University of Amsterdam
Korteweg de Vries Institute for Mathematics
Universiteit Leiden
Mathematical Institute
Google DeepMind

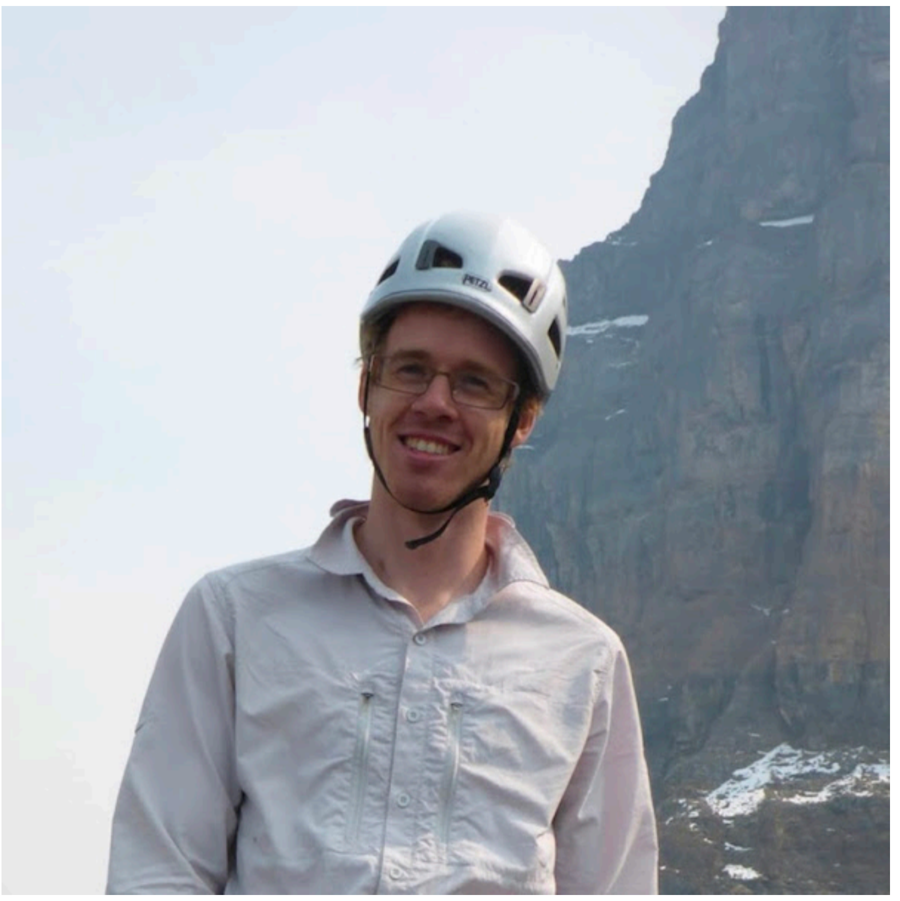
A Newton Method for Bandit Convex Optimisation

2024-07-01

Joint Work

- All work presented was created in collaboration with:

Jack Mayo
University of Amsterdam



Tor Lattimore
DeepMind

Dirk van der Hoeven
Leiden University

Outline

- Introduction to Bandit Convex Optimisation
- Regret Result
- Convex Extension, Bandit Version
- Gaussian Optimistic Smoothing
- Algorithm

Bandit Convex Optimisation

An introduction

Bandit Convex Optimisation

General setting

Let:

- $K \subseteq \mathbb{R}^d$, convex body
- Convex $\ell_1, \ldots, \ell_n : K \to [0,1]$
- In each round $t = 1, \ldots, n$:
  - Learner chooses action $A_t \in K$
  - Suffers loss $\ell_t(A_t)$
  - Observes $Y_t = \ell_t(A_t) + \varepsilon_t$
  - ε_t conditionally Sub-Gaussian and mean 0.

Regret is measured as:

$$\text{Reg}_n = \sum_{t=1}^{n} \ell_t(A_t) - \min_{x \in K} \sum_{t=1}^{n} \ell_t(x)$$

Bandit Convex Optimisation

Adversarial setting

Let:

- $K \subseteq \mathbb{R}^d$, with $\mathbb{B}(1) \subseteq K \subseteq 2\mathbb{B}(d+1)$
- Convex $\ell_1, \ldots, \ell_n : K \to [0,1]$
- In each round $t = 1, \ldots, n$:
  - Learner chooses action $A_t \in K$
  - Suffers loss $\ell_t(A_t)$
  - Observes $Y_t = \ell_t(A_t) + \varepsilon_t$
  - ε_t conditionally Sub-Gaussian and mean 0.
  - Or $\varepsilon_t = 0$ for all $t = 1, \ldots, n$

