ZEROTH-ORDER (STOCHASTIC) OPTIMISATION

Tor Lattimore

Google DeepMind, London

CLASSICAL PROBLEM

How to (efficiently) find (an approximation of)

$$x_\star = \arg\min_{x \in \mathcal{K}} f(x)$$

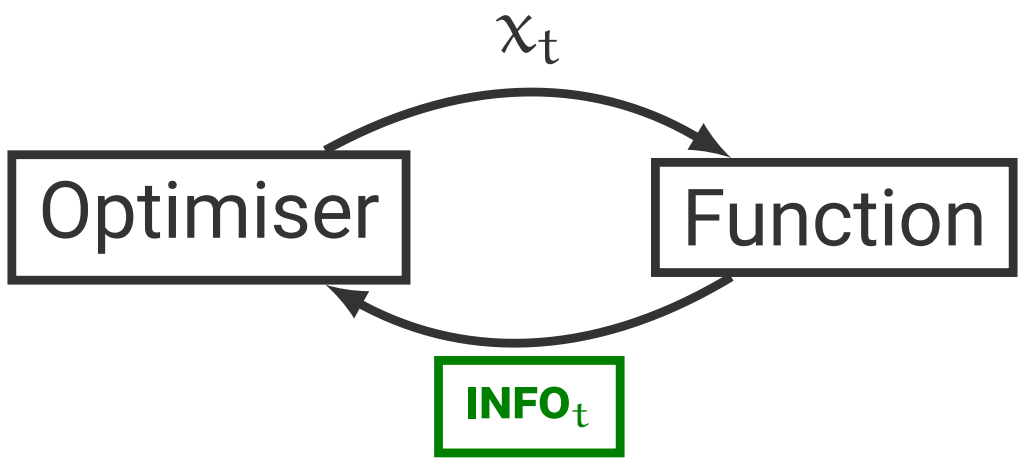
THREE DIMENSIONS OF COMPLEXITY

1. Assumptions made on f
2. Assumptions made on $\mathcal{K}\,(\subset \mathbb{R}^d)$
3. What is the interaction protocol

INTERACTION PROTOCOLS

Optimiser sequentially chooses points x_t in $\mathcal{K}$

Observes information about f at the chosen point



$$\text{INFO}_t \subset \{f(x_t), \nabla f(x_t), \nabla^2 f(x_t), \ldots, \widehat{f}(x_t), \widehat{\nabla f}(x_t)\}$$

$$\widehat{f}(x_t) = f(x_t) + \varepsilon_t \qquad \widehat{\nabla f}(x_t) = \nabla f(x_t) + \boldsymbol{\varepsilon_t}$$

NOISE TERMS $\mathbb{E}[\varepsilon_t] = 0$ and $\mathbb{E}[\boldsymbol{\varepsilon_t}] = \mathbf{0}$

INTERACTION PROTOCOLS

INFO	NAME
$f(x_t)$	zeroth order
$\nabla f(x_t)$	first order
$\nabla f(x_t)$ and $\nabla^2 f(x_t)$	second order
$\widehat{f}(x_t)$	stochastic zeroth order
$\widehat{\nabla f}(x_t)$	stochastic first order

