Online Learning Guided Quasi-Newton Methods: Improved Global Non-asymptotic Guarantees

Aryan Mokhtari

Department of Electrical and Computer Engineering, UT Austin

Based on joint work with Ruichen Jiang and Qiujiang Jin

Cornell University, October 31st, 2023

Convex Minimization

- Consider the general unconstrained minimization problem

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

where f is L_1-smooth (i.e. $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\| \leq L_1 \|\boldsymbol{x} - \boldsymbol{y}\|$)

Convex Minimization

- Consider the general unconstrained minimization problem

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

where f is L_1-smooth (i.e. $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\| \leq L_1 \|\boldsymbol{x} - \boldsymbol{y}\|$)

- We will focus on two general settings
  - Case I: f is μ-strongly convex
  - Case II: f is (only) convex

Convex Minimization

- Consider the general unconstrained minimization problem

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

where f is L_1-smooth (i.e. $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\| \leq L_1\|\boldsymbol{x} - \boldsymbol{y}\|$)

- We will focus on two general settings
  - Case I: f is μ-strongly convex
  - Case II: f is (only) convex

- We are interested in settings where we can only query first-order information
  $\Rightarrow$ We only have access to $\nabla f(x)$

Gradient Descent-type Methods

- Popular methods: Gradient Descent (GD) and its Accelerated version (AGD)
  - Require only access to gradient oracle $\Rightarrow$ Cost per iteration $\mathcal{O}(d)$

Gradient Descent-type Methods

- Popular methods: Gradient Descent (GD) and its Accelerated version (AGD)
  - Require only access to gradient oracle $\Rightarrow$ Cost per iteration $\mathcal{O}(d)$
  - In Case I (SCVX): Achieve a Global linear convergence rate

$$\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 \leq \rho^k \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2$$

  where $\rho = 1 - \mu/L_1$ for GD and $\rho = 1 - \sqrt{\mu/L_1}$ for AGD.

Gradient Descent-type Methods

- Popular methods: Gradient Descent (GD) and its Accelerated version (AGD)
  - Require only access to gradient oracle $\Rightarrow$ Cost per iteration $\mathcal{O}(d)$
  - In Case I (SCVX): Achieve a Global linear convergence rate

$$\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 \leq \rho^k \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2$$

    where $\rho = 1 - \mu/L_1$ for GD and $\rho = 1 - \sqrt{\mu/L_1}$ for AGD.
  - In Case II (CVX): Achieve a Global sublinear convergence rate

$$f(\boldsymbol{x}_k) - f^* \leq \frac{C}{k^\alpha}$$

    where $\alpha = 1$ for GD and $\alpha = 2$ for AGD.

Quasi-Newton Methods

- Quasi-Newton (QN) methods aim at speeding up GD-type methods by approximating the function's curvature and using a preconditioner

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \rho_k \mathbf{B}_k^{-1} \nabla f(\boldsymbol{x}_k)$$

Quasi-Newton Methods

- Quasi-Newton (QN) methods aim at speeding up GD-type methods by approximating the function's curvature and using a preconditioner

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \rho_k \mathbf{B}_k^{-1} \nabla f(\mathbf{x}_k)$$

- When $\mathbf{B}_k \approx \nabla^2 f(\mathbf{x}_k)$ they mimic Newton's method

Quasi-Newton Methods

- Quasi-Newton (QN) methods aim at speeding up GD-type methods by approximating the function's curvature and using a preconditioner

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \rho_k \mathbf{B}_k^{-1} \nabla f(\boldsymbol{x}_k)$$

- When $\mathbf{B}_k \approx \nabla^2 f(\boldsymbol{x}_k)$ they mimic Newton's method
- Only use gradient to construct $\mathbf{B}_k \Rightarrow$ Still first-order methods

Quasi-Newton Methods

- Quasi-Newton (QN) methods aim at speeding up GD-type methods by approximating the function's curvature and using a preconditioner

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \rho_k \mathbf{B}_k^{-1} \nabla f(\boldsymbol{x}_k)$$

- When $\mathbf{B}_k \approx \nabla^2 f(\boldsymbol{x}_k)$ they mimic Newton's method
- Only use gradient to construct $\mathbf{B}_k \Rightarrow$ Still first-order methods
- Various updates for $\mathbf{B}_k$ have been proposed with cost $\mathcal{O}(d^2)$ (BFGS, DFP, SR1)

Quasi-Newton Methods

- Quasi-Newton (QN) methods aim at speeding up GD-type methods by approximating the function's curvature and using a preconditioner

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \rho_k \mathbf{B}_k^{-1} \nabla f(\mathbf{x}_k)$$

- When $\mathbf{B}_k \approx \nabla^2 f(\mathbf{x}_k)$ they mimic Newton's method
- Only use gradient to construct $\mathbf{B}_k \Rightarrow$ Still first-order methods
- Various updates for $\mathbf{B}_k$ have been proposed with cost $\mathcal{O}(d^2)$ (BFGS, DFP, SR1)
- Main ideas:
  - *Proximity condition*: Keep $\mathbf{B}_k$ and $\mathbf{B}_{k+1}$ close
  - *Secant condition*: $\mathbf{B}_{k+1}\mathbf{s}_k = \mathbf{y}_k$ where $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, $\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$

Quasi-Newton Methods

- Quasi-Newton (QN) methods aim at speeding up GD-type methods by approximating the function's curvature and using a preconditioner

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \rho_k \mathbf{B}_k^{-1} \nabla f(\mathbf{x}_k)$$

- When $\mathbf{B}_k \approx \nabla^2 f(\mathbf{x}_k)$ they mimic Newton's method
- Only use gradient to construct $\mathbf{B}_k \Rightarrow$ Still first-order methods

- Various updates for $\mathbf{B}_k$ have been proposed with cost $\mathcal{O}(d^2)$ (BFGS, DFP, SR1)
- Main ideas:
  - *Proximity condition*: Keep $\mathbf{B}_k$ and $\mathbf{B}_{k+1}$ close
  - *Secant condition*: $\mathbf{B}_{k+1}\mathbf{s}_k = \mathbf{y}_k$ where $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, $\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$

$$\begin{aligned} \mathbf{B}_{k+1} \;&=\; \operatorname{argmin} \;\|\mathbf{B} - \mathbf{B}_k\|_{\mathbf{V}} \quad \Longleftrightarrow \quad \mathbf{B}_{k+1}^{-1} = \left(\mathbf{I} - \frac{\mathbf{s}_k \mathbf{y}_k^\top}{\mathbf{s}_k^\top \mathbf{y}_k}\right) \mathbf{B}_k^{-1} \left(\mathbf{I} - \frac{\mathbf{y}_k \mathbf{s}_k^\top}{\mathbf{s}_k^\top \mathbf{y}_k}\right) + \frac{\mathbf{s}_k \mathbf{s}_k^\top}{\mathbf{s}_k^\top \mathbf{y}_k} \\ &\text{s.t.} \quad \mathbf{B}\,\mathbf{s}_k = \mathbf{y}_k, \quad \mathbf{B} \succeq \mathbf{0} \end{aligned}$$

Performance of QN Methods

- Several studies have illustrated the superior performance of QN methods
- However, there is no result proving this advantage for QN algorithms

Results on Quasi-Newton Methods (Strongly Convex setting)

- In the SCVX setting, classic results have shown asymptotic local superlinear convergence for QN methods: when $\|\mathbf{x}_k - \mathbf{x}^*\|$ is small,

$$\lim_{k\to\infty} \frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}_k - \mathbf{x}^*\|} = 0$$

Results on Quasi-Newton Methods (Strongly Convex setting)

- In the SCVX setting, classic results have shown asymptotic local superlinear convergence for QN methods: when $\|\mathbf{x}_k - \mathbf{x}^*\|$ is small,

$$\lim_{k\to\infty} \frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}_k - \mathbf{x}^*\|} = 0$$

  - Local superlinear rate [Broyden-Dennis-Moré'73][Dennis-Moré'74]
  - Global and superlinear rate with exact linesearch [Powell'71][Dixon'72]
  - Global and superlinear rate with inexact linesearch [Powell'76][Bryd-Nocedal-Yuan'87]
  - Many other works: [Griewank-Toint'82; Dennis-Martinez-Tapia'89; Yuan'91; Al-Baali'98; Li-Fukushima'99; Yabe-Ogasawara-Yoshino'07; M-Eisen-Ribeiro'18; Gao-Goldfarb'19]

Results on Quasi-Newton Methods (Strongly Convex setting)

- In the SCVX setting, classic results have shown asymptotic local superlinear convergence for QN methods: when $\|\mathbf{x}_k - \mathbf{x}^*\|$ is small,

$$\lim_{k\to\infty} \frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}_k - \mathbf{x}^*\|} = 0$$

  - Local superlinear rate [Broyden-Dennis-Moré'73][Dennis-Moré'74]
  - Global and superlinear rate with exact linesearch [Powell'71][Dixon'72]
  - Global and superlinear rate with inexact linesearch [Powell'76][Bryd-Nocedal-Yuan'87]
  - Many other works: [Griewank-Toint'82; Dennis-Martinez-Tapia'89; Yuan'91; Al-Baali'98; Li-Fukushima'99; Yabe-Ogasawara-Yoshino'07; M-Eisen-Ribeiro'18; Gao-Goldfarb'19]
- However, they are all asymptotic and fail to provide an explicit convergence rate
- The global linear results are no better than GD or AGD

Recent Results on Quasi-Newton Methods (Strongly Convex setting)

- Recent results show explicit non-asymptotic local superlinear rate for BFGS and DFP

Recent Results on Quasi-Newton Methods (Strongly Convex setting)

- Recent results show explicit non-asymptotic local superlinear rate for BFGS and DFP
- Rodomanov-Nesterov'21 and Jin-M'22 concurrently but using different Lyapunov functions showed superlinear rates of the form $O((1/\sqrt{k})^k)$

Recent Results on Quasi-Newton Methods (Strongly Convex setting)

- Recent results show explicit non-asymptotic local superlinear rate for BFGS and DFP
- Rodomanov-Nesterov'21 and Jin-M'22 concurrently but using different Lyapunov functions showed superlinear rates of the form $O((1/\sqrt{k})^k)$

	cond. on $\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|$	cond. on $\mathbf{B}_0$	rate
[Jin-M'22]	$\mathcal{O}\left(\frac{1}{\sqrt{d}}\right)$	$\mathbf{B}_0 = \nabla^2 f(\boldsymbol{x}_0)$	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)^k$
[Rodomanov-Nesterov'21]	$\mathcal{O}\left(\frac{1}{d}\right)$	$\nabla^2 f(\boldsymbol{x}) \preceq \mathbf{B}_0 \preceq \kappa \nabla^2 f(\boldsymbol{x})$	$\mathcal{O}\left(\sqrt{\frac{d \ln \kappa}{k}}\right)^k$