Regret is measured as:

$$\mathrm{Reg}_n = \sum_{t=1}^{n} \ell_t(A_t) - \min_{x \in K} \sum_{t=1}^{n} \ell_t(x)$$

Bandit Convex Optimisation

Stochastic setting

Let:

- $K \subseteq \mathbb{R}^d$, with $\mathbb{B}(1) \subseteq K \subseteq 2\mathbb{B}(d+1)$
- Convex $\ell : K \to [0,1]$
- In each round $t = 1,\ldots,n$:
  - Learner chooses action $A_t \in K$
  - Suffers loss $\ell(A_t)$
  - Observes $Y_t = \ell(A_t) + \varepsilon_t$
  - ε_t conditionally Sub-Gaussian and mean 0

Regret is measured as:

$$\mathrm{Reg}_n = \sum_{t=1}^{n} \ell(A_t) - \min_{x \in K} \sum_{t=1}^{n} \ell(x)$$

Regret
With some previous results

Previous Regret Results

Paper	Assumptions	Regret Stochastic	Regret Adversarial	Running Time
[Flaxman et al., 2005]	bounded convex, Lipschitz	$\tilde{O}(\sqrt{d}n^{\frac{3}{4}})$	$\tilde{O}(\sqrt{d}n^{\frac{3}{4}})$	$O(d)$
[Abernethy et al., 2009]	linear	$\tilde{O}(d\sqrt{n})$	$\tilde{O}(d\sqrt{n})$	$O(d^2)$
[Hazan and Levy, 2014]	strongly convex, smooth	$\tilde{O}(d\sqrt{\left(\vartheta + \frac{\beta}{\alpha}\right) n}$	$\tilde{O}(d\sqrt{\left(\vartheta + \frac{\beta}{\alpha}\right) n}$	$O(d)$
[Suggala et al. 2021]	convex quadratic	$\tilde{O}(d^{16}\sqrt{n})$	$\tilde{O}(d^{16}\sqrt{n})$	$O(d^4)$
[Bubeck et al., 2017]	bounded convex	$\tilde{O}(d^{10.5}\sqrt{n})$	$\tilde{O}(d^{10.5}\sqrt{n})$	poly(d, T)
[Lattimore, 2020]	convex	$\tilde{O}(d^{2.5}\sqrt{n})$	$\tilde{O}(d^{2.5}\sqrt{n})$	$\exp(d, T)$
[Lattimore and Gyorgy, 2021]	convex	$\tilde{O}(d^{4.5}\sqrt{n})$	$\times$	poly(d)
[Lattimore and Gyorgy, 2023]	Lipschitz convex, unconstrained	$\tilde{O}(d^{1.5}\sqrt{n})$	$\times$	$O(d^3)$

Previous Regret Results

Paper	Assumptions	Regret Stochastic	Regret Adversarial	Running Time
[Flaxman et al., 2005]	bounded convex, Lipschitz	$\tilde{O}(\sqrt{d}n^{\frac{3}{4}})$	$\tilde{O}(\sqrt{d}n^{\frac{3}{4}})$	$O(d)$
[Abernethy et al., 2009]	linear	$\tilde{O}(d\sqrt{n})$	$\tilde{O}(d\sqrt{n})$	$O(d^2)$
[Hazan and Levy, 2014]	strongly convex, smooth	$\tilde{O}(d\sqrt{\left(\vartheta + \frac{\beta}{\alpha}\right) n}$	$\tilde{O}(d\sqrt{\left(\vartheta + \frac{\beta}{\alpha}\right) n}$	$O(d)$
[Suggala et al. 2021]	convex quadratic	$\tilde{O}(d^{16}\sqrt{n})$	$\tilde{O}(d^{16}\sqrt{n})$	$O(d^4)$
[Bubeck et al., 2017]	bounded convex	$\tilde{O}(d^{10.5}\sqrt{n})$	$\tilde{O}(d^{10.5}\sqrt{n})$	poly(d, T)
[Lattimore, 2020]	convex	$\tilde{O}(d^{2.5}\sqrt{n})$	$\tilde{O}(d^{2.5}\sqrt{n})$	$\exp(d, T)$
[Lattimore and Gyorgy, 2021]	convex	$\tilde{O}(d^{4.5}\sqrt{n})$	$\times$	poly(d)
[Lattimore and Gyorgy, 2023]	Lipschitz convex, unconstrained	$\tilde{O}(d^{1.5}\sqrt{n})$	$\times$	$O(d^3)$
[Suggala et al., 2024]	κ-convex, bounded (gradients)	$\tilde{O}(d^{2.5}\kappa^2\sqrt{n})$	$\tilde{O}(d^{2.5}\kappa^2\sqrt{n})$	$O(d^2)$
Ours	Bounded Convex	$\tilde{O}([d^{1.5}, d^{1,75}]\sqrt{n})$	$\tilde{O}(d^{3.5}\sqrt{n})$	$O(d^3)$

