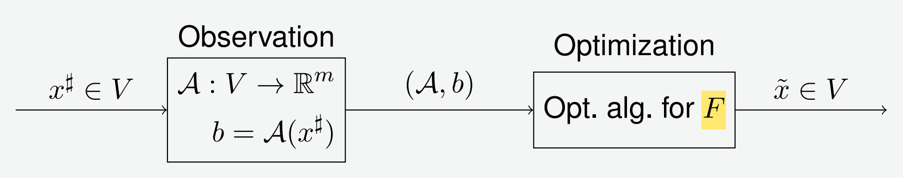
Related: Beck and Teboulle [2009], Tibshirani [1996]

A penalty formulation

- **(Constrained)** $$\min_{x \in V} \left\{ f(x) : \begin{array}{l} \mathcal{A}(x) = b \\ x \in K \end{array} \right\}$$
- Penalty formulation: let $r \asymp \sqrt{k}$ be a penalty parameter

$$F(x) := f(x) + r \left\| \mathcal{A}(x) - b \right\|_1 + 2 \operatorname{dist}_1(x, K)$$

- Compare: Lasso $\|Ax - b\|_2^2$ vs $\|Ax - b\|_1$



Related: Beck and Teboulle [2009], Tibshirani [1996]

Theorem (Structural)

F is μ-sharp in the ℓ_1 norm where μ is a function of “RIP constants of $\mathcal{A}$”

$$F(x) - F(x^\sharp) \geq \mu \left\|x - x^\sharp\right\|_1, \qquad \forall x \in V$$

and L-Lipschitz in the ℓ_1 norm with $L \asymp \sqrt{k}$

$$|F(x) - F(y)| \leq L \|x - y\|_1, \qquad \forall x, y.$$

Related: Candes and Tao [2005], Recht et al. [2010]

Theorem (Structural)

F is μ-sharp in the ℓ_1 norm where μ is a function of "RIP constants of $\mathcal{A}$"

$$F(x) - F(x^\sharp) \geq \mu \left\|x - x^\sharp\right\|_1, \qquad \forall x \in V$$

and L-Lipschitz in the ℓ_1 norm with $L \asymp \sqrt{k}$

$$|F(x) - F(y)| \leq L \left\|x - y\right\|_1, \qquad \forall x,\, y.$$

- μ increasing with "RIP constants of $\mathcal{A}$", in turn depends on sample size

Related: Candes and Tao [2005], Recht et al. [2010]

Theorem (Structural)

F is μ-sharp in the ℓ_1 norm where μ is a function of "RIP constants of $\mathcal{A}$"

$$F(x) - F(x^\sharp) \geq \mu \left\|x - x^\sharp\right\|_1, \qquad \forall x \in V$$

and L-Lipschitz in the ℓ_1 norm with $L \asymp \sqrt{k}$

$$|F(x) - F(y)| \leq L \left\|x - y\right\|_1, \qquad \forall x, y.$$

- μ increasing with "RIP constants of $\mathcal{A}$", in turn depends on sample size
- Sparse recovery: $\mu \asymp 1$ for $m \asymp k \log(n)$

Related: Candes and Tao [2005], Recht et al. [2010]

Theorem (Structural)

F is μ-sharp in the ℓ_1 norm where μ is a function of "RIP constants of $\mathcal{A}$"

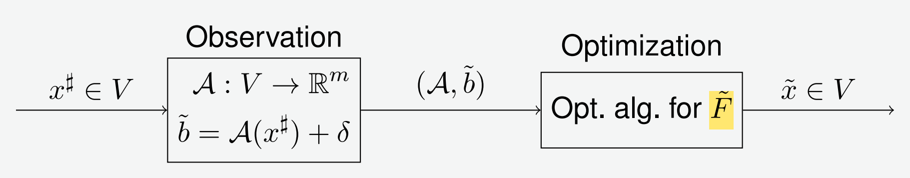
$$F(x) - F(x^\sharp) \geq \mu \left\|x - x^\sharp\right\|_1, \qquad \forall x \in V$$

and L-Lipschitz in the ℓ_1 norm with $L \asymp \sqrt{k}$

$$|F(x) - F(y)| \leq L \|x - y\|_1, \qquad \forall x, y.$$

- μ increasing with "RIP constants of $\mathcal{A}$", in turn depends on sample size
- Sparse recovery: $\mu \asymp 1$ for $m \asymp k \log(n)$
- Covariance estimation: $\mu \asymp 1$ for $m \asymp nk$

Related: Candes and Tao [2005], Recht et al. [2010]

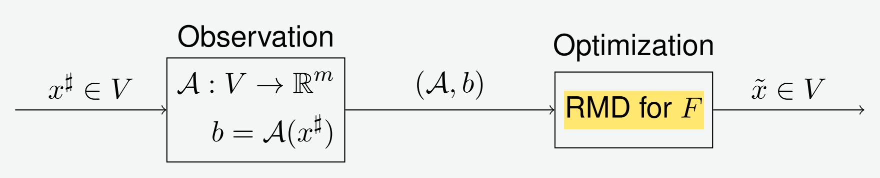


Corollary (Robustness)

Let $\tilde{x}$ be an ϵ minimizer of $\tilde{F}$.

- (to small noise) $\tilde{x}$ satisfies $\left\|\tilde{x} - x^\sharp\right\|_1 \lesssim \frac{\sqrt{k}}{\mu} \|\delta\|_1 + \frac{\epsilon}{\mu}$
- (to sparse noise) If $\frac{|\mathrm{supp}(\delta)|}{m} \lesssim 1/\sqrt{k}$, then $\left\|\tilde{x} - x^\sharp\right\|_1 \lesssim \frac{\epsilon}{\mu}$

Algorithms for minimizing F



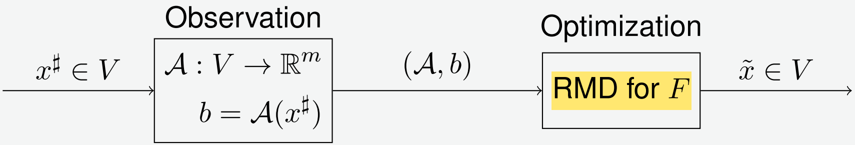
Corollary (Algorithms)

Restarted mirror descent (RMD) algorithm produces an ϵ-optimal solution to F in

$$O\left(\frac{k}{\mu^2} \log(n) \log(\epsilon^{-1})\right)$$

iterations of the mirror descent update.

Related: Polyak [1969], Roulet and d'Aspremont [2017], Yang and Lin [2018], Renegar and Grimmer [2022]



Corollary (Algorithms)

Restarted mirror descent (RMD) algorithm produces an ϵ-optimal solution to F in

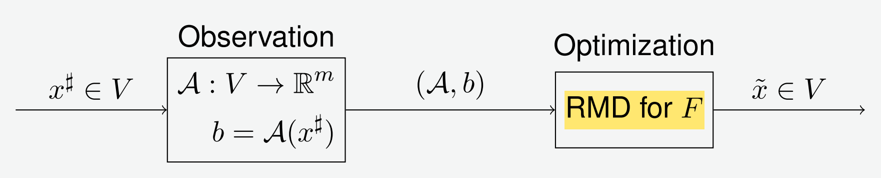
$$O\left(\frac{k}{\mu^2}\log(n)\log(\epsilon^{-1})\right)$$

iterations of the mirror descent update.

- Requires μ

Related: Polyak [1969], Roulet and d'Aspremont [2017], Yang and Lin [2018], Renegar and Grimmer [2022]

Algorithms for minimizing F



Corollary (Algorithms)

Restarted mirror descent (RMD) algorithm produces an ϵ-optimal solution to F in

$$O\left(\frac{k}{\mu^2}\log(n)\log(\epsilon^{-1})\right)$$

iterations of the mirror descent update.