OUR FOCUS TODAY

- (Stochastic) zeroth order
- f is convex
- f is Lipschitz

$$f(x) - f(y) \leqslant \|x - y\| \quad \forall x, y$$

- Unconstrained

$\mathcal{K} = \mathbb{R}^d$ and $x_\star \in \{x : \|x\| \leqslant 1\}$

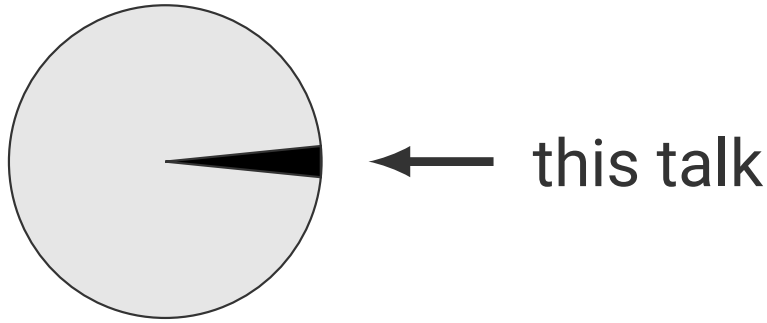
NOT IN FOCUS

- Non-geometric problems
- Bayesian optimisation

Frazier 2018

- Non-convex problems

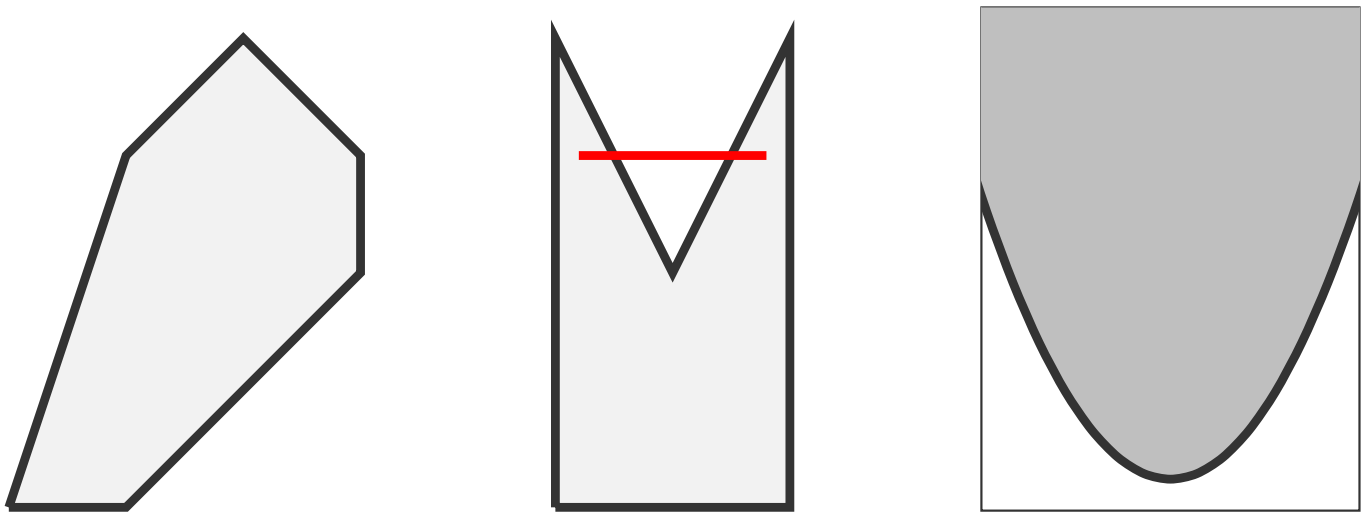
Roy et al. 2022



CONVEXITY

A set $\mathcal{K} \subset \mathbb{R}^d$ is convex if for all $x, y \in \mathcal{K}$ and $\lambda \in [0, 1]$,

$$(1 - \lambda)x + \lambda y \in \mathcal{K}$$



$f : \mathcal{K} \to \mathbb{R}$ is convex if $\mathcal{K}$ is convex and any of the following hold:

- $\nabla^2 f(x)$ is positive semi-definite for all $x \in \mathcal{K}$
- $(1 - \lambda)f(x) + \lambda f(y) \geqslant f((1 - \lambda)x + \lambda y)$
- $\mathbb{E}[f(X)] \geqslant f(\mathbb{E}[X])$ for all random $X \in \mathcal{K}$ [Jensen's inequality]

WHY STUDY ZEROTH-ORDER OPTIMISATION?

- Fundamental problem
- Applications
  - Reinforcement learning
  - Operations research
  - Simulation optimisation
  - Adversarial attacks
- Simple algorithms with super low complexity [Malladi et al., 2023]

SIMULATION OPTIMISATION

Buy simulator for water treatment plant

Non-differentiable black box

Use zeroth-order optimisation to optimise control policy

RL AND BANDIT PROBLEMS FOR OPERATIONS RESEARCH

- Parameterised set of policies $\{\pi_\theta : \theta \in \mathcal{K}\}$
- Want to find (an approximation of)

$$\arg\max_{\theta \in \mathcal{K}} V^{\pi_\theta} \qquad V^{\pi_\theta} \text{ is the value of policy } \pi_\theta$$

- **EXAMPLE** In dynamic pricing the learner chooses a price and the expected profit is an unknown concave function of the price
- Learner sequentially chooses prices, observes profit and learns the optimal price