Regret Guarantee

Theorem 1 & 2

There exists an algorithm such that with probability at least $1 - \delta$,

$$\text{Reg}_n \leq d^{3.5}\sqrt{n}\,\text{polylog}(n, d, 1/\delta)$$

In the Stochastic setting this can be improved to

$$\text{Reg}_n \leq Md^{2}\sqrt{n}\,\text{polylog}(n, d, 1/\delta)$$

Where $M \in [d^{-1/2}, d^{-1/4}]$, depending on the geometry of the body K

- Only Boundedness and Convexity needed

Some Intuition

Plan of attack

- Adapt the algorithm for the unconstrained setting of [Lattimore and György, 2023]
  - **Unconstrained -> Constrained:** Construct a "Bandit" extension based on the Minkowski functional inspired by [Mhammedi, 2022]
  - **Stochastic -> Adversarial:**
    - Add "Negative" bonuses, which can be seen as an increasing learning rate and improves exploration
    - Use a similar restarting technique as in [Suggala et al. 2021] and [Bubeck et al, 2017]

Convex Extension, Bandit version

Convex extension, bandit edition

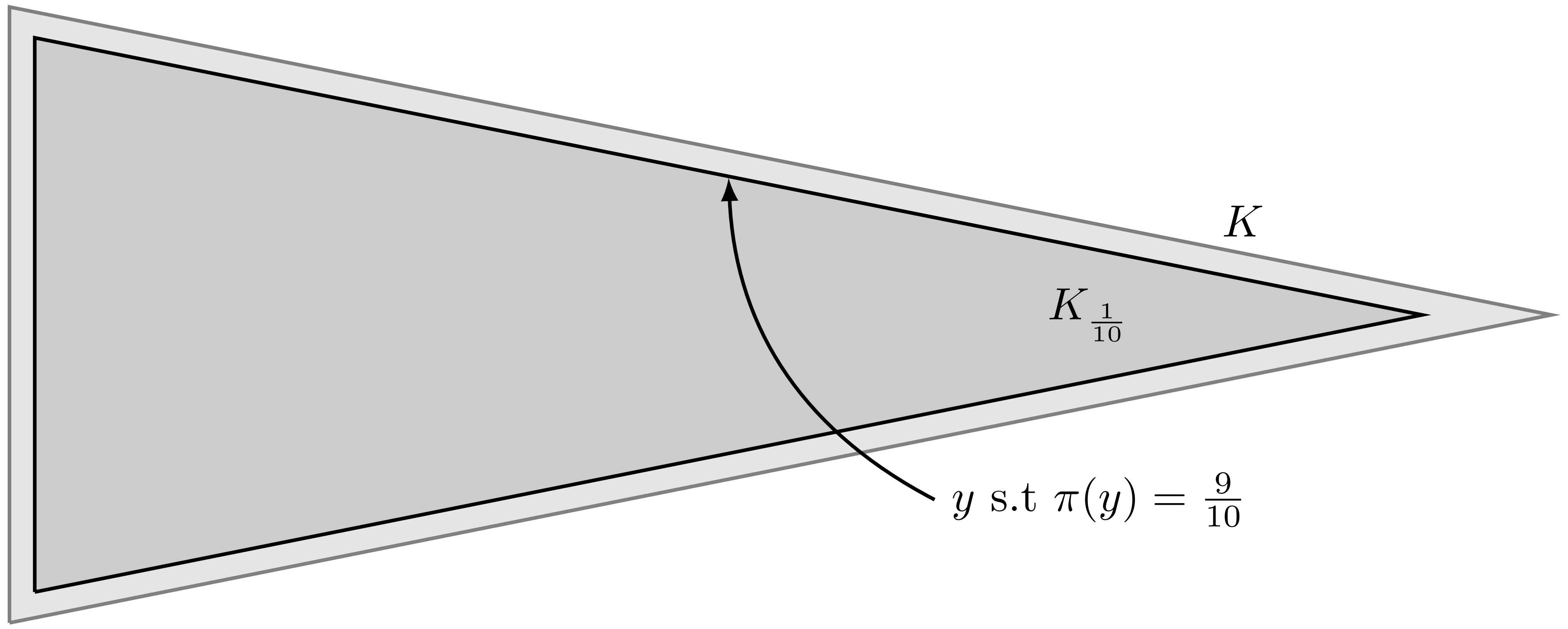
Utilise the Minkowski functional:
$\pi(x) = \inf\{t > 0 : x \in tK\}$

Shrink $K \to K_\varepsilon = \left\{ x \in \mathbb{R}^d \mid \pi_\epsilon(x) = \frac{\pi(x)}{1-\epsilon} \leq 1 \right\}$