- Requires μ
- If μ is not known, extra $\log(\epsilon^{-1})$ factor

Related: Polyak [1969], Roulet and d'Aspremont [2017], Yang and Lin [2018], Renegar and Grimmer [2022]

Algorithms for minimizing F

- Suppose we run MD from x_0 for t iterations with step size η and mirror map

$$h(x) \approx \frac{1}{2} \|x - x_0\|_1^2$$

Related: Polyak [1969], Roulet and d'Aspremont [2017], Yang and Lin [2018], Renegar and Grimmer [2022]

Algorithms for minimizing F

- Suppose we run MD from x_0 for t iterations with step size η and mirror map

$$h(x) \approx \frac{1}{2} \|x - x_0\|_1^2$$

- MD: output y

$$F(y) - F(x^\sharp) \leq \frac{L^2 \eta \ln n}{2} + \frac{D_h(x^\sharp || x_0)}{2\eta t}$$

Related: Polyak [1969], Roulet and d'Aspremont [2017], Yang and Lin [2018], Renegar and Grimmer [2022]

Algorithms for minimizing F

- Suppose we run MD from x_0 for t iterations with step size η and mirror map

$$h(x) \approx \frac{1}{2} \|x - x_0\|_1^2$$

- MD: output y

$$F(y) - F(x^\sharp) \leq \frac{L^2 \eta \ln n}{2} + \frac{D_h(x^\sharp || x_0)}{2\eta t} \approx \frac{L^2 \eta \ln n}{2} + \frac{\left\|x^\sharp - x_0\right\|_1^2}{4\eta t}$$

Related: Polyak [1969], Roulet and d'Aspremont [2017], Yang and Lin [2018], Renegar and Grimmer [2022]

Algorithms for minimizing F

- Suppose we run MD from x_0 for t iterations with step size η and mirror map

$$h(x) \approx \frac{1}{2} \|x - x_0\|_1^2$$

- MD: output y

$$F(y) - F(x^\sharp) \leq \frac{L^2 \eta \ln n}{2} + \frac{D_h(x^\sharp || x_0)}{2\eta t} \approx \frac{L^2 \eta \ln n}{2} + \frac{\left\|x^\sharp - x_0\right\|_1^2}{4\eta t}$$

$$= L \left\|x^\sharp - x_0\right\|_1 \sqrt{\frac{\ln n}{2t}}$$

Related: Polyak [1969], Roulet and d'Aspremont [2017], Yang and Lin [2018], Renegar and Grimmer [2022]

Algorithms for minimizing F

- Suppose we run MD from x_0 for t iterations with step size η and mirror map

$$h(x) \approx \frac{1}{2} \|x - x_0\|_1^2$$

- MD: output y

$$F(y) - F(x^\sharp) \leq \frac{L^2 \eta \ln n}{2} + \frac{D_h(x^\sharp \| x_0)}{2\eta t} \approx \frac{L^2 \eta \ln n}{2} + \frac{\left\|x^\sharp - x_0\right\|_1^2}{4\eta t}$$
$$= L \left\|x^\sharp - x_0\right\|_1 \sqrt{\frac{\ln n}{2t}}$$

- Applying sharpness $\longrightarrow$

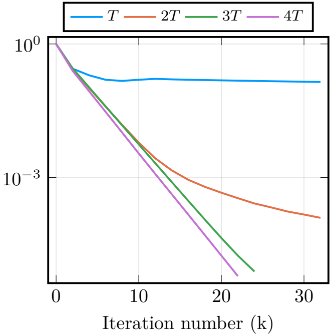
$$F(y) - F(x^\sharp) \leq \frac{1}{2} \left(F(x_0) - F(x^\sharp)\right)$$

after $\asymp \frac{L^2}{\mu^2} \ln n$ iterations

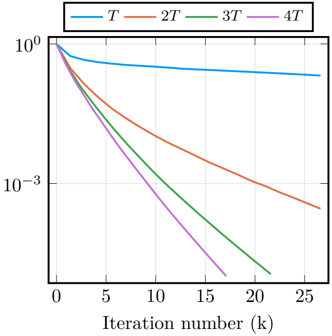
Related: Polyak [1969], Roulet and d'Aspremont [2017], Yang and Lin [2018], Renegar and Grimmer [2022]

Numerical experiments

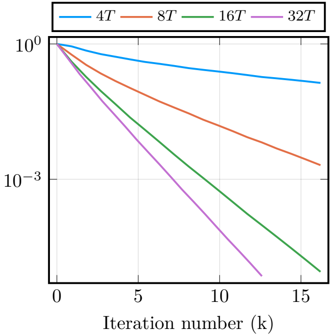
- Let T be statistical threshold for sparse recovery, low-rank matrix sensing (covariance estimation without PSD constraint), and phase retrieval (covariance estimation with $k = 1$)



sparse recovery
$(n, k) = (10^4, 5)$



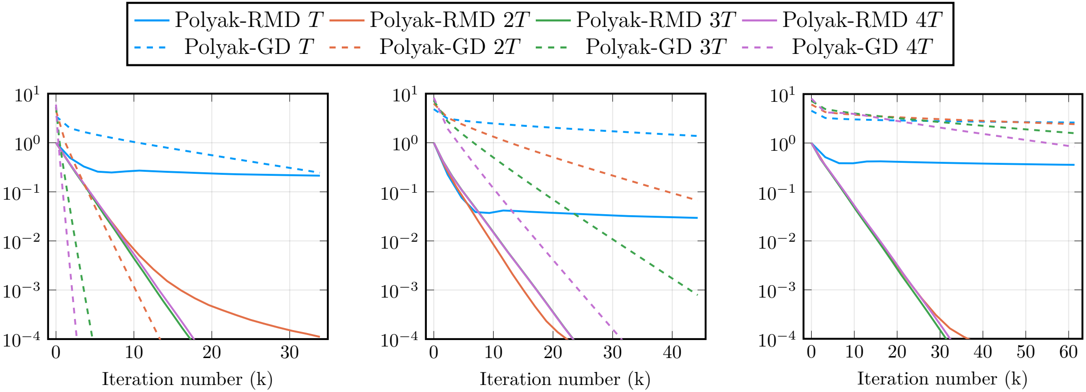
matrix sensing
$(n, k) = (100, 5)$



phase retrieval
$n = 100$

Restarted mirror descent vs. Polyak subgradient

- Polyak subgradient converges linearly on sharp Lipschitz functions in ℓ_2 norm



sparse recovery $(n, k) = (10^4, 5)$

sparse recovery $(n, k) = (10^5, 5)$

sparse recovery $(n, k) = (10^6, 5)$

Conclusion

- Abstract statistical signal recovery problem: sparse recovery, covariance estimation, matrix sensing, phase retrieval

Conclusion

- Abstract statistical signal recovery problem: sparse recovery, covariance estimation, matrix sensing, phase retrieval
- Contributions

Conclusion

- Abstract statistical signal recovery problem: sparse recovery, covariance estimation, matrix sensing, phase retrieval
- Contributions
  - **Structural**: ℓ_1 sharp and Lipschitz penalty formulation

Conclusion

- Abstract statistical signal recovery problem: sparse recovery, covariance estimation, matrix sensing, phase retrieval
- Contributions
  - **Structural**: ℓ_1 sharp and Lipschitz penalty formulation
  - **Robustness**: to observation error and optimization error

Conclusion

- Abstract statistical signal recovery problem: sparse recovery, covariance estimation, matrix sensing, phase retrieval
- Contributions
  - **Structural**: ℓ_1 sharp and Lipschitz penalty formulation
  - **Robustness**: to observation error and optimization error
  - **Algorithms**: Restarted Mirror Descent converges linearly

Conclusion

- Abstract statistical signal recovery problem: sparse recovery, covariance estimation, matrix sensing, phase retrieval
- Contributions
  - **Structural**: ℓ_1 sharp and Lipschitz penalty formulation
  - **Robustness**: to observation error and optimization error
  - **Algorithms**: Restarted Mirror Descent converges linearly

Questions?