Table: Definition $\kappa = L_1/\mu$

Recent Results on Quasi-Newton Methods (Strongly Convex setting)

- Recent results show explicit non-asymptotic local superlinear rate for BFGS and DFP
- Rodomanov-Nesterov'21 and Jin-M'22 concurrently but using different Lyapunov functions showed superlinear rates of the form $O((1/\sqrt{k})^k)$

	cond. on $\|\mathbf{x}_0 - \mathbf{x}^*\|$	cond. on $\mathbf{B}_0$	rate
[Jin-M'22]	$\mathcal{O}\left(\frac{1}{\sqrt{d}}\right)$	$\mathbf{B}_0 = \nabla^2 f(\mathbf{x}_0)$	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)^k$
[Rodomanov-Nesterov'21]	$\mathcal{O}\left(\frac{1}{d}\right)$	$\nabla^2 f(\mathbf{x}) \preceq \mathbf{B}_0 \preceq \kappa \nabla^2 f(\mathbf{x})$	$\mathcal{O}\left(\sqrt{\frac{d \ln \kappa}{k}}\right)^k$

Table: Definition $\kappa = L_1/\mu$

- These results are only local, it is unclear how to extend them into global guarantees
  $\Rightarrow$ The condition on $\mathbf{B}_0$ may not hold when $\|\mathbf{x}_0 - \mathbf{x}^*\|$ becomes small
- Moreover, there is no global result matching the linear rate of AGD or GD

Results for the Convex Setting

- In the CVX setting, few results are known for classical QN methods
  - $\lim_{k\to\infty} f(\boldsymbol{x}_k) = f(\boldsymbol{x}^*)$ with exact line search [Powell'72]
  - $\liminf_{k\to\infty} \|\nabla f(\boldsymbol{x}_k)\| = 0$ with inexact line search [Powell'76; Byrd-Nocedal-Yuan'87]

Results for the Convex Setting

- In the CVX setting, few results are known for classical QN methods
  - $\lim_{k\to\infty} f(\mathbf{x}_k) = f(\mathbf{x}^*)$ with exact line search [Powell'72]
  - $\liminf_{k\to\infty} \|\nabla f(\mathbf{x}_k)\| = 0$ with inexact line search [Powell'76; Byrd-Nocedal-Yuan'87]
- Along another line of work, analyzing QN methods as preconditioned GD methods
  - $\mathcal{O}(1/k)$ rate is shown in [Scheinberg-Tang'16]
  - An accelerated $\mathcal{O}(1/k^2)$ rate is achieved in [Ghanbari-Scheinberg'18]
- However, the rates are no better than that of AGD $\Rightarrow$ no provable gain

Goal and Main Ideas of our Proposed Approach

- **Goal:** Designing QN methods with superior gradient complexity compared to GD-type methods in both CVX and SCVX settings.

Goal and Main Ideas of our Proposed Approach

- **Goal:** Designing QN methods with superior gradient complexity compared to GD-type methods in both CVX and SCVX settings.
- **Our Approach:** Online-Learning guided Quasi-Newton Proximal Extragradient (QNPE) Algorithms

Goal and Main Ideas of our Proposed Approach

- **Goal:** Designing QN methods with superior gradient complexity compared to GD-type methods in both CVX and SCVX settings.

- **Our Approach:** Online-Learning guided Quasi-Newton Proximal Extragradient (QNPE) Algorithms

- **Main Ideas:**
  - Instead of the classic template of QN methods ($\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \rho_k \mathbf{B}_k^{-1} \nabla f(\boldsymbol{x}_k)$), we follow the Hybrid Proximal Extragradient (HPE) framework
  - Instead of updating $\mathbf{B}_k$ by enforcing the Proximity condition and Secant condition, we use an Online Learning framework for updating $\mathbf{B}_k$ inspired by our analysis

Our Contributions (Strongly-Convex Setting)

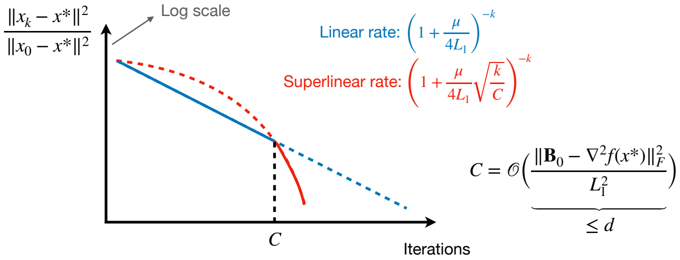
- Global convergence rates (no conditions on $\boldsymbol{x}_0$ or $\mathbf{B}_0$) [Jiang-Jin-M, COLT '23]

$$\frac{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2} \leq \min\left\{\left(1 + \frac{\mu}{4L_1}\right)^{-k}, \left(1 + \frac{\mu}{4L_1}\sqrt{\frac{k}{C}}\right)^{-k}\right\}$$

Our Contributions (Strongly-Convex Setting)

- Global convergence rates (no conditions on $\boldsymbol{x}_0$ or $\mathbf{B}_0$) [Jiang-Jin-M, COLT '23]

$$\frac{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2} \leq \min\left\{\left(1 + \frac{\mu}{4L_1}\right)^{-k}, \left(1 + \frac{\mu}{4L_1}\sqrt{\frac{k}{C}}\right)^{-k}\right\}$$



- For $k \leq d$, QNPE matches the linear rate of GD
- After at most $\mathcal{O}(d)$ iterations QNPE becomes provably faster than GD

Our Contributions (Convex Setting)

- An accelerated QN proximal extragradient method [Jiang-M, NeurIPS '23]

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}) \leq \mathcal{O}\left(\min\left\{\frac{1}{k^2}, \frac{\sqrt{d \log k}}{k^{2.5}}\right\}\right)$$

  - for $k \leq d \log d$, it matches the rate of AGD
  - for $k \geq d \log d$, it provably converges faster than AGD

Our Contributions (Convex Setting)

- An accelerated QN proximal extragradient method [Jiang-M, NeurIPS '23]

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}) \leq \mathcal{O}\left(\min\left\{\frac{1}{k^2}, \frac{\sqrt{d\log k}}{k^{2.5}}\right\}\right)$$

  - for $k \leq d \log d$, it matches the rate of AGD
  - for $k \geq d \log d$, it provably converges faster than AGD

- Lower bound discussion:
  - This result does not violate the lower bound for first-order methods
  - The lower bound of $\Omega\left(\frac{1}{k^2}\right)$ only holds for $k \leq d$

Hybrid Proximal Extragradient

- We follow (a variant of) the Hybrid Proximal Extragradient (HPE) framework [Solodov-Svaiter'99; Monteiro-Svaiter'10]

Hybrid Proximal Extragradient

- We follow (a variant of) the Hybrid Proximal Extragradient (HPE) framework [Solodov-Svaiter'99; Monteiro-Svaiter'10]
- Stage 1: Inexact proximal point update

$$\hat{\boldsymbol{x}}_k \approx \boldsymbol{x}_k - \eta_k \nabla f(\hat{\boldsymbol{x}}_k)$$

Hybrid Proximal Extragradient

- We follow (a variant of) the Hybrid Proximal Extragradient (HPE) framework [Solodov-Svaiter'99; Monteiro-Svaiter'10]
- Stage 1: Inexact proximal point update

$$\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k + \eta_k \nabla f(\hat{\boldsymbol{x}}_k)\| \leq \frac{1}{2}\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|$$

Hybrid Proximal Extragradient

- We follow (a variant of) the Hybrid Proximal Extragradient (HPE) framework [Solodov-Svaiter'99; Monteiro-Svaiter'10]
- Stage 1: Inexact proximal point update

$$\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k + \eta_k \nabla f(\hat{\boldsymbol{x}}_k)\| \leq \frac{1}{2}\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|$$

- Stage 2: Extragradient step

$$\boldsymbol{x}_{k+1} = \gamma_k[\boldsymbol{x}_k - \eta_k \nabla f(\hat{\boldsymbol{x}}_k)] + (1-\gamma_k)\hat{\boldsymbol{x}}_k, \quad \gamma_k = \frac{1}{1+2\eta_k\mu}$$

Hybrid Proximal Extragradient

- We follow (a variant of) the Hybrid Proximal Extragradient (HPE) framework [Solodov-Svaiter'99; Monteiro-Svaiter'10]
- Stage 1: Inexact proximal point update

$$\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k + \eta_k \nabla f(\hat{\boldsymbol{x}}_k)\| \leq \frac{1}{2}\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|$$

- Stage 2: Extragradient step

$$\boldsymbol{x}_{k+1} = \gamma_k[\boldsymbol{x}_k - \eta_k \nabla f(\hat{\boldsymbol{x}}_k)] + (1-\gamma_k)\hat{\boldsymbol{x}}_k, \quad \gamma_k = \frac{1}{1+2\eta_k\mu}$$

Hybrid Proximal Extragradient

- We follow (a variant of) the Hybrid Proximal Extragradient (HPE) framework [Solodov-Svaiter'99; Monteiro-Svaiter'10]
- Stage 1: Inexact proximal point update

$$\|\hat{\mathbf{x}}_k - \mathbf{x}_k + \eta_k \nabla f(\hat{\mathbf{x}}_k)\| \leq \frac{1}{2}\|\hat{\mathbf{x}}_k - \mathbf{x}_k\|$$

- Stage 2: Extragradient step

$$\mathbf{x}_{k+1} = \gamma_k[\mathbf{x}_k - \eta_k \nabla f(\hat{\mathbf{x}}_k)] + (1-\gamma_k)\hat{\mathbf{x}}_k, \quad \gamma_k = \frac{1}{1+2\eta_k\mu}$$

- $\|\mathbf{x}_{k+1} - \mathbf{x}^*\|^2 \leq \frac{1}{1+2\eta_k\mu}\|\mathbf{x}_k - \mathbf{x}^*\|^2 \Rightarrow$ any rate can be achieved as $\eta_k \uparrow$

Newton Proximal Extragradient

- Issue: Subproblem in Stage 1 is costly!

$$\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k + \eta_k \nabla f(\hat{\boldsymbol{x}}_k)\| \leq \frac{1}{2}\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|$$

Newton Proximal Extragradient

- Issue: Subproblem in Stage 1 is costly!

$$\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k + \eta_k \nabla f(\hat{\boldsymbol{x}}_k)\| \leq \frac{1}{2}\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|$$

- Solution: Linearize $\nabla f(\hat{\boldsymbol{x}}_k) \approx \nabla f(\boldsymbol{x}_k) + \nabla^2 f(\boldsymbol{x}_k)(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)$
  $\Rightarrow$ subproblem becomes a linear system of equations

Newton Proximal Extragradient

- Issue: Subproblem in Stage 1 is costly!

$$\|\hat{\mathbf{x}}_k - \mathbf{x}_k + \eta_k \nabla f(\hat{\mathbf{x}}_k)\| \leq \frac{1}{2}\|\hat{\mathbf{x}}_k - \mathbf{x}_k\|$$