Define the extension of ℓ from $K_\varepsilon \to \mathbb{R}^d$,

$$f(x) = \pi_+(x)\ell\left(\frac{x}{\pi_+(x)}\right) + \frac{2(\pi_+(x)-1)}{\epsilon}$$
$$= \pi_+(x)\ell\left(\frac{x}{\pi_+(x)}\right) + 2v(x)\,.$$

$\pi_+(x) = \max(1, \pi_\varepsilon(x))$



Inspired by [Lattimore 2024, Figure 3.1]

"Linearly" extend from the boundary of K_ε

Convex extension, bandit edition

Estimation

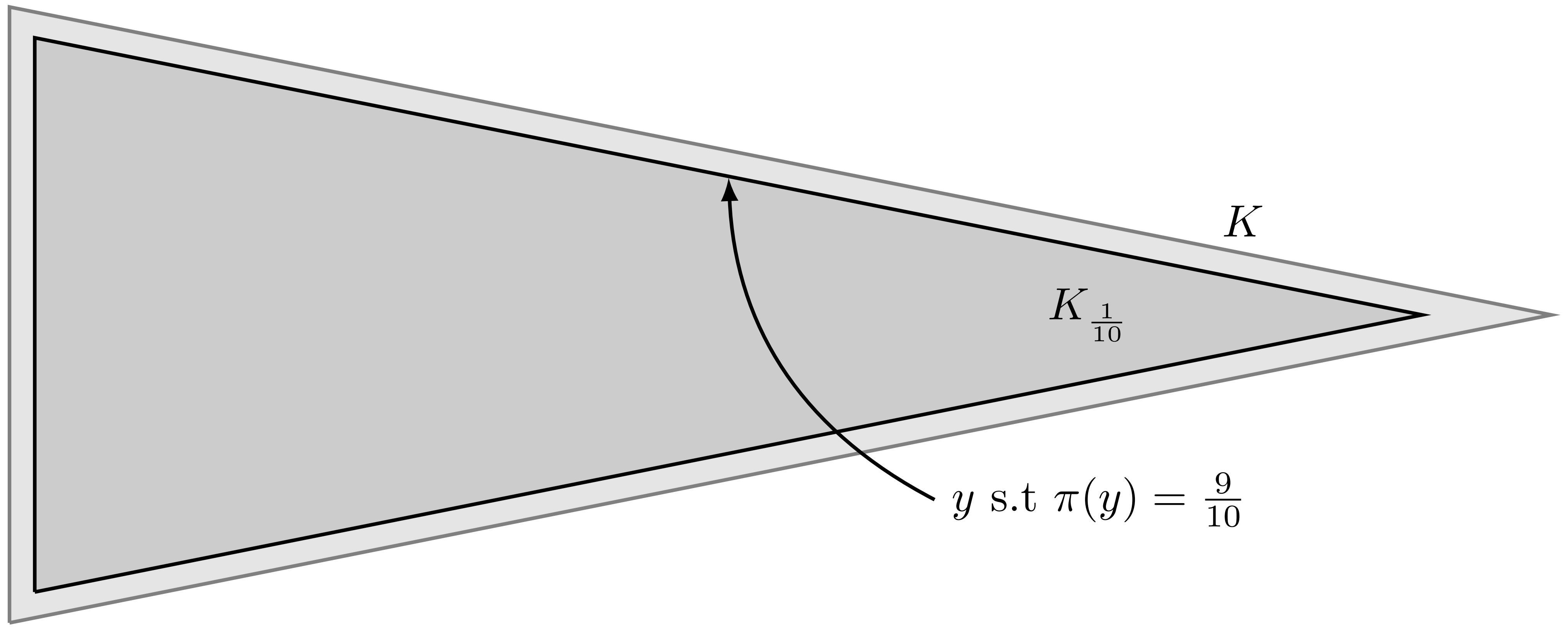
Utilise the Minkowski functional:
$\pi(x) = \inf\{t > 0 : x \in tK\}$

Shrink $K \to K_\varepsilon$ using $\pi_\epsilon(x) = \frac{\pi(x)}{1-\epsilon} \leq 1$

Set the action to $A = X/\pi_+(X)$, and estimate the loss

$$Y = \pi_+(X)[\ell(A) + \varepsilon] + 2v(X).$$

$$\pi_+(x) = \max(1, \pi_\varepsilon(x))$$



Inspired by [Lattimore 2024, Figure 3.1]

"Linearly" extend from the boundary of K_ε

Convex extension, bandit edition

Properties ℓ_t extension

$$f(x) = \pi_+(x)\ell\left(\frac{x}{\pi_+(x)}\right) + 2v(x)$$

Lemma 4

- $f(x) = \ell(x)$ for all $x \in K_\varepsilon$
- f is convex on $\mathbb{R}^d$
- $\partial_x f(x) \geq 0$ for all $x \notin K_\varepsilon$

- Need $\varepsilon = \Theta(1/\sqrt{n})$
- Leads to f being $O(\sqrt{n})$-Lipschitz
- But still $\sum_{t=1}^{n} Y_t^2 = \tilde{O}(n)$

Gaussian Optimistic Smoothing

Motivation

Full Information setting:

- Run OGD, Online Newton, etc.