Part 2

$O(1/T^{1.02449})$ Convergence of long-step gradient descent

Joint work with Benjamin Grimmer, Kevin Shu

Outline

- Preview of results (better guarantees for smooth convex minimization)

Outline

- Preview of results (better guarantees for smooth convex minimization)
- Why to expect this (history of prior works)

Outline

- Preview of results (better guarantees for smooth convex minimization)
- Why to expect this (history of prior works)
- Conceptual contributions

Outline

- Preview of results (better guarantees for smooth convex minimization)
- Why to expect this (history of prior works)
- Conceptual contributions
- Computer assisted design/proofs

Preview of results

Smooth convex optimization and gradient descent

- Want gradient descent-style algorithms for general convex functions f with

Smooth convex optimization and gradient descent

- Want gradient descent-style algorithms for general convex functions f with
  - f is 1-smooth

Smooth convex optimization and gradient descent

- Want gradient descent-style algorithms for general convex functions f with
  - f is 1-smooth
  - f has minimizer $x^\star$

Smooth convex optimization and gradient descent

- Want gradient descent-style algorithms for general convex functions f with
  - f is 1-smooth
  - f has minimizer $x^\star$
  - $\sup_{x\in\mathbb{R}^n} \{\|x - x^\star\| : f(x) \leq f(x_0)\} \leq 1$

Smooth convex optimization and gradient descent

- Want gradient descent-style algorithms for general convex functions f with
  - f is 1-smooth
  - f has minimizer $x^\star$
  - $\sup_{x\in\mathbb{R}^n} \{\|x - x^\star\| : f(x) \leq f(x_0)\} \leq 1$
- Gradient descent with steplength sequence $h = (h_0, h_1, \dots)$

$$x_1 = x_0 - h_0 \nabla f(x_0) \qquad x_2 = x_1 - h_1 \nabla f(x_1) \qquad \dots$$

$$x_{i+1} = x_i - h_i \nabla f(x_i)$$

Smooth convex optimization and gradient descent

- Want gradient descent-style algorithms for general convex functions f with
  - f is 1-smooth
  - f has minimizer $x^\star$
  - $\sup_{x\in\mathbb{R}^n} \{\|x - x^\star\| : f(x) \leq f(x_0)\} \leq 1$
- Gradient descent with steplength sequence $h = (h_0, h_1, \dots)$

$$x_1 = x_0 - h_0 \nabla f(x_0) \qquad x_2 = x_1 - h_1 \nabla f(x_1) \qquad \dots$$

$$x_{i+1} = x_i - h_i \nabla f(x_i)$$

- **Goal**: pick steplength sequence $(h_0, h_1, \dots)$ to maximize convergence rate

What we knew prior to 2021

- $f(x_{i+1}) < f(x_i)$ is guaranteed if and only if $h_i \in (0, 2)$

What we knew prior to 2021

- $f(x_{i+1}) < f(x_i)$ is guaranteed if and only if $h_i \in (0, 2)$
- Per-iteration guaranteed worst-case descent maximized by $h_i = 1$

What we knew prior to 2021

- $f(x_{i+1}) < f(x_i)$ is guaranteed if and only if $h_i \in (0, 2)$
- Per-iteration guaranteed worst-case descent maximized by $h_i = 1$
- For $h = (1, 1, 1, \dots)$,

$$f(x_T) - f(x^\star) \leq \frac{1}{2T}$$

What we knew prior to 2021

- $f(x_{i+1}) < f(x_i)$ is guaranteed if and only if $h_i \in (0, 2)$
- Per-iteration guaranteed worst-case descent maximized by $h_i = 1$
- For $h = (1, 1, 1, \dots)$,

$$f(x_T) - f(x^\star) \leq \frac{1}{2T}$$

Today: a per-iteration analysis is too short-sighted

What we knew prior to 2021

- $f(x_{i+1}) < f(x_i)$ is guaranteed if and only if $h_i \in (0, 2)$
- Per-iteration guaranteed worst-case descent maximized by $h_i = 1$
- For $h = (1, 1, 1, \dots)$,

$$f(x_T) - f(x^\star) \le \frac{1}{2T}$$

  Today: a per-iteration analysis is too short-sighted
- Optimal rates for first-order methods: Accelerated gradient descent

$$f(x_T) - f(x^\star) \le \frac{2}{T^2}$$

  Note: this is *not* a gradient descent-style algorithm

Taking larger steps: breaking some intuitions

- Consider $h = 0.99 \times \left(\boxed{\frac{3}{2}, 5, \frac{3}{2}}, \boxed{\frac{3}{2}, 5, \frac{3}{2}}, \ldots\right)$

Taking larger steps: breaking some intuitions

- Consider $h = 0.99 \times \left(\boxed{\frac{3}{2}, 5, \frac{3}{2}}, \boxed{\frac{3}{2}, 5, \frac{3}{2}}, \ldots \right)$
- We can guarantee

$$f(x_T) - f(x^\star) \leq \frac{1}{2.66 \cdot T} + O\left(\frac{1}{T^2}\right) \qquad \text{for all } T \equiv 0 \bmod 3$$

Taking larger steps: breaking some intuitions

- Consider $h = 0.99 \times \left(\boxed{\frac{3}{2}, 5, \frac{3}{2}}, \boxed{\frac{3}{2}, 5, \frac{3}{2}}, \ldots \right)$
- We can guarantee

$$f(x_T) - f(x^\star) \leq \frac{1}{2.66 \cdot T} + O\left(\frac{1}{T^2}\right) \qquad \text{for all } T \equiv 0 \bmod 3$$

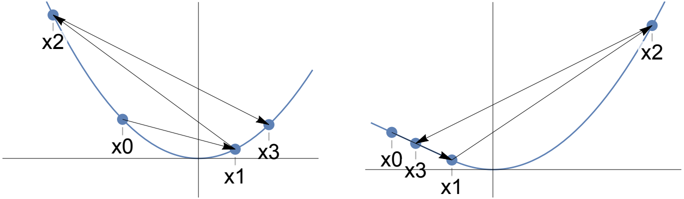
- This is *faster* even though we cannot guarantee per-iteration descent!

Taking larger steps: breaking some intuitions

- Consider $h = 0.99 \times \left(\boxed{\frac{3}{2}, 5, \frac{3}{2}}, \boxed{\frac{3}{2}, 5, \frac{3}{2}}, \dots \right)$
- We can guarantee

$$f(x_T) - f(x^\star) \leq \frac{1}{2.66 \cdot T} + O\left(\frac{1}{T^2}\right) \qquad \text{for all } T \equiv 0 \bmod 3$$

- This is *faster* even though we cannot guarantee per-iteration descent!

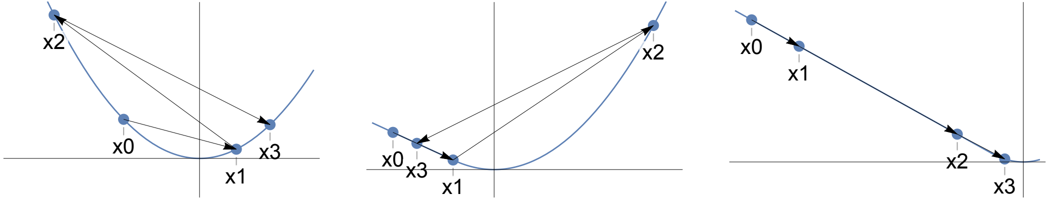


Taking larger steps: breaking some intuitions

- Consider $h = 0.99 \times \left(\boxed{\frac{3}{2}, 5, \frac{3}{2}}, \boxed{\frac{3}{2}, 5, \frac{3}{2}}, \ldots\right)$
- We can guarantee

$$f(x_T) - f(x^\star) \leq \frac{1}{2.66 \cdot T} + O\left(\frac{1}{T^2}\right) \qquad \text{for all } T \equiv 0 \bmod 3$$

- This is *faster* even though we cannot guarantee per-iteration descent!



The full sequence

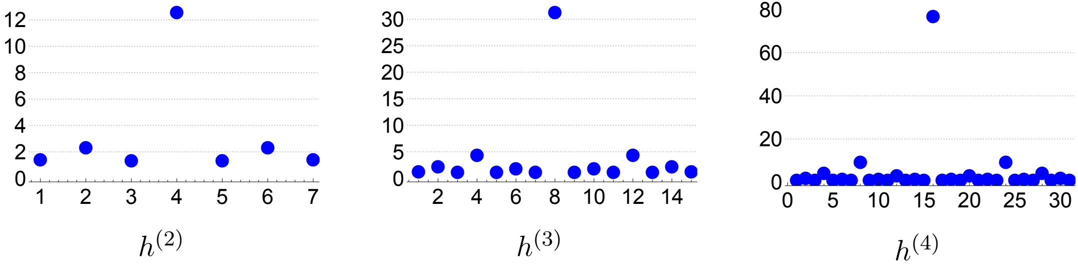
- We construct steplength blocks $h^{(k)} \in \mathbb{R}^{2^{k+1}-1}$ that can be scaled down to guarantee descent

The full sequence

- We construct steplength blocks $h^{(k)} \in \mathbb{R}^{2^{k+1}-1}$ that can be scaled down to guarantee descent
- $h^{(0)} = (1), \quad h^{(1)} = \left(\frac{3}{2}, 5, \frac{3}{2}\right), \quad h^{(2)} = \left(\frac{3}{2}, 1+\sqrt{2}, \sqrt{2}, 7+4\sqrt{2}, \sqrt{2}, 1+\sqrt{2}, \frac{3}{2}\right)$