- Solution: Linearize $\nabla f(\hat{\mathbf{x}}_k) \approx \nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)$
  $\Rightarrow$ subproblem becomes a linear system of equations
- Stage 1: Newton proximal step [Monteiro-Svaiter'12]

$$\|\hat{\mathbf{x}}_k - \mathbf{x}_k + \eta_k(\nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k))\| \leq \frac{1}{4}\|\hat{\mathbf{x}}_k - \mathbf{x}_k\|,$$

$$\eta_k\|\nabla f(\hat{\mathbf{x}}_k) - (\nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k))\| \leq \frac{1}{4}\|\hat{\mathbf{x}}_k - \mathbf{x}_k\|$$

Newton Proximal Extragradient

- Issue: Subproblem in Stage 1 is costly!

$$\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k + \eta_k \nabla f(\hat{\boldsymbol{x}}_k)\| \leq \frac{1}{2}\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|$$

- Solution: Linearize $\nabla f(\hat{\boldsymbol{x}}_k) \approx \nabla f(\boldsymbol{x}_k) + \nabla^2 f(\boldsymbol{x}_k)(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)$
  $\Rightarrow$ subproblem becomes a linear system of equations
- Stage 1: Newton proximal step [Monteiro-Svaiter'12]

$$\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k + \eta_k(\nabla f(\boldsymbol{x}_k) + \nabla^2 f(\boldsymbol{x}_k)(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k))\| \leq \frac{1}{4}\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|,$$

$$\eta_k\|\nabla f(\hat{\boldsymbol{x}}_k) - (\nabla f(\boldsymbol{x}_k) + \nabla^2 f(\boldsymbol{x}_k)(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k))\| \leq \frac{1}{4}\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|$$

Newton Proximal Extragradient

- Issue: Subproblem in Stage 1 is costly!

$$\|\hat{\mathbf{x}}_k - \mathbf{x}_k + \eta_k \nabla f(\hat{\mathbf{x}}_k)\| \leq \frac{1}{2}\|\hat{\mathbf{x}}_k - \mathbf{x}_k\|$$

- Solution: Linearize $\nabla f(\hat{\mathbf{x}}_k) \approx \nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)$
  $\Rightarrow$ subproblem becomes a linear system of equations
- Stage 1: Newton proximal step [Monteiro-Svaiter'12]

$$\|\hat{\mathbf{x}}_k - \mathbf{x}_k + \eta_k(\nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k))\| \leq \frac{1}{4}\|\hat{\mathbf{x}}_k - \mathbf{x}_k\|,$$

$$\eta_k\|\nabla f(\hat{\mathbf{x}}_k) - (\nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k))\| \leq \frac{1}{4}\|\hat{\mathbf{x}}_k - \mathbf{x}_k\|$$

  - η_k is not arbitrary; requires backtracking line search

Newton Proximal Extragradient

- Issue: Subproblem in Stage 1 is costly!

$$\|\hat{\mathbf{x}}_k - \mathbf{x}_k + \eta_k \nabla f(\hat{\mathbf{x}}_k)\| \leq \frac{1}{2}\|\hat{\mathbf{x}}_k - \mathbf{x}_k\|$$

- Solution: Linearize $\nabla f(\hat{\mathbf{x}}_k) \approx \nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)$
  $\Rightarrow$ subproblem becomes a linear system of equations
- Stage 1: Newton proximal step [Monteiro-Svaiter'12]

$$\|\hat{\mathbf{x}}_k - \mathbf{x}_k + \eta_k(\nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k))\| \leq \frac{1}{4}\|\hat{\mathbf{x}}_k - \mathbf{x}_k\|,$$

$$\eta_k\|\nabla f(\hat{\mathbf{x}}_k) - (\nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k))\| \leq \frac{1}{4}\|\hat{\mathbf{x}}_k - \mathbf{x}_k\|$$

  - η_k is not arbitrary; requires backtracking line search
- To obtain a quasi-Newton method, we first replace $\nabla^2 f(\mathbf{x}_k)$ by $\mathbf{B}_k$

Quasi-Newton Proximal Extragradient

- Stage 1: Quasi-Newton proximal step

$$\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k + \eta_k(\nabla f(\boldsymbol{x}_k) + \mathbf{B}_k(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k))\| \leq \frac{1}{4}\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|,$$

$$\eta_k\|\nabla f(\hat{\boldsymbol{x}}_k) - (\nabla f(\boldsymbol{x}_k) + \mathbf{B}_k(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k))\| \leq \frac{1}{4}\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|$$

  - Given $\boldsymbol{x}_k$ and $\mathbf{B}_k$, use backtracking line search to find η_k and $\hat{\boldsymbol{x}}_k$

- Stage 2: Extragradient step

$$\boldsymbol{x}_{k+1} = \gamma_k[\boldsymbol{x}_k - \eta_k \nabla f(\hat{\boldsymbol{x}}_k)] + (1 - \gamma_k)\hat{\boldsymbol{x}}_k, \quad \gamma_k = \frac{1}{1 + 2\eta_k \mu}$$

Quasi-Newton Proximal Extragradient

- Stage 1: Quasi-Newton proximal step

$$\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k + \eta_k(\nabla f(\boldsymbol{x}_k) + \mathbf{B}_k(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k))\| \leq \frac{1}{4}\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|,$$

$$\eta_k\|\nabla f(\hat{\boldsymbol{x}}_k) - (\nabla f(\boldsymbol{x}_k) + \mathbf{B}_k(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k))\| \leq \frac{1}{4}\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|$$

  - Given $\boldsymbol{x}_k$ and $\mathbf{B}_k$, use backtracking line search to find η_k and $\hat{\boldsymbol{x}}_k$

- Stage 2: Extragradient step

$$\boldsymbol{x}_{k+1} = \gamma_k[\boldsymbol{x}_k - \eta_k \nabla f(\hat{\boldsymbol{x}}_k)] + (1 - \gamma_k)\hat{\boldsymbol{x}}_k, \quad \gamma_k = \frac{1}{1 + 2\eta_k \mu}$$

- The remaining question: how to select $\{\mathbf{B}_k\}$?

How to Update $\mathbf{B}_k$: Starting from Convergence Analysis

- How should we select/update $\{\mathbf{B}_k\}$?

How to Update $\mathbf{B}_k$: Starting from Convergence Analysis

- How should we select/update $\{\mathbf{B}_k\}$?
- We don't follow the classic approaches that use Proximity and Secant conditions

How to Update $\mathbf{B}_k$: Starting from Convergence Analysis

- How should we select/update $\{\mathbf{B}_k\}$?
- We don't follow the classic approaches that use Proximity and Secant conditions
- We let the convergence analysis guide our choice of $\mathbf{B}_k$!

How to Update $\mathbf{B}_k$: Starting from Convergence Analysis

- How should we select/update $\{\mathbf{B}_k\}$?
- We don't follow the classic approaches that use Proximity and Secant conditions
- We let the convergence analysis guide our choice of $\mathbf{B}_k$!
- We know that $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 \leq \frac{1}{1+2\eta_k\mu}\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2$

How to Update $\mathbf{B}_k$: Starting from Convergence Analysis

- How should we select/update $\{\mathbf{B}_k\}$?
- We don't follow the classic approaches that use Proximity and Secant conditions
- We let the convergence analysis guide our choice of $\mathbf{B}_k$!
- We know that $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|^2 \leq \frac{1}{1+2\eta_k\mu}\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2$
- η_k is constrained by

$$\eta_k\|\nabla f(\hat{\boldsymbol{x}}_k) - (\nabla f(\boldsymbol{x}_k) + \mathbf{B}_k(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k))\| \leq \frac{1}{4}\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|$$

- **Initial result:** By backtracking line search:

$$\eta_k \simeq \frac{\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|}{\|\nabla f(\hat{\boldsymbol{x}}_k) - \nabla f(\boldsymbol{x}_k) - \mathbf{B}_k(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)\|} = \frac{\|\boldsymbol{s}_k\|}{\|\boldsymbol{y}_k - \mathbf{B}_k\boldsymbol{s}_k\|},$$

where $\boldsymbol{y}_k = \nabla f(\hat{\boldsymbol{x}}_k) - \nabla f(\boldsymbol{x}_k)$ and $\boldsymbol{s}_k = \hat{\boldsymbol{x}}_k - \boldsymbol{x}_k$

How to Update $\mathbf{B}_k$: Starting from Convergence Analysis

- After N iterations, we have

$$\frac{\|\boldsymbol{x}_N - \boldsymbol{x}^*\|^2}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2} \leq \prod_{k=0}^{N-1} (1 + 2\eta_k \mu)^{-1} \leq \left(1 + 2\mu\sqrt{\frac{N}{\sum_{k=0}^{N-1} 1/\eta_k^2}}\right)^{-N}$$

How to Update $\mathbf{B}_k$: Starting from Convergence Analysis

- After N iterations, we have

$$\frac{\|\boldsymbol{x}_N - \boldsymbol{x}^*\|^2}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2} \leq \prod_{k=0}^{N-1} (1 + 2\eta_k \mu)^{-1} \leq \left(1 + 2\mu \sqrt{\frac{N}{\sum_{k=0}^{N-1} 1/\eta_k^2}}\right)^{-N}$$

- Since $\eta_k \simeq \|\boldsymbol{s}_k\| / \|\boldsymbol{y}_k - \mathbf{B}_k \boldsymbol{s}_k\|$, we further have

$$\frac{\|\boldsymbol{x}_N - \boldsymbol{x}^*\|^2}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2} \leq \left(1 + 2\mu \sqrt{\frac{N}{\mathcal{O}(\sum_{k=0}^{N-1} \|\boldsymbol{y}_k - \mathbf{B}_k \boldsymbol{s}_k\|^2 / \|\boldsymbol{s}_k\|^2)}}\right)^{-N},$$

How to Update $\mathbf{B}_k$: Starting from Convergence Analysis

- After N iterations, we have

$$\frac{\|\boldsymbol{x}_N - \boldsymbol{x}^*\|^2}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2} \leq \prod_{k=0}^{N-1} (1 + 2\eta_k \mu)^{-1} \leq \left(1 + 2\mu\sqrt{\frac{N}{\sum_{k=0}^{N-1} 1/\eta_k^2}}\right)^{-N}$$

- Since $\eta_k \simeq \|\boldsymbol{s}_k\|/\|\boldsymbol{y}_k - \mathbf{B}_k \boldsymbol{s}_k\|$, we further have

$$\frac{\|\boldsymbol{x}_N - \boldsymbol{x}^*\|^2}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2} \leq \left(1 + 2\mu\sqrt{\frac{N}{\mathcal{O}(\sum_{k=0}^{N-1} \|\boldsymbol{y}_k - \mathbf{B}_k \boldsymbol{s}_k\|^2/\|\boldsymbol{s}_k\|^2)}}\right)^{-N},$$

- Hence, the goal is to choose $\mathbf{B}_k$ such that we minimize

$$\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k) \; := \; \sum_{k=0}^{N-1} \frac{\|\boldsymbol{y}_k - \mathbf{B}_k \boldsymbol{s}_k\|^2}{\|\boldsymbol{s}_k\|^2}$$

Hessian Approximation Update via Online Learning

- We aim to minimize $\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k)$, but we observe ℓ_k after selecting $\mathbf{B}_k$!