To run Online Newton:

From 1 sample, Y_t, Estimate:

- $\ell_t(A_t)$
- $\nabla \ell_t(A_t)$
- $\nabla^2 \ell_t(A_t)$

Some Problems & Requirements:

- Twice Differentiable
- The obvious estimators need multiple queries
- For non-linear ℓ_t there exist no unbiased estimators

Gaussian Optimistic Smoothing

Definition

Definition

Let $f\colon \mathbb{R}^d \to \mathbb{R}$ be a bounded convex function, and $X \sim \mathcal{N}(\mu, \Sigma)$, given paramater $\lambda \in \left(0, \frac{1}{1+d}\right)$, define

$$s(x) = \mathbb{E}\left[\left(1 - \frac{1}{\lambda}\right) f(X) + \frac{1}{\lambda} f((1-\lambda)X + \lambda x)\right]$$

Set $q(x) = \langle s'(\mu), x - \mu \rangle + \frac{1}{4} \|x - \mu\|^2_{s''(\mu)}$

Gaussian Optimistic Smoothing

Estimation

Definition

Estimated versions of s and its gradients and hessians is obtained by

- $\hat{s}(z) = \left(1 + \frac{r(X,z) - 1}{\lambda}\right) Y$
- $\hat{s}'(\mu) = \frac{Yr(X,\mu)}{1-\lambda} \Sigma_t^{-1} \left(\frac{X-\mu}{1-\lambda}\right)$
- $\hat{s}'(\mu) = \frac{\lambda Y r(X,\mu)}{(1-\lambda)^2} \left(\Sigma^{-1}\left[\frac{X-\mu}{1-\lambda}\right]\left[\frac{X-\mu}{1-\lambda}\right]^\top \Sigma^{-1} - \Sigma^{-1}\right)$
- $\hat{q}(z) = \langle \hat{s}'(\mu), z - \mu \rangle + \frac{1}{4}\|z-\mu\|^2_{\hat{s}''(\mu)}$
- $r(X,z) = \frac{p\left(\frac{X-\lambda z}{1-\lambda}\right)}{(1-\lambda)^d p(X)}$, where p is the density of the $\mathcal{N}(\mu, \Sigma)$-distribution.

Gaussian Optimistic Smoothing

Estimation

Definition

Estimated versions of s and its gradients and hessians is obtained by

- $\hat{s}(z) = \left(1 + \frac{r(X,z) - 1}{\lambda}\right) Y$
- $\hat{s}'(\mu) = \frac{Yr(X,\mu)}{1-\lambda}\Sigma_t^{-1}\left(\frac{X-\mu}{1-\lambda}\right)$
- $\hat{s}'(\mu) = \frac{\lambda Y r(X,\mu)}{(1-\lambda)^2}\left(\Sigma^{-1}\left[\frac{X-\mu}{1-\lambda}\right]\left[\frac{X-\mu}{1-\lambda}\right]^\top \Sigma^{-1} - \Sigma^{-1}\right)$
- $\hat{q}(z) = \langle \hat{s}'(\mu), z - \mu\rangle + \frac{1}{4}\|z-\mu\|^2_{\hat{s}''(\mu)}$
- $r(X,z) = \frac{p\left(\frac{X-\lambda z}{1-\lambda}\right)}{(1-\lambda)^d p(X)}$, where p is the density of the $\mathcal{N}(\mu,\Sigma)$-distribution.

Only concentrates well in the focus regions

$$\left\{x \in K \mid \lambda\|x-\mu_t\|_{\Sigma_t^{-1}} \leq \frac{1}{L}\right\}$$

Summary

- Extend ℓ_t from K to $\mathbb{R}^d$
- Calculate s_t surrogate
- Estimate Quadratic Approximation

- Next step is: Run Follow-The-Regularized-Leader
  - (With 2 extra tricks in the Adversarial case)

Algorithm

For the Stochastic case

Algorithm

In words, Stochastic case

input n, η, λ, σ and $K_0 = K_\varepsilon$
for $t = 1$ **to** n
  let $\Phi_{t-1}(x) = \frac{1}{2\sigma^2}\|x\|^2 + \eta \sum_{u=1}^{t-1} \hat{q}_u(x)$
  compute $\mu_t = \arg\min_{x \in K_{t-1}} \Phi_{t-1}(x)$ and $\Sigma_t^{-1} = \Phi''_{t-1}(\mu_t)$
  sample $X_t \sim \mathcal{N}(\mu_t, \Sigma_t)$
  play $A_t = \frac{X_t}{\pi_+(X_t)}$ and observe $Y_t = \pi_+(X_t)[\ell_t(A_t) + \varepsilon_t] + 2v(X_t)$
  $K_t = K_{t-1} \cap \{x : \|x - \mu_t\|_t^2 \leq \mathrm{F}_{\max}\}$
end for