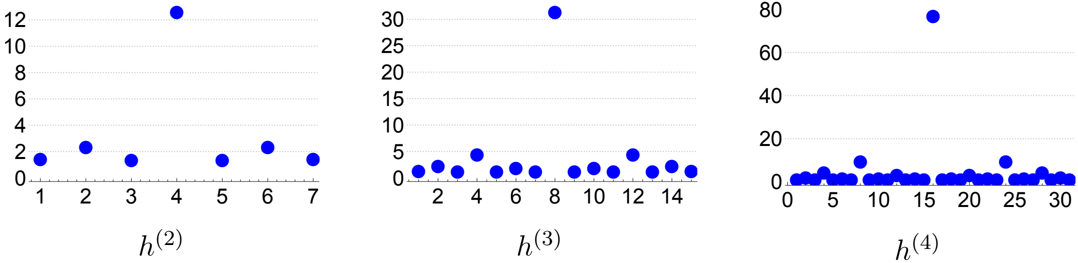
The full sequence

- We construct steplength blocks $h^{(k)} \in \mathbb{R}^{2^{k+1}-1}$ that can be scaled down to guarantee descent
- $h^{(0)} = (1), \quad h^{(1)} = \left(\frac{3}{2}, 5, \frac{3}{2}\right), \quad h^{(2)} = \left(\frac{3}{2}, 1+\sqrt{2}, \sqrt{2}, 7+4\sqrt{2}, \sqrt{2}, 1+\sqrt{2}, \frac{3}{2}\right)$



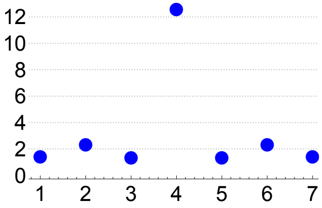
The full sequence

- We construct steplength blocks $h^{(k)} \in \mathbb{R}^{2^{k+1}-1}$ that can be scaled down to guarantee descent
- $h^{(0)} = (1), \quad h^{(1)} = \left(\frac{3}{2}, 5, \frac{3}{2}\right), \quad h^{(2)} = \left(\frac{3}{2}, 1+\sqrt{2}, \sqrt{2}, 7+4\sqrt{2}, \sqrt{2}, 1+\sqrt{2}, \frac{3}{2}\right)$

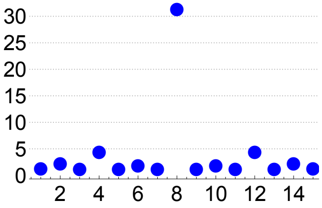


- Longer patterns have increasingly fast convergence rates

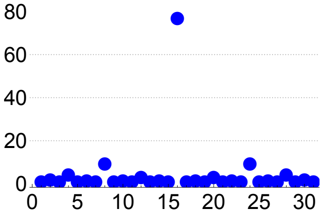
- We construct steplength blocks $h^{(k)} \in \mathbb{R}^{2^{k+1}-1}$ that can be scaled down to guarantee descent
- $h^{(0)} = (1), \quad h^{(1)} = (\frac{3}{2}, 5, \frac{3}{2}), \quad h^{(2)} = (\frac{3}{2}, 1+\sqrt{2}, \sqrt{2}, 7+4\sqrt{2}, \sqrt{2}, 1+\sqrt{2}, \frac{3}{2})$



$h^{(2)}$



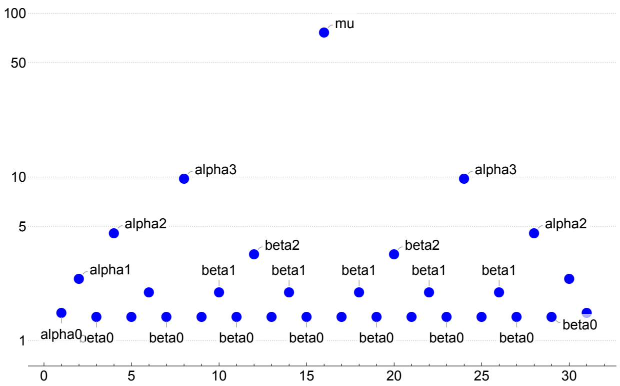
$h^{(3)}$



$h^{(4)}$

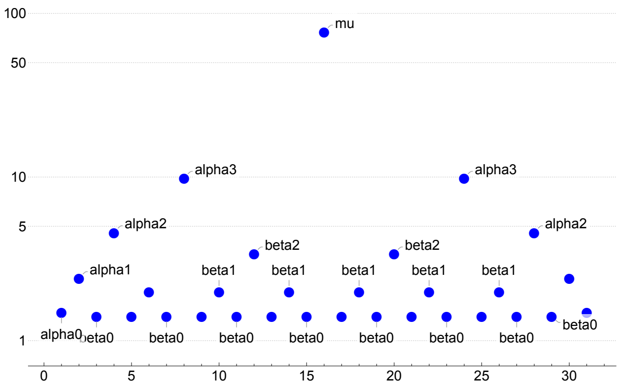
- Longer patterns have increasingly fast convergence rates
- $\mathrm{avg}(h^{(k)})$ is exponential in k

A closer look at $h^{(k)}$



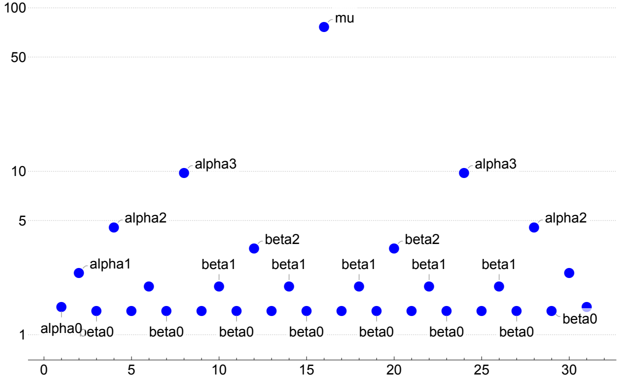
- $\beta_i = 1 + (1+\sqrt{2})^{i-1} \longrightarrow (1+\sqrt{2})$ is the silver ratio and dictates our rate

A closer look at $h^{(k)}$



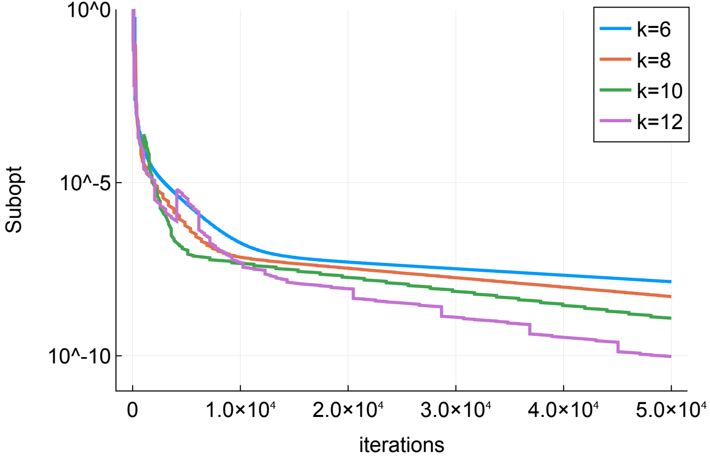
- $\beta_i = 1 + (1+\sqrt{2})^{i-1} \longrightarrow (1+\sqrt{2})$ is the silver ratio and dictates our rate
- μ is sum of all other stepsizes plus two

A closer look at $h^{(k)}$



- $\beta_i = 1 + (1+\sqrt{2})^{i-1} \longrightarrow (1+\sqrt{2})$ is the silver ratio and dictates our rate
- μ is sum of all other stepsizes plus two
- α_i picked so that $\prod_{\text{stepsizes}}(\text{stepsize} - 1) = 1$

Numerical comparison of $h^{(k)}$



- $h^{(12)}$ has length 8191

Theorem

Suppose

$$h = \frac{1}{2}\left(\boxed{h^{(0)}, \ldots, h^{(0)}}, \boxed{h^{(1)}, \ldots, h^{(1)}}, \ldots, \boxed{h^{(k)}, \ldots, h^{(k)}}, \ldots\right)$$

where each $h^{(k)}$ is repeated $\approx c^k$ times. Then

$$\left(\min_{t \le T} f(x_t)\right) - f(x^\star) = O\left(\frac{1}{T^{1.02449}}\right)$$

Why should we expect this?