Hessian Approximation Update via Online Learning

- We aim to minimize $\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k)$, but we observe ℓ_k after selecting $\mathbf{B}_k$!

  Why? given $\boldsymbol{x}_k, \mathbf{B}_k \rightarrow$ select $(\eta_k, \hat{\boldsymbol{x}}_k)$ by BLS $\rightarrow$ compute $\boldsymbol{s}_k, \boldsymbol{y}_k \rightarrow$ compute $\ell_k(\mathbf{B}_k)$

Hessian Approximation Update via Online Learning

- We aim to minimize $\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k)$, but we observe ℓ_k after selecting $\mathbf{B}_k$!

  Why? given $\boldsymbol{x}_k, \mathbf{B}_k \to$ select $(\eta_k, \hat{\boldsymbol{x}}_k)$ by BLS $\to$ compute $\boldsymbol{s}_k, \boldsymbol{y}_k \to$ compute $\ell_k(\mathbf{B}_k)$

- Key idea: View the update of $\mathbf{B}_k$ as an online convex opt problem
  - Choose $\mathbf{B}_k \in \mathcal{Z}$, where $\mathcal{Z} = \{\mathbf{B} : \mu \mathbf{I} \preceq \mathbf{B} \preceq L_1 \mathbf{I}\}$
  - Receive $\ell_k(\mathbf{B}_k)$
  - Update $\mathbf{B}_{k+1}$ by an online learning algorithm, e.g., Online Gradient Descent

$$\mathbf{B}_{k+1} = \Pi_{\mathcal{Z}}\left(\mathbf{B}_k - \rho \nabla \ell_k(\mathbf{B}_k)\right)$$

Hessian Approximation Update via Online Learning

- We aim to minimize $\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k)$, but we observe ℓ_k after selecting $\mathbf{B}_k$!

  Why? given $\boldsymbol{x}_k, \mathbf{B}_k \rightarrow$ select $(\eta_k, \hat{\boldsymbol{x}}_k)$ by BLS $\rightarrow$ compute $\boldsymbol{s}_k, \boldsymbol{y}_k \rightarrow$ compute $\ell_k(\mathbf{B}_k)$

- Key idea: View the update of $\mathbf{B}_k$ as an online convex opt problem
  - Choose $\mathbf{B}_k \in \mathcal{Z}$, where $\mathcal{Z} = \{\mathbf{B} : \mu\mathbf{I} \preceq \mathbf{B} \preceq L_1\mathbf{I}\}$
  - Receive $\ell_k(\mathbf{B}_k)$
  - Update $\mathbf{B}_{k+1}$ by an online learning algorithm, e.g., Online Gradient Descent

$$\mathbf{B}_{k+1} = \Pi_{\mathcal{Z}}\left(\mathbf{B}_k - \rho\nabla\ell_k(\mathbf{B}_k)\right)$$

- Side note: Why do we project to set $\mathcal{Z}$? You'll see!

Global Linear Convergence

- Now our goal is to upper bound $\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k)$
- A "trivial" bound: since $\mu \mathbf{I} \preceq \mathbf{B}_k \preceq L_1 \mathbf{I}$,

$$\ell_k(\mathbf{B}_k) = \frac{\|\boldsymbol{y}_k - \mathbf{B}_k \boldsymbol{s}_k\|^2}{\|\boldsymbol{s}_k\|^2} \leq \frac{2\|\boldsymbol{y}_k\|^2 + 2\|\mathbf{B}_k \boldsymbol{s}_k\|^2}{\|\boldsymbol{s}_k\|^2} \leq \frac{2\|\boldsymbol{y}_k\|^2}{\|\boldsymbol{s}_k\|^2} + 2L_1^2$$

Global Linear Convergence

- Now our goal is to upper bound $\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k)$
- A "trivial" bound: since $\mu \mathbf{I} \preceq \mathbf{B}_k \preceq L_1 \mathbf{I}$,

$$\ell_k(\mathbf{B}_k) = \frac{\|\boldsymbol{y}_k - \mathbf{B}_k \boldsymbol{s}_k\|^2}{\|\boldsymbol{s}_k\|^2} \leq \frac{2\|\boldsymbol{y}_k\|^2 + 2\|\mathbf{B}_k \boldsymbol{s}_k\|^2}{\|\boldsymbol{s}_k\|^2} \leq \frac{2\|\boldsymbol{y}_k\|^2}{\|\boldsymbol{s}_k\|^2} + 2L_1^2$$

- Recall that $\boldsymbol{y}_k = \nabla f(\hat{\boldsymbol{x}}_k) - \nabla f(\boldsymbol{x}_k)$ and $\boldsymbol{s}_k = \hat{\boldsymbol{x}}_k - \boldsymbol{x}_k$
- Since $\|\nabla f(\hat{\boldsymbol{x}}_k) - \nabla f(\boldsymbol{x}_k)\| \leq L_1 \|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|$, we have $\|\boldsymbol{y}_k\| \leq L_1 \|\boldsymbol{s}_k\|$

Global Linear Convergence

- Now our goal is to upper bound $\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k)$
- A "trivial" bound: since $\mu \mathbf{I} \preceq \mathbf{B}_k \preceq L_1 \mathbf{I}$,

$$\ell_k(\mathbf{B}_k) = \frac{\|\boldsymbol{y}_k - \mathbf{B}_k \boldsymbol{s}_k\|^2}{\|\boldsymbol{s}_k\|^2} \leq \frac{2\|\boldsymbol{y}_k\|^2 + 2\|\mathbf{B}_k \boldsymbol{s}_k\|^2}{\|\boldsymbol{s}_k\|^2} \leq \frac{2\|\boldsymbol{y}_k\|^2}{\|\boldsymbol{s}_k\|^2} + 2L_1^2$$

- Recall that $\boldsymbol{y}_k = \nabla f(\hat{\boldsymbol{x}}_k) - \nabla f(\boldsymbol{x}_k)$ and $\boldsymbol{s}_k = \hat{\boldsymbol{x}}_k - \boldsymbol{x}_k$
- Since $\|\nabla f(\hat{\boldsymbol{x}}_k) - \nabla f(\boldsymbol{x}_k)\| \leq L_1 \|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|$, we have $\|\boldsymbol{y}_k\| \leq L_1 \|\boldsymbol{s}_k\|$
- Thus, we always have $\ell_k(\mathbf{B}_k) \leq 4L_1^2 \quad \Rightarrow \quad \sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k) \leq 4L_1^2 N$
- Plugging this bound back:

$$\frac{\|\boldsymbol{x}_N - \boldsymbol{x}^*\|^2}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2} \leq \left(1 + 2\mu \sqrt{\frac{N}{\mathcal{O}(\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k))}}\right)^{-N} = \left(1 + \Omega\left(\frac{\mu}{L_1}\right)\right)^{-N}$$

Global Superlinear Convergence

- Moreover, we can use a "small-loss" regret bound for Online Gradient Descent

$$\mathbf{B}_{k+1} = \Pi_{\mathcal{Z}}\left(\mathbf{B}_k - \rho \nabla \ell_k(\mathbf{B}_k)\right)$$

Global Superlinear Convergence

- Moreover, we can use a "small-loss" regret bound for Online Gradient Descent

$$\mathbf{B}_{k+1} = \Pi_{\mathcal{Z}}\left(\mathbf{B}_k - \rho \nabla \ell_k(\mathbf{B}_k)\right)$$

  - Standard analysis shows that, for any $\mathbf{H} \in \mathcal{Z}$:

$$\ell_k(\mathbf{B}_k) - \ell_k(\mathbf{H}) \leq \frac{1}{2\rho}\|\mathbf{B}_k - \mathbf{H}\|_F^2 - \frac{1}{2\rho}\|\mathbf{B}_{k+1} - \mathbf{H}\|_F^2 + \frac{\rho}{2}\|\nabla \ell_k(\mathbf{B}_k)\|_F^2$$

Global Superlinear Convergence

- Moreover, we can use a "small-loss" regret bound for Online Gradient Descent

$$\mathbf{B}_{k+1} = \Pi_{\mathcal{Z}}\left(\mathbf{B}_k - \rho \nabla \ell_k(\mathbf{B}_k)\right)$$

  - Standard analysis shows that, for any $\mathbf{H} \in \mathcal{Z}$:

$$\ell_k(\mathbf{B}_k) - \ell_k(\mathbf{H}) \leq \frac{1}{2\rho}\|\mathbf{B}_k - \mathbf{H}\|_F^2 - \frac{1}{2\rho}\|\mathbf{B}_{k+1} - \mathbf{H}\|_F^2 + \frac{\rho}{2}\|\nabla \ell_k(\mathbf{B}_k)\|_F^2$$

  - We also have $\|\nabla \ell_k(\mathbf{B}_k)\|_F^2 \leq 4\ell_k(\mathbf{B}_k)$. Thus, by taking $\rho = 1/4$, we get

$$\ell_k(\mathbf{B}_k) - \ell_k(\mathbf{H}) \leq 2\|\mathbf{B}_k - \mathbf{H}\|_F^2 - 2\|\mathbf{B}_{k+1} - \mathbf{H}\|_F^2 + \frac{1}{2}\ell_k(\mathbf{B}_k)$$

$$\Rightarrow \qquad \ell_k(\mathbf{B}_k) \leq 4\|\mathbf{B}_k - \mathbf{H}\|_F^2 - 4\|\mathbf{B}_{k+1} - \mathbf{H}\|_F^2 + 2\ell_k(\mathbf{H})$$

Global Superlinear Convergence

- Moreover, we can use a "small-loss" regret bound for Online Gradient Descent

$$\mathbf{B}_{k+1} = \Pi_{\mathcal{Z}}\left(\mathbf{B}_k - \rho \nabla \ell_k(\mathbf{B}_k)\right)$$

  - Standard analysis shows that, for any $\mathbf{H} \in \mathcal{Z}$:

$$\ell_k(\mathbf{B}_k) - \ell_k(\mathbf{H}) \leq \frac{1}{2\rho}\|\mathbf{B}_k - \mathbf{H}\|_F^2 - \frac{1}{2\rho}\|\mathbf{B}_{k+1} - \mathbf{H}\|_F^2 + \frac{\rho}{2}\|\nabla \ell_k(\mathbf{B}_k)\|_F^2$$