- Run FTRL on Quadratic estimation + Regularizer
- Determines μ_t and Σ_t^{-1}
- Sample action and observe loss
- Update Focus region

In the paper

- Adversarial Case:
  - Add negative terms to the objective
  - Restart condition
- How the geometry of K affects the regret result in the stochastic case
- Submodular minimisation
- Proofs

Thank you for your attention!

References

- Flaxman, A. D., Kalai, A. T., & McMahan, H. B. (2005, January). Online convex optimization in the bandit setting: gradient descent without a gradient. In Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms (pp. 385-394).
- Abernethy, J., Hazan, E., & Rakhlin, A. (2008, December). Competing in the dark: An efficient algorithm for bandit linear optimization. In Conference on Learning Theory
- Hazan, E., & Levy, K. (2014). Bandit convex optimization: Towards tight bounds. Advances in Neural Information Processing Systems, 27.
- Lattimore, T., & György, A. (2023, July). A second-order method for stochastic bandit convex optimisation. In *The Thirty Sixth Annual Conference on Learning Theory* (pp. 2067-2094). PMLR.
- Suggala, A. S., Ravikumar, P., & Netrapalli, P. (2021, July). Efficient bandit convex optimization: Beyond linear losses. In *Conference on Learning Theory* (pp. 4008-4067). PMLR.Proofs
- Mhammedi, Z. (2022, June). Efficient projection-free online convex optimization with membership oracle. In *Conference on Learning Theory* (pp. 5314-5390). PMLR.
- Bubeck, S., Lee, Y. T., & Eldan, R. (2017, June). Kernel-based methods for bandit convex optimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing* (pp. 72-85)
- Lattimore, T., & Gyorgy, A. (2021, July). Improved regret for zeroth-order stochastic convex bandits. In Conference on Learning Theory (pp. 2938-2964). PMLR.
- Lattimore, T. (2020). Improved regret for zeroth-order adversarial bandit convex optimisation. Mathematical Statistics and Learning, 2(3), 311-334.
- Suggala, A., Sun, Y. J., Netrapalli, P., & Hazan, E. (2024). Second Order Methods for Bandit Optimization and Control. In Conference on Learning Theory. (pp. x-x). PMLR.

Algorithm

For the Adversarial case

Algorithm

In words, Full version

```
input n, η, λ, γ, σ and K_0 = K_ε
for t = 1 to n
    let Φ_{t-1}(x) = ...
```

input n, η, λ, γ, σ and $K_0 = K_\varepsilon$

for $t = 1$ **to** n

   let $\Phi_{t-1}(x) = \frac{1}{2}\|x\|^2 + \sum_{u=1}^{t-1} \flat_u(x) + \eta \sum_{u=1}^{t-1} \hat{q}_u(x)$

   compute $\mu_t = \arg\min_{x \in K_{t-1}} \Phi_{t-1}(x)$ and $\Sigma_t^{-1} = \Phi''_{t-1}(\mu_t)$

   sample $X_t \sim \mathcal{N}(\mu_t, \Sigma_t)$

   play $A_t = \frac{X_t}{\pi_+(X_t)}$ and observe $Y_t = \pi_+(X_t)[\ell_t(A_t) + \varepsilon_t] + 2v(X_t)$

   $K_t = K_{t-1} \cap \{x : \|x - \mu_t\|_t^2 \leq \mathrm{F}_{\max}\}$

   compute $z_t = \arg\min_{z \in \mathbb{R}^d} \sum_{s=1}^{t-1} \mathbf{1}(\flat_s \neq \mathbf{0}) \|z - \mu_s\|_s^2$

$$\flat_t(x) = \begin{cases} 0 & \text{if } \sum_{s=1}^{t-1} \mathbf{1}(\flat_s \neq \mathbf{0}) \|z_t - \mu_s\|_s^2 \geq \frac{\mathrm{F}_{\max}}{16} \\ -\gamma \|x - \mu_t\|_t^2 & \text{if } \|\cdot\|_t^2 \not\preceq \sum_{s=1}^{t-1} \mathbf{1}(\flat_s \neq \mathbf{0}) \|\cdot\|_s^2 \\ -\gamma \|x - \mu_t\|_t^2 & \text{if } \|\mu_t - z_t\|_t^2 \geq \frac{\mathrm{F}_{\max}}{8} \\ 0 & \text{otherwise}. \end{cases}$$

   if $\max_{y \in K_t} \eta \sum_{u=1}^{t} (\hat{s}_u(\mu_u) - \hat{s}_u(y)) \leq -\frac{\gamma \mathrm{F}_{\max}}{32}$

      then restart algorithm

   end if

end for

- Run FTRL on Quadratic estimation + Bonus + Regularizer
- Determines μ_t and Σ_t^{-1}
- Sample action and observe loss
- Update Focus region
- Add bonus in Adversarial setting
- Check if the optimum is not moving away