ADVERSARIAL ATTACKS

- Black-box access to a neural network $N : \mathbb{R}^m \to \mathbb{R}^k$
- Inputs are (say) images and outputs are class logits
- **GOAL** Given input image x with one-hot class label c, find y close to x so that

$$\mathrm{KL}(c, \mathrm{softmax}(N(y))) \text{ is large}$$

- Written as zeroth-order optimisation problem by

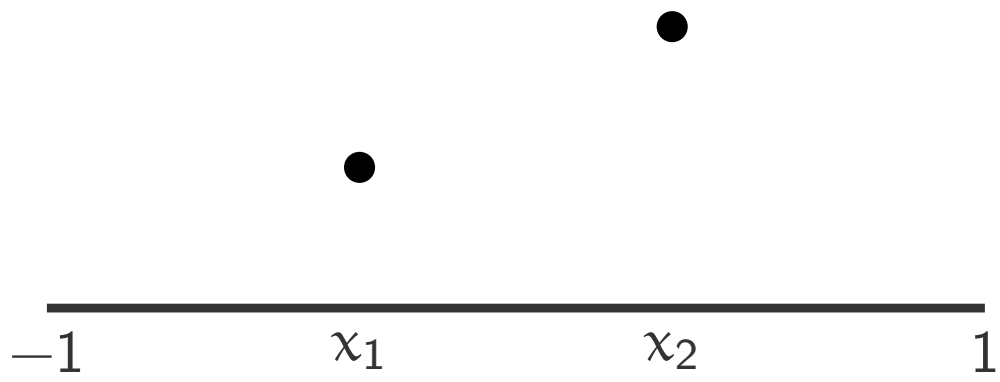
$$f(y) = -\mathrm{KL}(c, \mathrm{softmax}(N(y)))$$
$$\mathcal{K} = \{y : \|x - y\| \leqslant \varepsilon\}$$

PLAN FOR THE REMAINDER

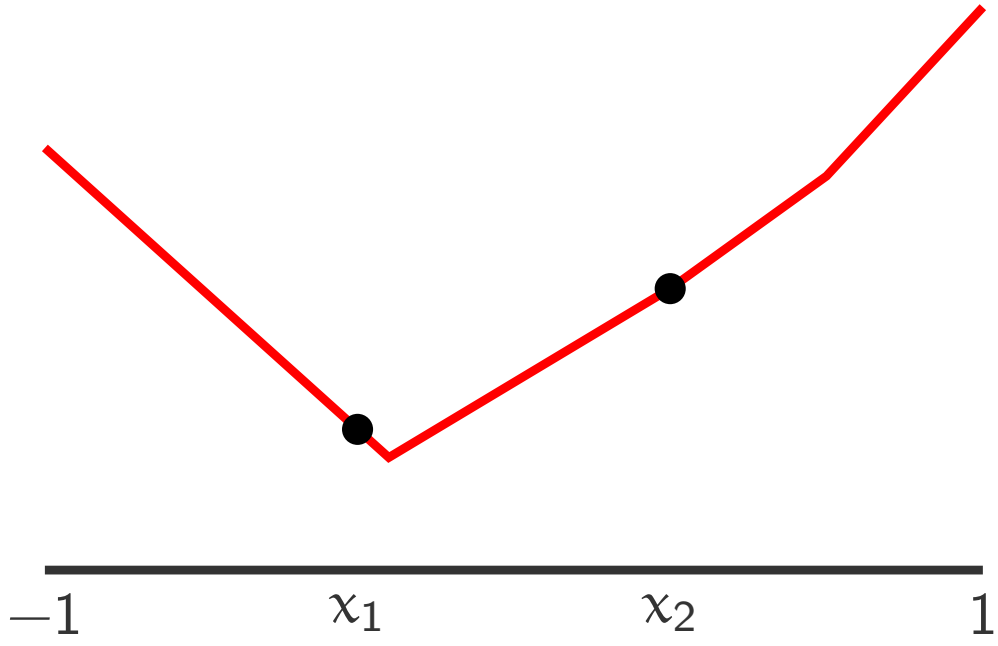
- A simple algorithm for 1-dimensional, noise free problems
- Evaluation of algorithms
- Gradient descent without gradients in stochastic problems