  - We also have $\|\nabla \ell_k(\mathbf{B}_k)\|_F^2 \leq 4\ell_k(\mathbf{B}_k)$. Thus, by taking $\rho = 1/4$, we get

$$\begin{aligned} \ell_k(\mathbf{B}_k) - \ell_k(\mathbf{H}) &\leq 2\|\mathbf{B}_k - \mathbf{H}\|_F^2 - 2\|\mathbf{B}_{k+1} - \mathbf{H}\|_F^2 + \frac{1}{2}\ell_k(\mathbf{B}_k) \\ \Rightarrow \qquad \ell_k(\mathbf{B}_k) &\leq 4\|\mathbf{B}_k - \mathbf{H}\|_F^2 - 4\|\mathbf{B}_{k+1} - \mathbf{H}\|_F^2 + 2\ell_k(\mathbf{H}) \end{aligned}$$

  - Summing up the inequalities:

$$\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k) \leq 4\|\mathbf{B}_0 - \mathbf{H}\|_F^2 + 2\sum_{k=0}^{N-1} \ell_k(\mathbf{H})$$

Global Superlinear Convergence

- We showed that for any $\mathbf{H} \in \mathcal{Z} = \{\mathbf{B} : \mu\mathbf{I} \preceq \mathbf{B} \preceq L_1\mathbf{I}\}$:

$$\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k) \leq 4\|\mathbf{B}_0 - \mathbf{H}\|_F^2 + 2\sum_{k=0}^{N-1} \ell_k(\mathbf{H})$$

Global Superlinear Convergence

- We showed that for any $\mathbf{H} \in \mathcal{Z} = \{\mathbf{B} : \mu\mathbf{I} \preceq \mathbf{B} \preceq L_1\mathbf{I}\}$:

$$\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k) \leq 4\|\mathbf{B}_0 - \mathbf{H}\|_F^2 + 2\sum_{k=0}^{N-1} \ell_k(\mathbf{H})$$

- A natural choice: $\mathbf{H} = \nabla^2 f(\boldsymbol{x}^*)$

Global Superlinear Convergence

- We showed that for any $\mathbf{H} \in \mathcal{Z} = \{\mathbf{B} : \mu\mathbf{I} \preceq \mathbf{B} \preceq L_1\mathbf{I}\}$:

$$\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k) \leq 4\|\mathbf{B}_0 - \mathbf{H}\|_F^2 + 2\sum_{k=0}^{N-1} \ell_k(\mathbf{H})$$

- A natural choice: $\mathbf{H} = \nabla^2 f(\boldsymbol{x}^*)$
- We can show that

$$\ell_k(\nabla^2 f(\boldsymbol{x}^*)) = \frac{\|\boldsymbol{y}_k - \nabla^2 f(\boldsymbol{x}^*)\boldsymbol{s}_k\|^2}{\|\boldsymbol{s}_k\|^2} \lesssim \|\nabla^2 f(\boldsymbol{x}_k) - \nabla^2 f(\boldsymbol{x}^*)\|_{\text{op}}^2 \leq L_2^2\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2$$

Global Superlinear Convergence

- We showed that for any $\mathbf{H} \in \mathcal{Z} = \{\mathbf{B} : \mu \mathbf{I} \preceq \mathbf{B} \preceq L_1 \mathbf{I}\}$:

$$\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k) \leq 4\|\mathbf{B}_0 - \mathbf{H}\|_F^2 + 2\sum_{k=0}^{N-1} \ell_k(\mathbf{H})$$

- A natural choice: $\mathbf{H} = \nabla^2 f(\boldsymbol{x}^*)$
- We can show that

$$\ell_k(\nabla^2 f(\boldsymbol{x}^*)) = \frac{\|\boldsymbol{y}_k - \nabla^2 f(\boldsymbol{x}^*)\boldsymbol{s}_k\|^2}{\|\boldsymbol{s}_k\|^2} \lesssim \|\nabla^2 f(\boldsymbol{x}_k) - \nabla^2 f(\boldsymbol{x}^*)\|_{\mathrm{op}}^2 \leq L_2^2\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2$$

- By using $\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 \leq \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 \left(1 + \Omega(\frac{L_1}{\mu})\right)^{-k}$, we further get

$$\sum_{k=0}^{N-1} \ell_k(\nabla^2 f(\boldsymbol{x}^*)) = \mathcal{O}\left(\frac{L_1 L_2^2 \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{\mu}\right)$$

Global Superlinear Convergence

- Putting everything together:

$$\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k) = \mathcal{O}\left(\|\mathbf{B}_0 - \nabla^2 f(\boldsymbol{x}^*)\|_F^2 + \frac{L_1 L_2^2 \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{\mu} \right)$$

  Note that the upper bound is independent of N!

- Thus,

$$\frac{\|\boldsymbol{x}_N - \boldsymbol{x}^*\|^2}{\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2} \leq \left(1 + 2\mu \sqrt{\frac{N}{\mathcal{O}(\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k))}}\right)^{-N} = \left(1 + \frac{\mu}{L_1}\sqrt{\frac{N}{C}}\right)^{-N},$$

  where $C = \mathcal{O}\left(\frac{\|\mathbf{B}_0 - \nabla^2 f(\boldsymbol{x}^*)\|_F^2}{L_1^2} + \frac{L_2^2 \|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{\mu}\right) \Rightarrow$ Worst case: $C = \mathcal{O}(d)$

Projection-Free Online Learning

- One issue: Euclidean projection onto $\mathcal{Z} = \{\mathbf{B} : \mu\mathbf{I} \preceq \mathbf{B} \preceq L_1\mathbf{I}\}$ is expensive
  - It requires full eigen-decomposition, which costs $\mathcal{O}(d^3)$

Projection-Free Online Learning

- One issue: Euclidean projection onto $\mathcal{Z} = \{\mathbf{B} : \mu \mathbf{I} \preceq \mathbf{B} \preceq L_1 \mathbf{I}\}$ is expensive
  - It requires full eigen-decomposition, which costs $\mathcal{O}(d^3)$
- Observation: it is simpler to do "gauge projection" [Mhammedi'22]
  - For a given $\mathbf{B} \in \mathbb{S}^d$, compute λ_{min} and λ_{max}
  - If $\mu \leq \lambda_{\text{min}} \leq \lambda_{\text{max}} \leq L_1$, then $\mathbf{B} \in \mathcal{Z}$
  - Otherwise, we obtain a feasible point $\hat{\mathbf{B}}$ by "pulling" it towards to the "center"

$$\hat{\mathbf{B}} = c\mathbf{B} + (1-c)\frac{L_1 + \mu}{2}\mathbf{I}_d, \quad 0 < c < 1$$

Projection-Free Online Learning

- One issue: Euclidean projection onto $\mathcal{Z} = \{\mathbf{B} : \mu\mathbf{I} \preceq \mathbf{B} \preceq L_1\mathbf{I}\}$ is expensive
  - It requires full eigen-decomposition, which costs $\mathcal{O}(d^3)$
- Observation: it is simpler to do "gauge projection" [Mhammedi'22]
  - For a given $\mathbf{B} \in \mathbb{S}^d$, compute λ_{min} and λ_{max}
  - If $\mu \leq \lambda_{\text{min}} \leq \lambda_{\text{max}} \leq L_1$, then $\mathbf{B} \in \mathcal{Z}$
  - Otherwise, we obtain a feasible point $\hat{\mathbf{B}}$ by "pulling" it towards to the "center"

$$\hat{\mathbf{B}} = c\mathbf{B} + (1-c)\frac{L_1 + \mu}{2}\mathbf{I}_d, \quad 0 < c < 1$$

- Solution: We adopted a projection-free approach inspired by [Mhammedi'22]
- To better illustrate the technique, consider a general online learning problem

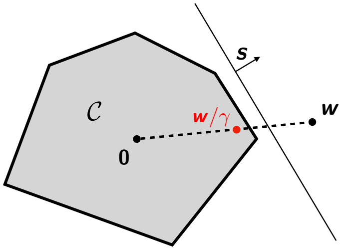
Projection-Free Online Learning

- Consider a standard online learning problem over the constraint set $\mathcal{C}$
- For $k = 0, 1, \ldots, N-1$:
  - Learner chooses $\boldsymbol{x}_k \in \mathcal{C}$
  - Learner observes a convex loss $\ell_k : \mathcal{C} \to \mathbb{R}$
- The goal is to minimize the regret: $\mathrm{Reg}_N(\boldsymbol{x}) = \sum_{k=0}^{N-1} (\ell_k(\boldsymbol{x}_k) - \ell_k(\boldsymbol{x}))$
- The Euclidean projection onto $\mathcal{C}$ can be computationally expensive.
  $\Rightarrow$ But, we have access to a separation oracle

Separation Oracle

- WLOG, we assume that $0 \in \mathcal{C} \subset B_R(0)$. Moreover, we have a separation oracle for $\mathcal{C}$
  - Input: $\boldsymbol{w} \in B_R(0)$
  - Output: $\gamma > 0$, $\boldsymbol{s} \in \mathbb{R}^n$ such that

$$\begin{cases} \gamma \leq 1 \;\Rightarrow\; \boldsymbol{w} \in \mathcal{C}; \\ \gamma > 1 \;\Rightarrow\; \boldsymbol{w}/\gamma \in \mathcal{C} \text{ and } \langle \boldsymbol{s}, \boldsymbol{w} - \boldsymbol{x} \rangle \geq \gamma - 1, \; \forall \boldsymbol{x} \in \mathcal{C} \end{cases}$$



Projection-Free Online Learning

- We introduce an auxiliary online learning problem over the set $B_R(0)$
- For $k = 0, 1, \ldots, N-1$:
  - Learner chooses $\boldsymbol{w}_k \in B_R(0)$
  - Observes $\tilde{\ell}_k(\cdot) = \langle \tilde{\boldsymbol{g}}_k, \cdot \rangle$

Projection-Free Online Learning

- We introduce an auxiliary online learning problem over the set $B_R(0)$
- For $k = 0, 1, \ldots, N-1$:
  - Learner chooses $\boldsymbol{w}_k \in B_R(0)$
  - Observes $\tilde{\ell}_k(\cdot) = \langle \tilde{\boldsymbol{g}}_k, \cdot \rangle$
- We will show that $\sum_{k=0}^{N-1} (\ell_k(\boldsymbol{x}_k) - \ell_k(\boldsymbol{x})) \leq \sum_{k=0}^{N-1} \langle \tilde{\boldsymbol{g}}_k, \boldsymbol{w}_k - \boldsymbol{x} \rangle, \quad \forall \boldsymbol{x} \in \mathcal{C}$