Geometry of the Constraint Set

Stochastic Case Only

The mean width of the polar body K° of the constraint set K

$$M(K^\circ) = \int_{\mathbb{S}(1)} \pi(x) \mathrm{d}\rho(x)$$

The paramater of interest is $M = \max(d^{-1/2}, M(K^\circ))$

Dimension dependence is controlled by Md^2

(a) Without any assumption on K you can take $M = d^{-1/2}$, but the algorithm may be computationally inefficient.

(b) Given access to sampling and membership oracles for K you can take $M = d^{-1/4}$ and the algorithm is efficient.

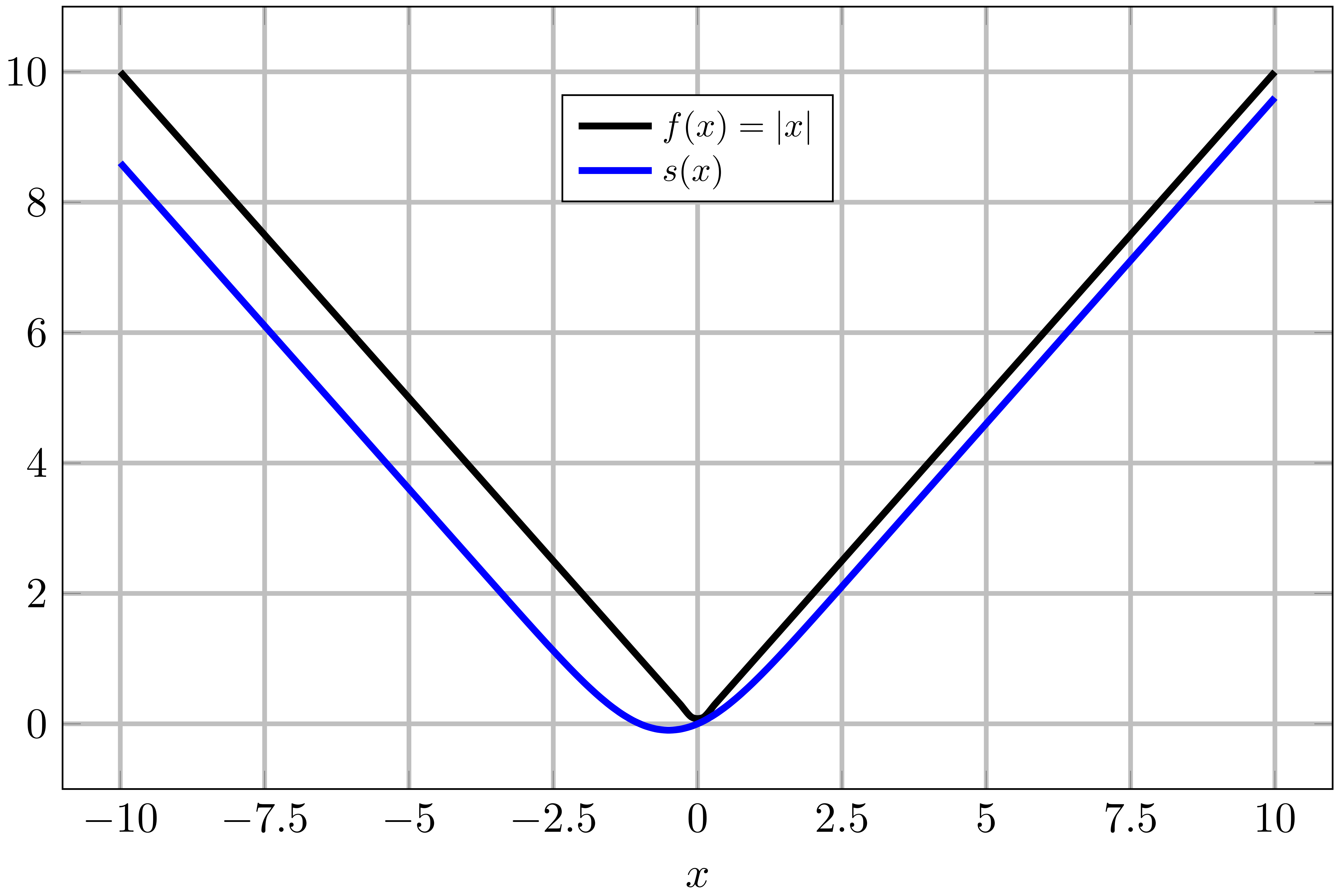
(c) Given access to sampling and membership oracles for a symmetric K you can take $M = d^{-1/2}$ and the algorithm is efficient.