A SIMPLE ALGORITHM

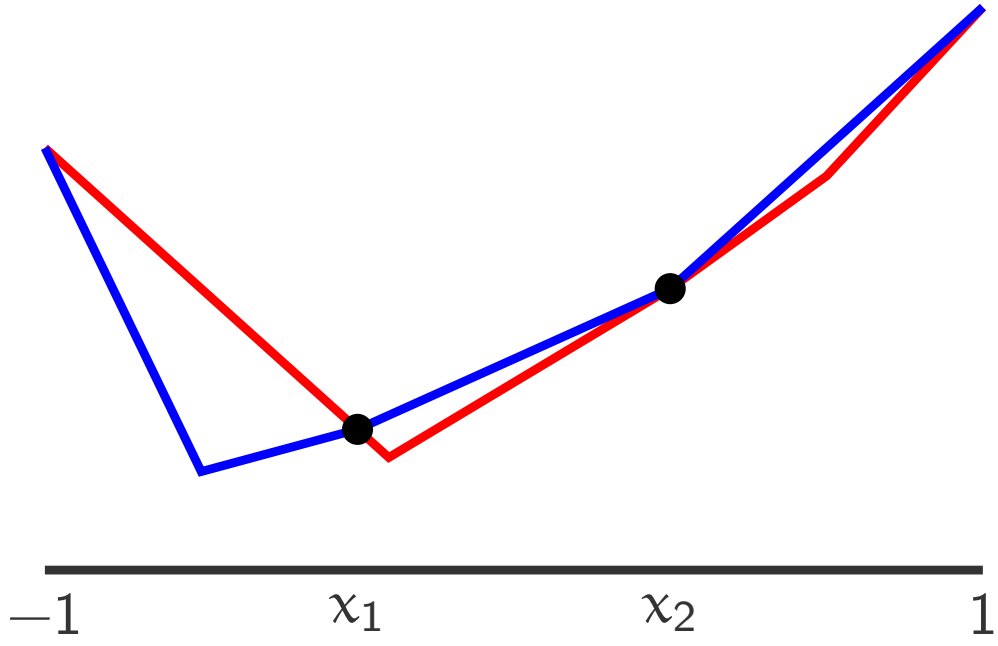
A SIMPLE ALGORITHM



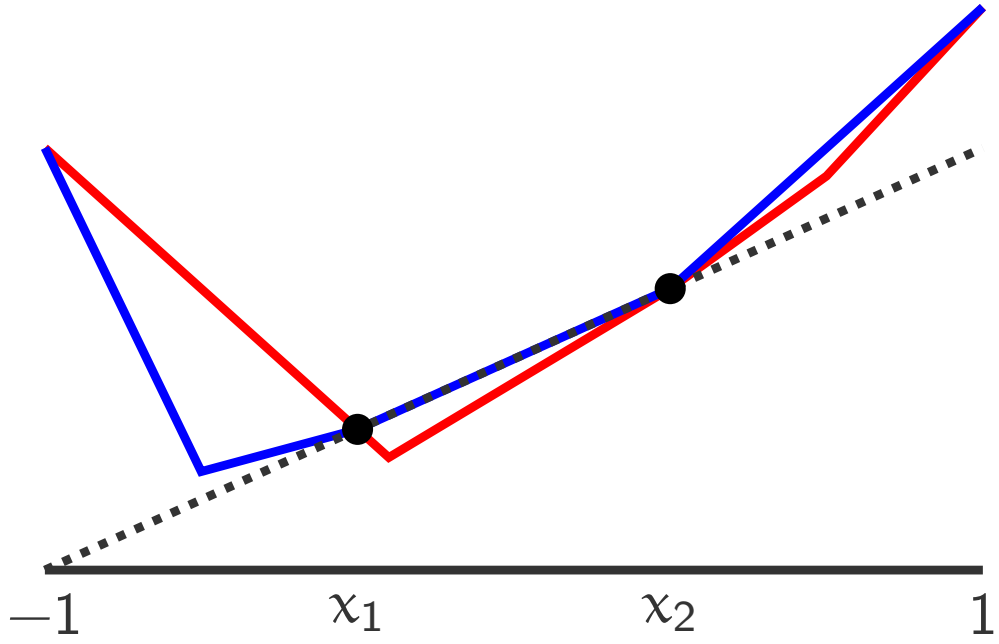
A SIMPLE ALGORITHM