  $\Rightarrow$ It suffices to bound the regret of the auxiliary problem

Projection-Free Online Learning

- We introduce an auxiliary online learning problem over the set $B_R(0)$
- For $k = 0, 1, \ldots, N-1$:
  - Learner chooses $\boldsymbol{w}_k \in B_R(0)$
  - Observes $\tilde{\ell}_k(\cdot) = \langle \tilde{\boldsymbol{g}}_k, \cdot \rangle$
- We will show that $\sum_{k=0}^{N-1}(\ell_k(\boldsymbol{x}_k) - \ell_k(\boldsymbol{x})) \leq \sum_{k=0}^{N-1}\langle \tilde{\boldsymbol{g}}_k, \boldsymbol{w}_k - \boldsymbol{x} \rangle, \quad \forall \boldsymbol{x} \in \mathcal{C}$
  $\Rightarrow$ It suffices to bound the regret of the auxiliary problem
- It is simple to compute the Euclidean projection onto the set $B_R(0)$
  $\Rightarrow$ We can use Online Gradient Descent: $\boldsymbol{w}_{k+1} = \Pi_{B_R(0)}(\boldsymbol{w}_k - \rho \tilde{\boldsymbol{g}}_k)$

Projection-Free Online Learning

- We introduce an auxiliary online learning problem over the set $B_R(0)$
- For $k = 0, 1, \ldots, N-1$:
  - Learner chooses $\boldsymbol{w}_k \in B_R(0)$
  - Observes $\tilde{\ell}_k(\cdot) = \langle \tilde{\boldsymbol{g}}_k, \cdot \rangle$
- We will show that $\sum_{k=0}^{N-1}(\ell_k(\boldsymbol{x}_k) - \ell_k(\boldsymbol{x})) \leq \sum_{k=0}^{N-1}\langle \tilde{\boldsymbol{g}}_k, \boldsymbol{w}_k - \boldsymbol{x}\rangle, \quad \forall \boldsymbol{x} \in \mathcal{C}$
  $\Rightarrow$ It suffices to bound the regret of the auxiliary problem
- It is simple to compute the Euclidean projection onto the set $B_R(0)$
  $\Rightarrow$ We can use Online Gradient Descent: $\boldsymbol{w}_{k+1} = \Pi_{B_R(0)}(\boldsymbol{w}_k - \rho\tilde{\boldsymbol{g}}_k)$
- Question:
  - How to generate the "actual" iterate $\boldsymbol{x}_k \in \mathcal{C}$?
  - How to define the surrogate loss vector $\tilde{\boldsymbol{g}}_k$?

  We will use the separation oracle!

Projection-Free Online Learning

- Initialize $\boldsymbol{w}_0 = \boldsymbol{x}_0 \in \mathcal{C}$ and $\tilde{\boldsymbol{g}}_0 \leftarrow \nabla \ell_0(\boldsymbol{x}_0)$
- For $k = 0, 1, \ldots, N-1$:
  - Update $\boldsymbol{w}_{k+1} \leftarrow \Pi_{B_R(0)}(\boldsymbol{w}_k - \rho \tilde{\boldsymbol{g}}_k)$
  - Let $(\gamma_{k+1}, \boldsymbol{s}_{k+1}) \leftarrow \mathrm{SEP}(\boldsymbol{w}_{k+1})$
  - We consider two cases:

$$\begin{cases} \text{If } \gamma_{k+1} \leq 1: & \text{set } \boldsymbol{x}_{k+1} \leftarrow \boldsymbol{w}_{k+1},\ \tilde{\boldsymbol{g}}_{k+1} \leftarrow \boldsymbol{g}_{k+1} \\ \text{If } \gamma_{k+1} > 1: & \text{set } \boldsymbol{x}_{k+1} \leftarrow \frac{\boldsymbol{w}_{k+1}}{\gamma_{k+1}},\ \ \tilde{\boldsymbol{g}}_{k+1} \leftarrow \boldsymbol{g}_{k+1} + |\langle \boldsymbol{g}_{k+1}, \boldsymbol{x}_{k+1} \rangle| \boldsymbol{s}_{k+1} \end{cases}$$

where $\boldsymbol{g}_{k+1} = \nabla \ell_{k+1}(\boldsymbol{x}_{k+1})$

$$\sum_{k=0}^{N-1} (\ell_k(\boldsymbol{x}_k) - \ell_k(\boldsymbol{x})) \leq \sum_{k=0}^{N-1} \langle \tilde{\boldsymbol{g}}_k, \boldsymbol{w}_k - \boldsymbol{x} \rangle, \quad \forall \boldsymbol{x} \in \mathcal{C}$$

Separation Oracle in Our Setting

- Recall that in our case, the constraint set is $\mathcal{Z} = \{\mathbf{B} : \mu\mathbf{I} \preceq \mathbf{B} \preceq L_1\mathbf{I}\}$
- By translation and rescaling, we work with $\mathcal{C} \triangleq \{\hat{\mathbf{B}} : -\mathbf{I} \preceq \hat{\mathbf{B}} \preceq \mathbf{I}\} = \{\hat{\mathbf{B}} : \|\hat{\mathbf{B}}\|_{\mathrm{op}} \leq 1\}$

- Input: $\boldsymbol{w} \in B_R(0)$
- Output: $\gamma > 0$, $\boldsymbol{s} \in \mathbb{R}^n$ such that

$$\begin{cases} \gamma \leq 1 \;\Rightarrow\; \boldsymbol{w} \in \mathcal{C}; \\ \gamma > 1 \;\Rightarrow\; \begin{array}{l} \boldsymbol{w}/\gamma \in \mathcal{C}, \\ \langle \boldsymbol{s}, \boldsymbol{w} - \boldsymbol{x} \rangle \geq \gamma - 1, \ \forall \boldsymbol{x} \in \mathcal{C} \end{array} \end{cases}$$

- Input: $\mathbf{W} \in \mathbb{S}^d$ satisfying $\|\mathbf{W}\|_F \leq \sqrt{d}$
- Output: $\gamma > 0$, $\mathbf{S} \in \mathbb{S}^d$ such that

$$\begin{cases} \gamma \leq 1 \;\Rightarrow\; \|\mathbf{W}\|_{\mathrm{op}} \leq 1; \\ \gamma > 1 \;\Rightarrow\; \begin{array}{l} \|\mathbf{W}/\gamma\|_{\mathrm{op}} \leq 1, \\ \langle \mathbf{S}, \mathbf{W} - \hat{\mathbf{B}} \rangle \geq \gamma - 1, \ \forall \|\hat{\mathbf{B}}\|_{\mathrm{op}} \leq 1 \end{array} \end{cases}$$

- We only need to approximately compute $(\lambda_{\min}, \boldsymbol{v}_{\min})$ and $(\lambda_{\max}, \boldsymbol{v}_{\max})$ of $\mathbf{W}$
- We rely on the Lanczos method with a random start [Kuczyński-Woźniakowski'92]

Summary of Convergence Rates (Strongly Convex)

Theorem: [Jiang-Jin-M, *COLT*, '23]

Assume that $\mu \mathbf{I} \preceq \nabla^2 f(\mathbf{x}) \preceq L_1 \mathbf{I}$ *and* $\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})\| \leq L_2 \|\mathbf{x} - \mathbf{y}\|$. *Then*

(a) *(Linear convergence) For any* $k \geq 0$, *we have* $\frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|^2}{\|\mathbf{x}_k - \mathbf{x}^*\|^2} \leq \left(1 + \frac{\mu}{4L_1}\right)^{-1}$.

(b) *(Superlinear convergence) For any* $k \geq 0$,

$$\frac{\|\mathbf{x}_k - \mathbf{x}^*\|^2}{\|\mathbf{x}_0 - \mathbf{x}^*\|^2} \leq \left(1 + \frac{\mu}{L_1}\sqrt{\frac{k}{C}}\right)^{-k} \approx \mathcal{O}\left(\left(\frac{1}{\sqrt{k}}\right)^k\right)$$

where $C = \mathcal{O}(\|\mathbf{B}_0 - \nabla^2 f(\mathbf{x}^*)\|_F^2 / L_1^2 + L_2^2 \|\mathbf{x}_0 - \mathbf{x}^*\|^2 / (L_1 \mu)) \approx \mathcal{O}(d)$

- As a corollary, the number of iterations to reach ϵ-accuracy can be bounded by

$$N_\epsilon = \begin{cases} \frac{L_1}{\mu} \log \frac{1}{\epsilon}, & \text{if } \epsilon > \exp(-\frac{\mu}{L_1} C) \\ \frac{\log \frac{1}{\epsilon}}{\log \log \frac{1}{\epsilon}}, & \text{if } \epsilon \ll \exp(-\frac{\mu}{L_1} C) \end{cases}$$

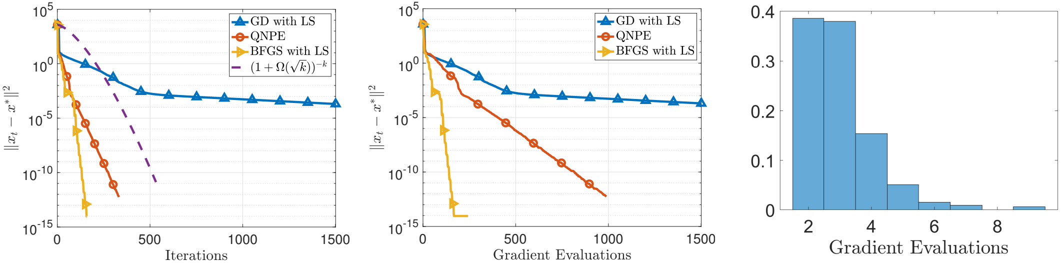
Computational Cost

Lemma: [Jiang-Jin-M, *COLT*, '23]

Let N_ϵ be the number of iterations to reach ϵ-accuracy. Then, the total number of gradient computations (due to BTLS) is bounded above by $3N_\epsilon$.

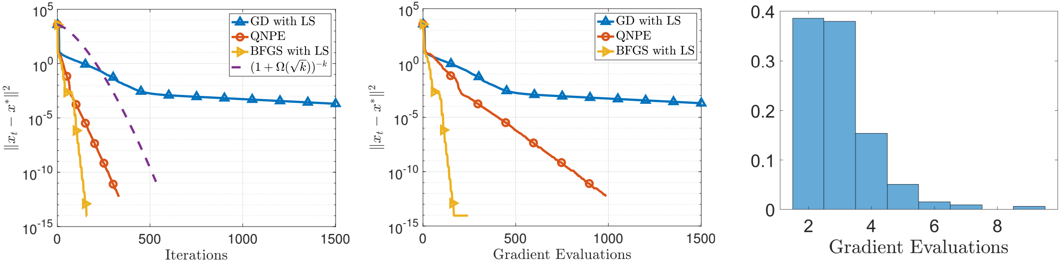
- Iteration and gradient complexity: $N_\epsilon = \mathcal{O}\left(\frac{\log \frac{1}{\epsilon}}{\log\log \frac{1}{\epsilon}}\right)$
- Matrix-vector products:
  - $\mathcal{O}\left(N_\epsilon \sqrt{\frac{L_1}{\mu}} \log\left(\frac{L_1\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{\mu\epsilon}\right)\right)$ from approx. linear system solving
  - $\mathcal{O}\left(N_\epsilon \sqrt{\frac{L_1}{\mu}} \log\left(dN_\epsilon^2\right)\right)$ from approx. eigenvector computation
- Total number of Matrix-vector products $\tilde{\mathcal{O}}(N_\epsilon\sqrt{\kappa})$

Numerical Experiment



(a) Convergence by iteration (b) Convergence by gradient evals (c) Histogram of gradient evals

Figure: Numerical results for an L_2-regularized logistic regression problem



(a) Convergence by iteration (b) Convergence by gradient evals (c) Histogram of gradient evals

Figure: Numerical results for an L_2-regularized logistic regression problem

What about the CVX setting?