AKA some recent work in the area

The Performance Estimation Problem (PEP) 1/2

- **Question**: Suppose we have a candidate $h = (h_0, h_1, \dots, h_{T-1})$. What is the worst case function? (Smoothness 1, initial distance 1, initial suboptimality δ)

- **Question**: Suppose we have a candidate $h = (h_0, h_1, \ldots, h_{T-1})$. What is the worst case function? (Smoothness 1, initial distance 1, initial suboptimality δ)

$$p_h(\delta) \coloneqq \max_{x_0, x^\star, f} \left\{ f(x_T) - f(x^\star) : \begin{array}{l} f \text{ is convex, } 1\text{-smooth} \\ \|x_0 - x^\star\|^2 \leq 1 \\ f(x_0) - f(x^\star) \leq \delta \\ \nabla f(x^\star) = 0 \\ x_{i+1} = x_i - h_i \nabla f(x_i) \end{array} \right\}$$

- **Question**: Suppose we have a candidate $h = (h_0, h_1, \dots, h_{T-1})$. What is the worst case function? (Smoothness 1, initial distance 1, initial suboptimality δ)

$$p_h(\delta) := \max_{x_0, x^\star, f} \left\{ f(x_T) - f(x^\star) : \begin{array}{l} f \text{ is convex, } 1\text{-smooth} \\ \|x_0 - x^\star\|^2 \leq 1 \\ f(x_0) - f(x^\star) \leq \delta \\ \nabla f(x^\star) = 0 \\ x_{i+1} = x_i - h_i \nabla f(x_i) \end{array} \right\}$$

- Iterates x_i only depend on gradients

- **Question**: Suppose we have a candidate $h = (h_0, h_1, \ldots, h_{T-1})$. What is the worst case function? (Smoothness 1, initial distance 1, initial suboptimality δ)

$$p_h(\delta) := \max_{x_0, x^\star, f} \left\{ f(x_T) - f(x^\star) : \begin{array}{l} f \text{ is convex, } 1\text{-smooth} \\ \|x_0 - x^\star\|^2 \leq 1 \\ f(x_0) - f(x^\star) \leq \delta \\ \nabla f(x^\star) = 0 \\ x_{i+1} = x_i - h_i \nabla f(x_i) \end{array} \right\}$$

- Iterates x_i only depend on gradients $\longrightarrow$ Optimize over gradients g_i and function values f_i for which there exists an 1-smooth, convex interpolating f

- **Question**: Suppose we have a candidate $h = (h_0, h_1, \ldots, h_{T-1})$. What is the worst case function? (Smoothness 1, initial distance 1, initial suboptimality δ)

$$p_h(\delta) := \max_{x_0, x^\star, f} \left\{ f(x_T) - f(x^\star) : \begin{array}{l} f \text{ is convex, } 1\text{-smooth} \\ \|x_0 - x^\star\|^2 \le 1 \\ f(x_0) - f(x^\star) \le \delta \\ \nabla f(x^\star) = 0 \\ x_{i+1} = x_i - h_i \nabla f(x_i) \end{array} \right\}$$

- Iterates x_i only depend on gradients $\longrightarrow$ Optimize over gradients g_i and function values f_i for which there exists an 1-smooth, convex interpolating f
  - Drori and Teboulle [2012], Taylor et al. [2017] give necessary and sufficient conditions for such a function to exist . . . nonconvex quadratic program

- **Question**: Suppose we have a candidate $h = (h_0, h_1, \ldots, h_{T-1})$. What is the worst case function? (Smoothness 1, initial distance 1, initial suboptimality δ)

$$p_h(\delta) := \max_{x_0, x^\star, f} \left\{ f(x_T) - f(x^\star) : \begin{array}{l} f \text{ is convex, } 1\text{-smooth} \\ \|x_0 - x^\star\|^2 \leq 1 \\ f(x_0) - f(x^\star) \leq \delta \\ \nabla f(x^\star) = 0 \\ x_{i+1} = x_i - h_i \nabla f(x_i) \end{array} \right\}$$

- Iterates x_i only depend on gradients $\longrightarrow$ Optimize over gradients g_i and function values f_i for which there exists an 1-smooth, convex interpolating f
  - Drori and Teboulle [2012], Taylor et al. [2017] give necessary and sufficient conditions for such a function to exist . . . nonconvex quadratic program
  - The SDP relaxation of this nonconvex quadratic program is exact!

The Performance Estimation Problem (PEP) 2/2

- $p_h(\delta)$

- $p_h(\delta) =$ maximum value of nonconvex infinite dimensional problem

The Performance Estimation Problem (PEP) 2/2

- $$\begin{aligned} p_h(\delta) &= \text{maximum value of nonconvex infinite dimensional problem} \\ &= \text{maximum value of a nonconvex quadratic program} \end{aligned}$$

- $$\begin{aligned} p_h(\delta) &= \text{maximum value of nonconvex infinite dimensional problem} \\ &= \text{maximum value of a nonconvex quadratic program} \\ &= \text{maximum value of an SDP} \end{aligned}$$

The Performance Estimation Problem (PEP) 2/2

- $$\begin{aligned} p_h(\delta) &= \text{maximum value of nonconvex infinite dimensional problem} \\ &= \text{maximum value of a nonconvex quadratic program} \\ &= \text{maximum value of an SDP} \\ &= \text{minimum value of the dual SDP} \end{aligned}$$

The Performance Estimation Problem (PEP) 2/2

- $$\begin{aligned} p_h(\delta) &= \text{maximum value of nonconvex infinite dimensional problem} \\ &= \text{maximum value of a nonconvex quadratic program} \\ &= \text{maximum value of an SDP} \\ &= \text{minimum value of the dual SDP} \end{aligned}$$
- **Take-aways**:

- $$\begin{aligned} p_h(\delta) &= \text{maximum value of nonconvex infinite dimensional problem} \\ &= \text{maximum value of a nonconvex quadratic program} \\ &= \text{maximum value of an SDP} \\ &= \text{minimum value of the dual SDP} \end{aligned}$$
- **Take-aways**:
  - $p_h(\delta)$ can be computed “efficiently” (for T small)

The Performance Estimation Problem (PEP) 2/2

- $$\begin{aligned} p_h(\delta) &= \text{maximum value of nonconvex infinite dimensional problem} \\ &= \text{maximum value of a nonconvex quadratic program} \\ &= \text{maximum value of an SDP} \\ &= \text{minimum value of the dual SDP} \end{aligned}$$
- **Take-aways**:
  - $p_h(\delta)$ can be computed “efficiently” (for T small)
  - Any feasible solution to the dual SDP gives an upper bound on worst-case performance!

- $$\begin{aligned} p_h(\delta) &= \text{maximum value of nonconvex infinite dimensional problem} \\ &= \text{maximum value of a nonconvex quadratic program} \\ &= \text{maximum value of an SDP} \\ &= \text{minimum value of the dual SDP} \end{aligned}$$
- **Take-aways**:
  - $p_h(\delta)$ can be computed "efficiently" (for T small)
  - Any feasible solution to the dual SDP gives an upper bound on worst-case performance!
- Now, how to design h? $\min_{h=(h_0,\ldots,h_{T-1})} p_h(\delta)$

Gradient descent with long steps

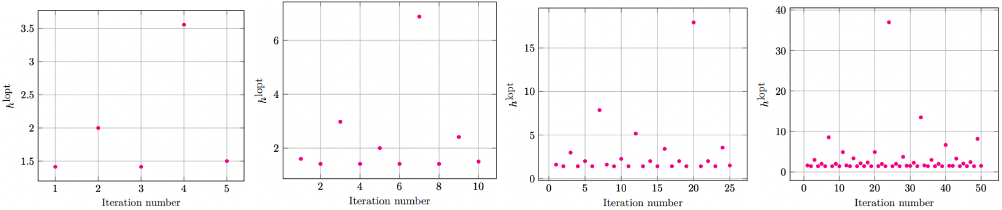
- Das Gupta et al. [2023]: Complex branch-and-bound scheme for $T \in [1, \ldots, 50]$

$$\min_{h=(h_0,\ldots,h_{T-1})} p_h(1/2)$$

Gradient descent with long steps

- Das Gupta et al. [2023]: Complex branch-and-bound scheme for $T \in [1, \ldots, 50]$