Restart condition

Informally:

- If the optimum is leaving the Focus region -> Restart triggered
- If a Restart is triggered -> Negative regret -> Restart is Safe

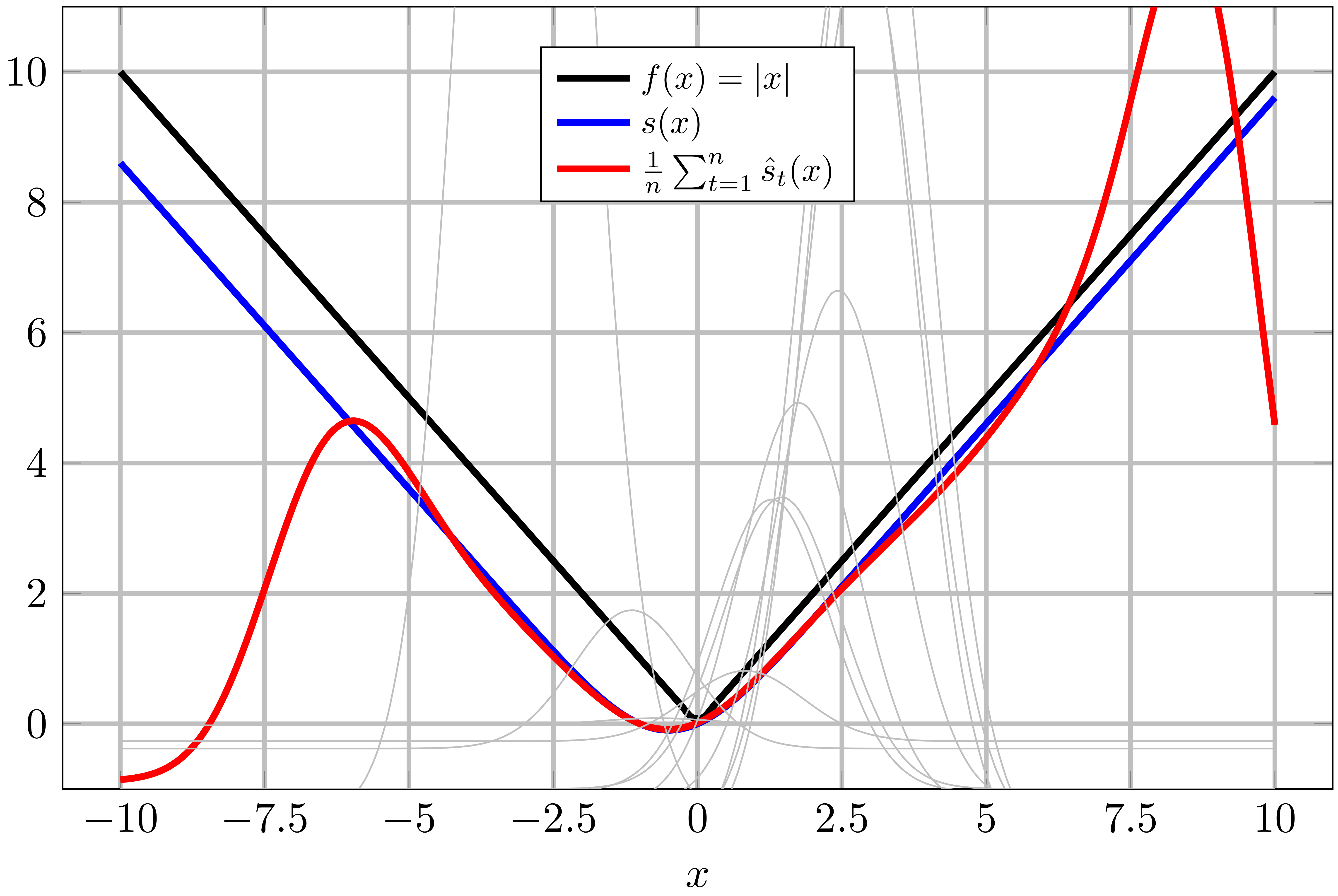
Example



- $\mu = \frac{1}{2}$, $\sigma^2 = 1$
- $\lambda = \frac{1}{2}$

- Approximation only good around μ
- Always Optimistic

Example



- $n = 10^4$
- $\mu = \frac{1}{2}$, $\sigma^2 = 1$
- $\lambda = \frac{1}{2}$

- Approximation only good around μ
- Estimation really only valid around μ
- Non-Convex!
- Not Optimistic everywhere