Accelerated Hybrid Proximal Extragradient (MS Acceleration)

- Our proposed method is based on the accelerated HPE framework [Monteiro-Svaiter'13]
- Initialization: $\boldsymbol{x}_0, \boldsymbol{z}_0 \in \mathbb{R}^d$ and $A_0 = 0$
- Stage 1: Pick η_k and compute

$$a_k = \frac{\eta_k + \sqrt{\eta_k^2 + 4\eta_k A_k}}{2}, \qquad \boldsymbol{y}_k = \frac{A_k}{A_k + a_k}\boldsymbol{x}_k + \frac{a_k}{A_k + a_k}\boldsymbol{z}_k$$

- Stage 2: Inexact proximal point update

$$\boldsymbol{x}_{k+1} \approx \boldsymbol{y}_k - \eta_k \nabla f(\boldsymbol{x}_{k+1})$$

- Stage 3: Extragradient step

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k - a_k \nabla f(\boldsymbol{x}_{k+1}), \quad A_{k+1} = A_k + a_k$$

- $f(\boldsymbol{x}_N) - f(\boldsymbol{x}^*) \leq 2\|\boldsymbol{z}_0 - \boldsymbol{x}^*\|^2 / (\sum_{k=0}^{N-1} \sqrt{\eta_k})^2 \Rightarrow$ any rate can be achieved as $\eta_k \uparrow$

Accelerated Newton Proximal Extragradient

- Issue: Subproblem in Stage 2 is costly!

$$\|\mathbf{x}_{k+1} - \mathbf{y}_k + \eta_k \nabla f(\mathbf{x}_{k+1})\| \leq \frac{1}{2}\|\mathbf{x}_{k+1} - \mathbf{y}_k\|$$

- Solution: Linearize $\nabla f(\mathbf{x}_{k+1}) \approx \nabla f(\mathbf{y}_k) + \nabla^2 f(\mathbf{y}_k)(\mathbf{x}_{k+1} - \mathbf{y}_k)$
- Stage 2: Newton Proximal Step [Monteiro-Svaiter'13]

$$\|\mathbf{x}_{k+1} - \mathbf{y}_k + \eta_k(\nabla f(\mathbf{y}_k) + \nabla^2 f(\mathbf{y}_k)(\mathbf{x}_{k+1} - \mathbf{y}_k))\| \leq \frac{1}{4}\|\mathbf{x}_{k+1} - \mathbf{y}_k\|,$$

$$\eta_k\|\nabla f(\mathbf{x}_{k+1}) - (\nabla f(\mathbf{y}_k) + \nabla^2 f(\mathbf{y}_k)(\mathbf{x}_{k+1} - \mathbf{y}_k))\| \leq \frac{1}{4}\|\mathbf{x}_{k+1} - \mathbf{y}_k\|$$

- However, there is another issue: η_k appears in both Stage 1 and 2
  $\Rightarrow$ The line search procedure is much more complicated
- In the paper, we adopt a refined MS acceleration framework by [Carmon et al.'22]

Accelerated Quasi-Newton Proximal Extragradient

- Stage 1: Pick η_k and compute

$$a_k = \frac{\eta_k + \sqrt{\eta_k^2 + 4\eta_k A_k}}{2}, \qquad \mathbf{y}_k = \frac{A_k}{A_k + a_k}\mathbf{x}_k + \frac{a_k}{A_k + a_k}\mathbf{z}_k$$

- Stage 2: Quasi-Newton proximal step

$$\|\mathbf{x}_{k+1} - \mathbf{y}_k + \eta_k(\nabla f(\mathbf{y}_k) + \mathbf{B}_k(\mathbf{x}_{k+1} - \mathbf{y}_k))\| \leq \frac{1}{4}\|\mathbf{x}_{k+1} - \mathbf{y}_k\|,$$

$$\eta_k\|\nabla f(\mathbf{x}_{k+1}) - (\nabla f(\mathbf{y}_k) + \mathbf{B}_k(\mathbf{x}_{k+1} - \mathbf{y}_k))\| \leq \frac{1}{4}\|\mathbf{x}_{k+1} - \mathbf{y}_k\|$$

  - Given $\mathbf{y}_k$ and $\mathbf{B}_k$, use backtracking line search to find η_k and $\mathbf{x}_{k+1}$

- Stage 3: Extragradient step

$$\mathbf{z}_{k+1} = \mathbf{z}_k - a_k\nabla f(\mathbf{x}_{k+1}), \quad A_{k+1} = A_k + a_k$$

How to Update $\mathbf{B}_k$: Starting from Convergence Analysis

- How should we select/update $\{\mathbf{B}_k\}$?

How to Update $\mathbf{B}_k$: Starting from Convergence Analysis

- How should we select/update $\{\mathbf{B}_k\}$?
- Same story: we let the convergence analysis guide our choice of $\mathbf{B}_k$!

How to Update $\mathbf{B}_k$: Starting from Convergence Analysis

- How should we select/update $\{\mathbf{B}_k\}$?
- Same story: we let the convergence analysis guide our choice of $\mathbf{B}_k$!
- We know that $f(\boldsymbol{x}_N) - f(\boldsymbol{x}^*) \leq 2\|\boldsymbol{z}_0 - \boldsymbol{x}^*\|^2/(\sum_{k=0}^{N-1}\sqrt{\eta_k})^2$

How to Update $\mathbf{B}_k$: Starting from Convergence Analysis

- How should we select/update $\{\mathbf{B}_k\}$?
- Same story: we let the convergence analysis guide our choice of $\mathbf{B}_k$!
- We know that $f(\boldsymbol{x}_N) - f(\boldsymbol{x}^*) \leq 2\|\boldsymbol{z}_0 - \boldsymbol{x}^*\|^2/(\sum_{k=0}^{N-1}\sqrt{\eta_k})^2$
- η_k is constrained by

$$\eta_k\|\nabla f(\boldsymbol{x}_{k+1}) - (\nabla f(\boldsymbol{y}_k) + \mathbf{B}_k(\boldsymbol{x}_{k+1} - \boldsymbol{y}_k))\| \leq \frac{1}{4}\|\boldsymbol{x}_{k+1} - \boldsymbol{y}_k\|$$

- **Initial result:** By backtracking line search:

$$\eta_k \simeq \frac{\|\boldsymbol{x}_{k+1} - \boldsymbol{y}_k\|}{\|\nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{y}_k) - \mathbf{B}_k(\boldsymbol{x}_{k+1} - \boldsymbol{y}_k)\|} = \frac{\|\boldsymbol{s}_k\|}{\|\boldsymbol{w}_k - \mathbf{B}_k\boldsymbol{s}_k\|},$$

where $\boldsymbol{w}_k = \nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{y}_k)$ and $\boldsymbol{s}_k = \boldsymbol{x}_{k+1} - \boldsymbol{y}_k$

How to Update $\mathbf{B}_k$: Starting from Convergence Analysis

- After N iterations, we have

$$f(\boldsymbol{x}_N) - f(\boldsymbol{x}^*) \le \frac{2\|\boldsymbol{z}_0 - \boldsymbol{x}^*\|^2}{(\sum_{k=0}^{N-1} \sqrt{\eta_k})^2} \le \frac{2\|\boldsymbol{z}_0 - \boldsymbol{x}^*\|^2}{N^{2.5}} \sqrt{\sum_{k=0}^{N-1} \frac{1}{\eta_k^2}}$$

- Since $\eta_k \simeq \|\boldsymbol{s}_k\| / \|\boldsymbol{w}_k - \mathbf{B}_k \boldsymbol{s}_k\|$, we further have

$$f(\boldsymbol{x}_N) - f(\boldsymbol{x}^*) \le \frac{2\|\boldsymbol{z}_0 - \boldsymbol{x}^*\|^2}{N^{2.5}} \sqrt{\mathcal{O}\left(\sum_{k=0}^{N-1} \frac{\|\boldsymbol{w}_k - \mathbf{B}_k \boldsymbol{s}_k\|^2}{\|\boldsymbol{s}_k\|^2}\right)}$$

- Hence, the goal is to choose $\mathbf{B}_k$ such that we minimize

$$\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k) \; := \; \sum_{k=0}^{N-1} \frac{\|\boldsymbol{w}_k - \mathbf{B}_k \boldsymbol{s}_k\|^2}{\|\boldsymbol{s}_k\|^2}$$

Hessian Approximation Update via Online Learning

- Again, we view the update of $\mathbf{B}_k$ as an online convex opt problem
  - Choose $\mathbf{B}_k \in \mathcal{Z}$, where $\mathcal{Z} = \{\mathbf{B} : 0 \preceq \mathbf{B} \preceq L_1 \mathbf{I}\}$
  - Receive $\ell_k(\mathbf{B}_k)$
  - Update $\mathbf{B}_{k+1}$ by an online learning algorithm, e.g., Online Gradient Descent

$$\mathbf{B}_{k+1} = \Pi_{\mathcal{Z}} \left(\mathbf{B}_k - \rho \nabla \ell_k(\mathbf{B}_k)\right)$$

- To avoid Euclidean projection, we use the same projection-free online learning approach

$\mathcal{O}(1/k^2)$ Convergence Rate

- Now our goal is to upper bound $\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k)$
- Using $0 \preceq \mathbf{B}_k \preceq L_1 \mathbf{I}$, we can always have $\ell_k(\mathbf{B}_k) \leq 4L_1^2 \quad \Rightarrow \quad \sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k) \leq 4L_1^2 N$
- Plugging this bound back:

$$f(\boldsymbol{x}_N) - f(\boldsymbol{x}^*) \leq \frac{2\|\boldsymbol{z}_0 - \boldsymbol{x}^*\|^2}{N^{2.5}} \sqrt{\mathcal{O}\left(\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k)\right)} = \mathcal{O}\left(\frac{L_1\|\boldsymbol{z}_0 - \boldsymbol{x}^*\|^2}{N^2}\right)$$