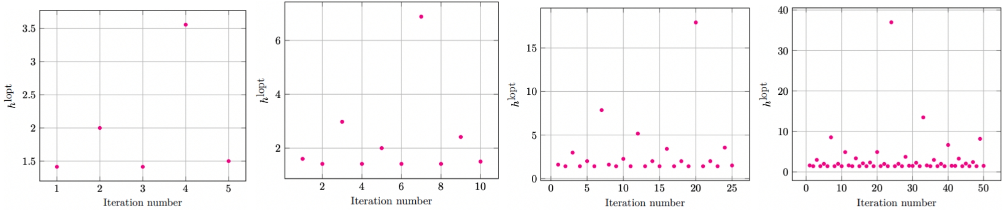
$$\min_{h=(h_0,\ldots,h_{T-1})} p_h(1/2)$$



(a) $N = 5$ **(b)** $N = 10$ **(c)** $N = 25$ **(d)** $N = 50$

- Das Gupta et al. [2023]: Complex branch-and-bound scheme for $T \in [1, \dots, 50]$

$$\min_{h=(h_0,\dots,h_{T-1})} p_h(1/2)$$



(a) $N = 5$ **(b)** $N = 10$ **(c)** $N = 25$ **(d)** $N = 50$

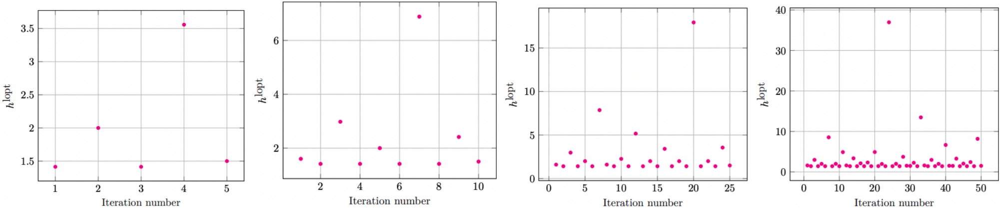
- Strongest guarantee

$$f(x_{50}) \leq 0.002 \approx \text{factor of 5 faster than } \frac{1}{2T}$$

Gradient descent with long steps

- Das Gupta et al. [2023]: Complex branch-and-bound scheme for $T \in [1, \ldots, 50]$

$$\min_{h=(h_0,\ldots,h_{T-1})} p_h(1/2)$$



(a) $N = 5$ **(b)** $N = 10$ **(c)** $N = 25$ **(d)** $N = 50$

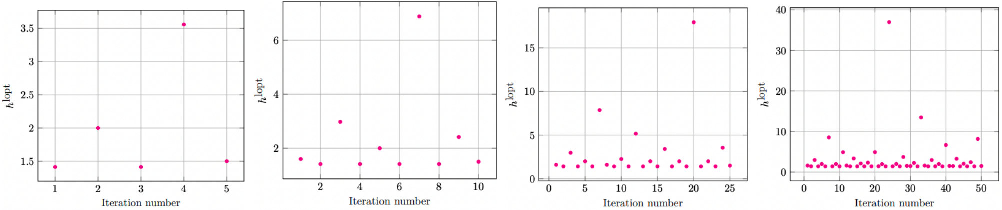
- Strongest guarantee

$$f(x_{50}) \leq 0.002 \approx \text{factor of 5 faster than } \frac{1}{2T}$$

- To get actual convergence rates, need a suitable induction

- Das Gupta et al. [2023]: Complex branch-and-bound scheme for $T \in [1, \ldots, 50]$

$$\min_{h=(h_0,\ldots,h_{T-1})} p_h(1/2)$$



(a) $N = 5$ **(b)** $N = 10$ **(c)** $N = 25$ **(d)** $N = 50$

- Strongest guarantee

$$f(x_{50}) \leq 0.002 \approx \text{factor of 5 faster than } \frac{1}{2T}$$

- To get actual convergence rates, need a suitable induction
- Our work: analytically construct solution for all δ, with $p_h(\delta)$ small

Long-step gradient descent in other contexts

- Young [1953], Lebedev and Finogenov [1971], Agarwal et al. [2021]: Long-step gradient descent for smooth strongly convex *quadratic functions*

Long-step gradient descent in other contexts

- Young [1953], Lebedev and Finogenov [1971], Agarwal et al. [2021]: Long-step gradient descent for smooth strongly convex *quadratic functions*
  - Can achieve full acceleration $O(\sqrt{\kappa}\log(\epsilon^{-1}))$

Long-step gradient descent in other contexts

- Young [1953], Lebedev and Finogenov [1971], Agarwal et al. [2021]: Long-step gradient descent for smooth strongly convex *quadratic functions*
  - Can achieve full acceleration $O(\sqrt{\kappa}\log(\epsilon^{-1}))$
- Oymak [2021]: Smooth strongly convex minimization with bimodal Hessians

Long-step gradient descent in other contexts

- Young [1953], Lebedev and Finogenov [1971], Agarwal et al. [2021]: Long-step gradient descent for smooth strongly convex *quadratic functions*
    - Can achieve full acceleration $O(\sqrt{\kappa}\log(\epsilon^{-1}))$
- Oymak [2021]: Smooth strongly convex minimization with bimodal Hessians
- Altschuler [2018]: Smooth strongly convex minimization (solved PEP for $T = 1, 2, 3$)

Long-step gradient descent in other contexts

- Young [1953], Lebedev and Finogenov [1971], Agarwal et al. [2021]: Long-step gradient descent for smooth strongly convex *quadratic functions*
  - Can achieve full acceleration $O(\sqrt{\kappa}\log(\epsilon^{-1}))$
- Oymak [2021]: Smooth strongly convex minimization with bimodal Hessians
- Altschuler [2018]: Smooth strongly convex minimization (solved PEP for $T = 1, 2, 3$)
  $\longrightarrow$ Altschuler, Parillo [yesterday] $O(1/T^{1.27})$ for smooth convex minimization

Long-step gradient descent in other contexts

- Young [1953], Lebedev and Finogenov [1971], Agarwal et al. [2021]: Long-step gradient descent for smooth strongly convex *quadratic functions*
  - Can achieve full acceleration $O(\sqrt{\kappa}\log(\epsilon^{-1}))$
- Oymak [2021]: Smooth strongly convex minimization with bimodal Hessians
- Altschuler [2018]: Smooth strongly convex minimization (solved PEP for $T = 1, 2, 3$)
  $\longrightarrow$ Altschuler, Parillo [yesterday] $O(1/T^{1.27})$ for smooth convex minimization
- Loshchilov and Hutter [2016], Smith [2015], Smith and Topin [2017]: Nonconvex, smooth minimization in neural networks

Conceptual contributions

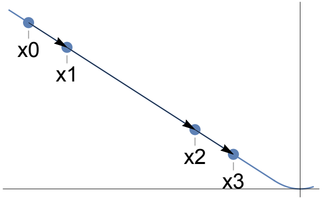
A suitable induction

- **Intuition**: Long flat regions of small slope is the worst case

- **Intuition**: Long flat regions of small slope is the worst case
- Suppose $\delta > 0$ small and consider

$$f(x) = \begin{cases} -\delta x - \delta^2/2 & \text{if } x \leq -\delta \\ \frac{1}{2}x^2 & \text{if } x \geq -\delta \end{cases}$$

$$x_0 = -(1 + \delta/2)$$

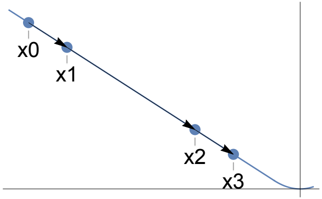


A suitable induction

- **Intuition**: Long flat regions of small slope is the worst case
- Suppose $\delta > 0$ small and consider

$$f(x) = \begin{cases} -\delta x - \delta^2/2 & \text{if } x \leq -\delta \\ \frac{1}{2}x^2 & \text{if } x \geq -\delta \end{cases}$$

$$x_0 = -(1 + \delta/2)$$



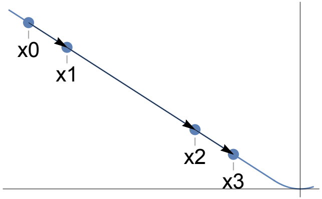
- Then, $f(x_0) = \delta$ and $f(x_T) = \delta - \delta^2 \sum h_i$