- This matches the rate of Nesterov's Accelerated Gradient

$\tilde{\mathcal{O}}(\sqrt{d}/k^{2.5})$ Convergence Rate

- Recall that in the strongly convex setting, a better bound on $\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k)$ can be obtained using regret analysis
  - For any $\mathbf{H} \in \mathcal{Z} = \{\mathbf{B} : 0 \preceq \mathbf{B} \preceq L_1 \mathbf{I}\}$, we have

$$\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k) \leq 4\|\mathbf{B}_0 - \mathbf{H}\|_F^2 + 2\sum_{k=0}^{N-1} \ell_k(\mathbf{H})$$

  - Choosing $\mathbf{H} = \nabla^2 f(\boldsymbol{x}^*)$, we showed that $\ell_k(\nabla^2 f(\boldsymbol{x}^*)) \lesssim L_2^2 \|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2$
  - By linear convergence, $\sum_{k=0}^{N-1} \|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 = \mathcal{O}\left(\frac{L_1}{\mu}\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2\right)$
- However, we do not have linear convergence in the convex setting!
- Have to take a different approach via dynamic regret bound

$\tilde{\mathcal{O}}(\sqrt{d}/k^{2.5})$ Convergence Rate

- For any sequence $\{\mathbf{H}_k\}_{k=0}^{N-1}$ with $\mathbf{H}_k \in \mathcal{Z} = \{\mathbf{B} : 0 \preceq \mathbf{B} \preceq L_1\mathbf{I}\}$, we can show that

$$\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k) = \mathcal{O}\left(\|\mathbf{B}_0 - \mathbf{H}_0\|_F^2 + \sum_{k=0}^{N-1} \ell_k(\mathbf{H}_k) + L_1\sqrt{d}\sum_{k=0}^{N-1} \|\mathbf{H}_{k+1} - \mathbf{H}_k\|_F \right)$$

- We then choose $\mathbf{H}_k = \nabla^2 f(\boldsymbol{y}_k)$ for $k = 0, \dots, N-1$
- We can show that

$$\ell_k(\nabla^2 f(\boldsymbol{y}_k)) \le L_2^2 \|\boldsymbol{x}_{k+1} - \boldsymbol{y}_k\|^2,$$
$$\|\nabla^2 f(\boldsymbol{y}_{k+1}) - \nabla^2 f(\boldsymbol{y}_k)\|_F \le \sqrt{d}\|\nabla^2 f(\boldsymbol{y}_{k+1}) - \nabla^2 f(\boldsymbol{y}_k)\|_{\mathrm{op}} \le \sqrt{d} L_2 \|\boldsymbol{y}_{k+1} - \boldsymbol{y}_k\|$$

- With some careful analysis, we can bound

$$\sum_{k=0}^{N-1} \|\boldsymbol{x}_{k+1} - \boldsymbol{y}_k\|^2 = \mathcal{O}\left(\|\boldsymbol{z}_0 - \boldsymbol{x}^*\|^2\right), \qquad \sum_{k=0}^{N-1} \|\boldsymbol{y}_{k+1} - \boldsymbol{y}_k\| = \mathcal{O}\left(\log N \|\boldsymbol{z}_0 - \boldsymbol{x}^*\|\right)$$

$\tilde{\mathcal{O}}(\sqrt{d}/k^{2.5})$ Convergence Rate

- Putting everything together:

$$\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k) = \mathcal{O}\left(\|\mathbf{B}_0 - \mathbf{H}_0\|_F^2 + \sum_{k=0}^{N-1} \ell_k(\mathbf{H}_k) + L_1\sqrt{d}\sum_{k=0}^{N-1} \|\mathbf{H}_{k+1} - \mathbf{H}_k\|_F\right)$$
$$= \mathcal{O}\left(\|\mathbf{B}_0 - \nabla^2 f(\boldsymbol{x}_0)\|_F^2 + L_2^2\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 + L_1 L_2 d\|\boldsymbol{z}_0 - \boldsymbol{x}^*\| \log N\right)$$

- Thus, we have

$$f(\boldsymbol{x}_N) - f(\boldsymbol{x}^*) \leq \frac{2\|\boldsymbol{z}_0 - \boldsymbol{x}^*\|^2}{N^{2.5}}\sqrt{\mathcal{O}\left(\sum_{k=0}^{N-1} \ell_k(\mathbf{B}_k)\right)} = \mathcal{O}\left(\frac{\sqrt{d}\log N}{N^{2.5}}\right)$$

Summary of our results for the Convex setting

Theorem: [Jiang-Jin-M, *NeurIPS*, '23]

Assume that $\mathbf{0} \preceq \nabla^2 f(\boldsymbol{x}) \preceq L_1 \mathbf{I}$ *and* $\|\nabla^2 f(\boldsymbol{x}) - \nabla^2 f(\boldsymbol{y})\| \leq L_2 \|\boldsymbol{x} - \boldsymbol{y}\|$. *Then the iterates if AQNPE satisfy*

$$f(\boldsymbol{x}_k) - f(\boldsymbol{x}) \leq \mathcal{O}\left(\min\left\{\frac{1}{k^2}, \frac{\sqrt{d \log k}}{k^{2.5}}\right\}\right)$$

- Iteration complexity: $N_\epsilon = \tilde{\mathcal{O}}(\min\{\frac{1}{\sqrt{\epsilon}}, \frac{d^{0.2}}{\epsilon^{0.4}}\})$
- Side result: Gradient evaluations: $3N_\epsilon$
- Hence, gradient complexity: $N_\epsilon = \tilde{\mathcal{O}}(\min\{\frac{1}{\sqrt{\epsilon}}, \frac{d^{0.2}}{\epsilon^{0.4}}\})$

Computational Cost

- Matrix-vector products:
  - $\mathcal{O}\left(N_\epsilon + \sqrt{\frac{1}{\epsilon}}\right)$ from approx. linear system solving
  - $\tilde{\mathcal{O}}\left(N_\epsilon^{1.25}\right)$ from approx. eigenvector computation

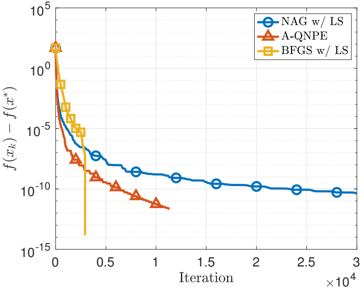
Methods	Gradient queires	Additional Matrix-vector products
AGD	$\mathcal{O}(\frac{1}{\sqrt{\epsilon}})$	N.A.
A-QNPE (ours)	$\tilde{\mathcal{O}}(\min\{\frac{1}{\sqrt{\epsilon}}, \frac{d^{0.2}}{\epsilon^{0.4}}\})$	$\tilde{\mathcal{O}}(\min\{\frac{1}{\epsilon^{0.625}}, \frac{d^{0.25}}{\epsilon^{0.5}}\})$

Computational Cost

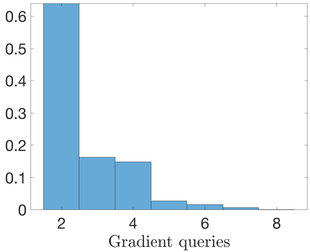
- Matrix-vector products:
  - $\mathcal{O}\left(N_\epsilon + \sqrt{\frac{1}{\epsilon}}\right)$ from approx. linear system solving
  - $\tilde{\mathcal{O}}\left(N_\epsilon^{1.25}\right)$ from approx. eigenvector computation

Methods	Gradient queires	Additional Matrix-vector products
AGD	$\mathcal{O}(\frac{1}{\sqrt{\epsilon}})$	N.A.
A-QNPE (ours)	$\tilde{\mathcal{O}}(\min\{\frac{1}{\sqrt{\epsilon}}, \frac{d^{0.2}}{\epsilon^{0.4}}\})$	$\tilde{\mathcal{O}}(\min\{\frac{1}{\epsilon^{0.625}}, \frac{d^{0.25}}{\epsilon^{0.5}}\})$

- Gradient complexity of AQNPE is always better than AGD
- Overall complexity is better when gradient query is more costly than mat-vec product



(a) Convergence by iteration



(b) Histogram of gradient evals

Figure: Numerical results for log-sum-exp function on a synthetic dataset.

References

- R. Jiang, Q. Jin, A. Mokhtari, "Online Learning Guided Curvature Approximation: A Quasi-Newton Method with Global Non-Asymptotic Superlinear Convergence," COLT 2023. arXiv: 2302.08580 [math.OC]
- R. Jiang, A. Mokhtari, "Accelerated Quasi-Newton Proximal Extragradient: Faster Rate for Smooth Convex Optimization," NeurIPS 2023 (Spotlight). arXiv: 2306.02212 [math.OC]

Thank you!

Quasi-Newton Proximal Extragradient

Initialization: initial point $\boldsymbol{x}_0 \in \mathbb{R}^d$ and initial $\mathbf{B}_0$ s.t. $\mu\mathbf{I} \preceq \mathbf{B}_0 \preceq L_1\mathbf{I}$

for iteration $k = 0, \ldots, N-1$ **do**

  Let η_k be the largest possible step size in $\{\sigma_k \beta^i : i \geq 0\}$ such that

$$\hat{\boldsymbol{x}}_k \approx_{\alpha_1} \boldsymbol{x}_k - \eta_k(\mathbf{I} + \eta_k \mathbf{B}_k)^{-1}\nabla f(\boldsymbol{x}_k),$$
$$\eta_k\|\nabla f(\hat{\boldsymbol{x}}_k) - \nabla f(\boldsymbol{x}_k) - \mathbf{B}_k(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)\| \leq \alpha_2\|\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k\|.$$

  Set $\sigma_{k+1} \leftarrow \eta_k/\beta$

  Update $\boldsymbol{x}_{k+1} \leftarrow \frac{1}{1+2\eta_k\mu}(\boldsymbol{x}_k - \eta_k\nabla f(\hat{\boldsymbol{x}}_k)) + \frac{2\eta_k\mu}{1+2\eta_k\mu}\hat{\boldsymbol{x}}_k$

  if $\eta_k = \sigma_k$ **then** *# Line search accepted the initial trial step size*

    Set $\mathbf{B}_{k+1} \leftarrow \mathbf{B}_k$

  else *# Line search bactracked*

    Let $\tilde{\boldsymbol{x}}_k$ be the last rejected iterate in the line search

    Set $\boldsymbol{y}_k \leftarrow \nabla f(\tilde{\boldsymbol{x}}_k) - \nabla f(\boldsymbol{x}_k)$ and $\boldsymbol{s}_k \leftarrow \tilde{\boldsymbol{x}}_k - \boldsymbol{x}_k$

    Define the loss function $\ell_k(\mathbf{B}) = \frac{\|\boldsymbol{y}_k - \mathbf{B}\boldsymbol{s}_k\|^2}{2\|\boldsymbol{s}_k\|^2}$

    Feed $\ell_k(\mathbf{B})$ to an online learning algorithm and obtain $\mathbf{B}_{k+1}$

  end if

end for