A suitable induction

- **Intuition**: Long flat regions of small slope is the worst case
- Suppose $\delta > 0$ small and consider

$$f(x) = \begin{cases} -\delta x - \delta^2/2 & \text{if } x \leq -\delta \\ \frac{1}{2}x^2 & \text{if } x \geq -\delta \end{cases}$$

$$x_0 = -(1 + \delta/2)$$



- Then, $f(x_0) = \delta$ and $f(x_T) = \delta - \delta^2 \sum h_i$
- Thus, for $\delta > 0$ small, $p_h(\delta) \geq \delta - \delta^2 \sum h_i$

Straightforward blocks

- We say a steplength *block* $h = (h_0, h_1, \ldots, h_{T-1})$ is Δ-straightforward if

$$p_h(\delta) \leq \delta - \delta^2 \sum h_i \qquad \forall \delta \in [0, \Delta]$$

Straightforward blocks

- We say a steplength *block* $h = (h_0, h_1, \ldots, h_{T-1})$ is Δ-straightforward if

$$p_h(\delta) \leq \delta - \delta^2 \sum h_i \qquad \forall \delta \in [0, \Delta]$$

- Solving a recurrence gives

$$\min_{s \leq T} f(x_s) - f(x^\star) \leq \frac{1}{\text{avg}(h)T} + O\left(\frac{1}{T^2}\right)$$

Straightforward blocks

- We say a steplength *block* $h = (h_0, h_1, \ldots, h_{T-1})$ is Δ-straightforward if

$$p_h(\delta) \leq \delta - \delta^2 \sum h_i \qquad \forall \delta \in [0, \Delta]$$

- Solving a recurrence gives

$$\min_{s \leq T} f(x_s) - f(x^\star) \leq \frac{1}{\operatorname{avg}(h) T} + O\left(\frac{1}{T^2}\right)$$

- **New goal**: maximize $\operatorname{avg}(h)$ over > 0-straightforward steplength blocks

Straightforward blocks

- We say a steplength *block* $h = (h_0, h_1, \ldots, h_{T-1})$ is Δ-straightforward if

$$p_h(\delta) \leq \delta - \delta^2 \sum h_i \qquad \forall \delta \in [0, \Delta]$$

- Solving a recurrence gives

$$\min_{s \leq T} f(x_s) - f(x^\star) \leq \frac{1}{\text{avg}(h)T} + O\left(\frac{1}{T^2}\right)$$

- **New goal**: maximize $\text{avg}(h)$ over > 0-straightforward steplength blocks
- We show that $\text{avg}(h^{(k)})$ is exponentially large in k, $\Delta^{(k)}$ is $\geq$ exponentially small $\longrightarrow$ accelerated convergence rates

Certifying straightforwardness

- Recall PEP: $p_h(\delta) =$ minimum value of SDP

Certifying straightforwardness

- Recall PEP: $p_h(\delta) =$ minimum value of SDP
- Then, h is Δ-straightforward if the following set is nonempty for all $\delta \in [0, \Delta]$

$$\mathcal{R}_{h,\delta} := \left\{ \lambda \in \mathbb{R}^{(t+2)\times(t+2)} : \begin{array}{l} \sum_{i\neq j} \lambda_{i,j} a_{i,j} = a_{\star,t} - (1 - 2\delta \sum_i h_i) a_{\star,0} \\ \lambda \geq 0 \\ Z_{h,\delta}(\lambda) \succeq 0 \end{array} \right\}$$

Certifying straightforwardness

- Recall PEP: $p_h(\delta) =$ minimum value of SDP
- Then, h is Δ-straightforward if the following set is nonempty for all $\delta \in [0, \Delta]$

$$\mathcal{R}_{h,\delta} := \left\{ \lambda \in \mathbb{R}^{(t+2)\times(t+2)} : \begin{array}{l} \sum_{i\neq j} \lambda_{i,j} a_{i,j} = a_{\star,t} - (1 - 2\delta \sum_i h_i) a_{\star,0} \\ \lambda \geq 0 \\ Z_{h,\delta}(\lambda) \succeq 0 \end{array} \right\}$$

- To get around solving one SDP for each $\delta \in [0, \Delta]$, we parameterize

$$\lambda(\delta) = \lambda_0 + \delta\gamma$$

Certifying straightforwardness

- Recall PEP: $p_h(\delta) =$ minimum value of SDP
- Then, h is Δ-straightforward if the following set is nonempty for all $\delta \in [0, \Delta]$

$$\mathcal{R}_{h,\delta} := \left\{ \lambda \in \mathbb{R}^{(t+2)\times(t+2)} : \begin{array}{l} \sum_{i\neq j} \lambda_{i,j} a_{i,j} = a_{\star,t} - (1 - 2\delta \sum_i h_i) a_{\star,0} \\ \lambda \geq 0 \\ Z_{h,\delta}(\lambda) \succeq 0 \end{array} \right\}$$

- To get around solving one SDP for each $\delta \in [0, \Delta]$, we parameterize

$$\lambda(\delta) = \lambda_0 + \delta\gamma$$

- This becomes a nonlinear SDP but can be "reformulated" into a regular SDP if we consider "limiting behavior as $\Delta \to 0$", at which point we can attempt to certify Δ-straightforwardness computationally

Computer assisted design/proofs

A few words on how we designed our stepsizes

- Mostly a guessing game:

A few words on how we designed our stepsizes

- Mostly a guessing game:
  numerically solve some instances $\longrightarrow$ conjecture patterns $\longrightarrow$ solve larger instances $\longrightarrow$ repeat

A few words on how we designed our stepsizes

- Mostly a guessing game:
  numerically solve some instances $\longrightarrow$ conjecture patterns $\longrightarrow$ solve larger instances $\longrightarrow$ repeat
- Exhaustive search for $T = 2, 3, 4, 5$ (three computers running for about a week)

A few words on how we designed our stepsizes

- Mostly a guessing game:
  numerically solve some instances $\longrightarrow$ conjecture patterns $\longrightarrow$ solve larger instances $\longrightarrow$ repeat
- Exhaustive search for $T = 2, 3, 4, 5$ (three computers running for about a week)
- Spot semidefinite term always rank-one and nonnegative at optimal solution $\longrightarrow T = 7, 15$

A few words on how we designed our stepsizes

- Mostly a guessing game:
  numerically solve some instances $\longrightarrow$ conjecture patterns $\longrightarrow$ solve larger instances $\longrightarrow$ repeat
- Exhaustive search for $T = 2, 3, 4, 5$ (three computers running for about a week)
- Spot semidefinite term always rank-one and nonnegative at optimal solution $\longrightarrow T = 7, 15$
- Spot optimal sparsity pattern for λ_0 and dependence of γ on $\lambda_0 \longrightarrow T = 31, 63$

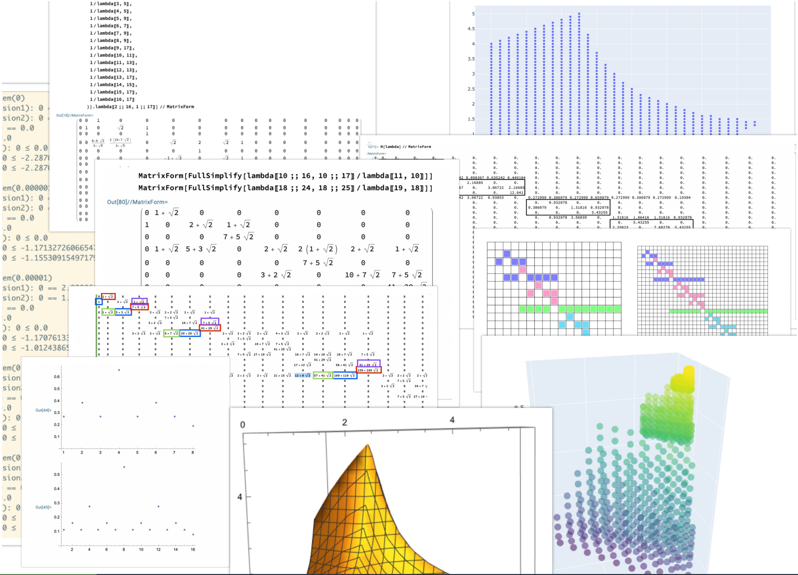
A few words on how we designed our stepsizes

- Mostly a guessing game:
  numerically solve some instances $\longrightarrow$ conjecture patterns $\longrightarrow$ solve larger instances $\longrightarrow$ repeat
- Exhaustive search for $T = 2, 3, 4, 5$ (three computers running for about a week)
- Spot semidefinite term always rank-one and nonnegative at optimal solution $\longrightarrow T = 7, 15$
- Spot optimal sparsity pattern for λ_0 and dependence of γ on λ_0 $\longrightarrow T = 31, 63$
- Spot recurrence relation in step lengths in optimal h $\longrightarrow T = 127, 255$

A few words on how we designed our stepsizes

- Mostly a guessing game:
  numerically solve some instances $\longrightarrow$ conjecture patterns $\longrightarrow$ solve larger instances $\longrightarrow$ repeat
- Exhaustive search for $T = 2, 3, 4, 5$ (three computers running for about a week)
- Spot semidefinite term always rank-one and nonnegative at optimal solution $\longrightarrow T = 7, 15$
- Spot optimal sparsity pattern for λ_0 and dependence of γ on $\lambda_0 \longrightarrow T = 31, 63$
- Spot recurrence relation in step lengths in optimal $h \longrightarrow T = 127, 255$
- Spot patterns in $\lambda_0 \longrightarrow T = 511$

Pictures of process



Conclusion

- First accelerated convergence guarantee for gradient descent using long steps

$$\min_{s \leq T} f(x_s) - f(x^\star) = O\left(\frac{1}{T^{1.02449}}\right)$$

Conclusion

- First accelerated convergence guarantee for gradient descent using long steps

$$\min_{s \le T} f(x_s) - f(x^\star) = O\left(\frac{1}{T^{1.02449}}\right)$$

- How did we do this?

Conclusion

- First accelerated convergence guarantee for gradient descent using long steps

$$\min_{s \leq T} f(x_s) - f(x^\star) = O\left(\frac{1}{T^{1.02449}}\right)$$

- How did we do this?
  - Straightforward patterns: where flat regions are worst case

Conclusion

- First accelerated convergence guarantee for gradient descent using long steps

$$\min_{s \leq T} f(x_s) - f(x^\star) = O\left(\frac{1}{T^{1.02449}}\right)$$

- How did we do this?
    - Straightforward patterns: where flat regions are worst case
    - Can certify straightforwardness by (analytically) solving SDPs

Conclusion

- First accelerated convergence guarantee for gradient descent using long steps

$$\min_{s \leq T} f(x_s) - f(x^\star) = O\left(\frac{1}{T^{1.02449}}\right)$$

- How did we do this?
  - Straightforward patterns: where flat regions are worst case
  - Can certify straightforwardness by (analytically) solving SDPs
  - Alternating between pattern spotting and solving larger and larger SDPs $\longrightarrow$ Conjecture for analytic form of solutions

Conclusion

- First accelerated convergence guarantee for gradient descent using long steps

$$\min_{s \le T} f(x_s) - f(x^\star) = O\left(\frac{1}{T^{1.02449}}\right)$$

- How did we do this?
    - Straightforward patterns: where flat regions are worst case
    - Can certify straightforwardness by (analytically) solving SDPs
    - Alternating between pattern spotting and solving larger and larger SDPs $\longrightarrow$ Conjecture for analytic form of solutions
- Where to go from here?

Conclusion

- First accelerated convergence guarantee for gradient descent using long steps

$$\min_{s \leq T} f(x_s) - f(x^\star) = O\left(\frac{1}{T^{1.02449}}\right)$$

- How did we do this?
  - Straightforward patterns: where flat regions are worst case
  - Can certify straightforwardness by (analytically) solving SDPs
  - Alternating between pattern spotting and solving larger and larger SDPs $\longrightarrow$ Conjecture for analytic form of solutions
- Where to go from here?
  - Strongly convex setting $\longrightarrow O(\kappa^{0.976} \log(\epsilon^{-1}))$

Conclusion

- First accelerated convergence guarantee for gradient descent using long steps

$$\min_{s\leq T} f(x_s) - f(x^\star) = O\left(\frac{1}{T^{1.02449}}\right)$$

- How did we do this?
  - Straightforward patterns: where flat regions are worst case
  - Can certify straightforwardness by (analytically) solving SDPs
  - Alternating between pattern spotting and solving larger and larger SDPs $\longrightarrow$ Conjecture for analytic form of solutions
- Where to go from here?
  - Strongly convex setting $\longrightarrow$ $O(\kappa^{0.976}\log(\epsilon^{-1}))$
  - Nonconvex setting

Conclusion

- First accelerated convergence guarantee for gradient descent using long steps

$$\min_{s \leq T} f(x_s) - f(x^\star) = O\left(\frac{1}{T^{1.02449}}\right)$$

- How did we do this?
  - Straightforward patterns: where flat regions are worst case
  - Can certify straightforwardness by (analytically) solving SDPs
  - Alternating between pattern spotting and solving larger and larger SDPs $\longrightarrow$ Conjecture for analytic form of solutions
- Where to go from here?
  - Strongly convex setting $\longrightarrow O(\kappa^{0.976} \log(\epsilon^{-1}))$
  - Nonconvex setting
  - Understanding/remedying robustness

Conclusion

- First accelerated convergence guarantee for gradient descent using long steps

$$\min_{s\leq T} f(x_s) - f(x^\star) = O\left(\frac{1}{T^{1.02449}}\right)$$

- How did we do this?
  - Straightforward patterns: where flat regions are worst case
  - Can certify straightforwardness by (analytically) solving SDPs
  - Alternating between pattern spotting and solving larger and larger SDPs $\longrightarrow$ Conjecture for analytic form of solutions
- Where to go from here?
  - Strongly convex setting $\longrightarrow$ $O(\kappa^{0.976}\log(\epsilon^{-1}))$
  - Nonconvex setting
  - Understanding/remedying robustness

Questions?

References I

Agarwal, N., Goel, S., and Zhang, C. (2021). Acceleration via fractal learning rate schedules. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 87–99.

Altschuler, J. (2018). Greed, hedging, and acceleration in convex optimization. Master's thesis, Massachusetts Institute of Technology.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202.

Candès, E. J., Strohmer, T., and Voroninski, V. (2013). Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming. *Communications on Pure and Applied Mathematics*, 66(8):1241–1274.

Candes, E. J. and Tao, T. (2005). Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215.

Chen, Y., Chi, Y., and Goldsmith, A. J. (2015). Exact and stable covariance estimation from quadratic sampling via convex programming. *IEEE Transactions on Information Theory*, 61(7):4034–4059.

Das Gupta, S., Parys, B. P. V., and Ryu, E. (2023). Branch-and-bound performance estimation programming: A unified methodology for constructing optimal optimization methods. *Mathematical Programming*.

References II

Drori, Y. and Teboulle, M. (2012). Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145:451–482.

Lebedev, V. and Finogenov, S. (1971). Ordering of the iterative parameters in the cyclical chebyshev iterative method. *USSR Computational Mathematics and Mathematical Physics*, 11(2):155–170.

Loshchilov, I. and Hutter, F. (2016). Sgdr: Stochastic gradient descent with restarts. *ArXiv*, abs/1608.03983.

Oymak, S. (2021). Provable super-convergence with a large cyclical learning rate. *IEEE Signal Process. Lett.*, 28:1645–1649.

Polyak, B. T. (1969). Minimization of unsmooth functionals. *USSR Computational Mathematics and Mathematical Physics*, 9(3):14–29.

Recht, B., Fazel, M., and Parrilo, P. A. (2010). Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501.

Renegar, J. and Grimmer, B. (2022). A simple nearly optimal restart scheme for speeding up first-order methods. *Foundations of Computational Mathematics*, 22(1):211–256.

Roulet, V. and d'Aspremont, A. (2017). Sharpness, restart and acceleration. *Advances in Neural Information Processing Systems*, 30.

References III

Smith, L. N. (2015). Cyclical learning rates for training neural networks. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472.

Smith, L. N. and Topin, N. (2017). Super-convergence: very fast training of neural networks using large learning rates. In *Defense + Commercial Sensing*.

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

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288.

Yang, T. and Lin, Q. (2018). Rsg: Beating subgradient method without smoothness and strong convexity. *The Journal of Machine Learning Research*, 19(1):236–268.

Young, D. (1953). On richardson's method for solving linear systems with positive definite matrices. *Journal of Mathematics and Physics*, 32(1-4):243–255.