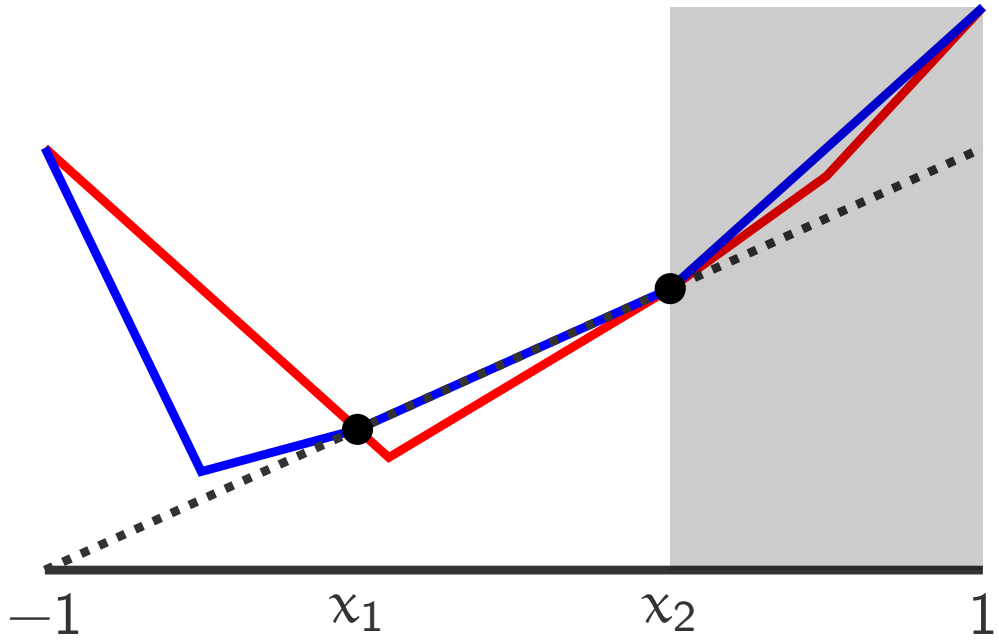
A SIMPLE ALGORITHM



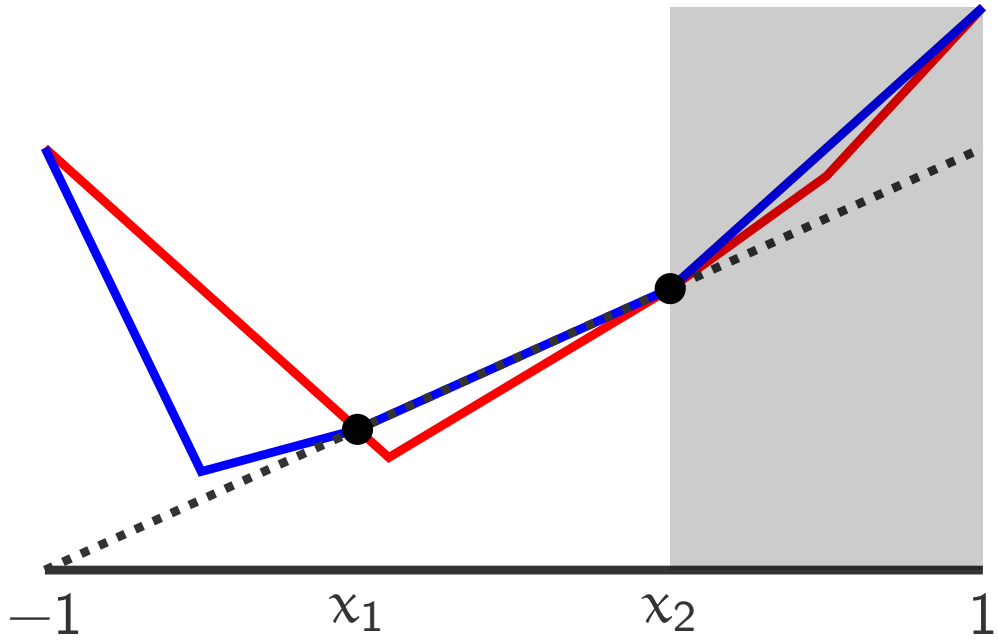
A SIMPLE ALGORITHM

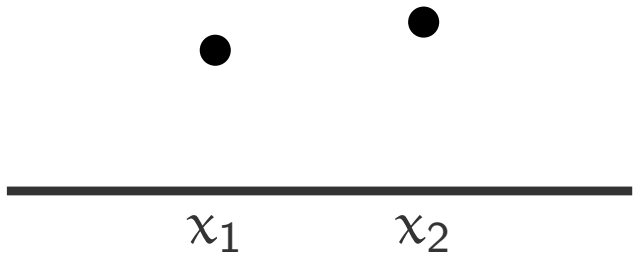


A SIMPLE ALGORITHM

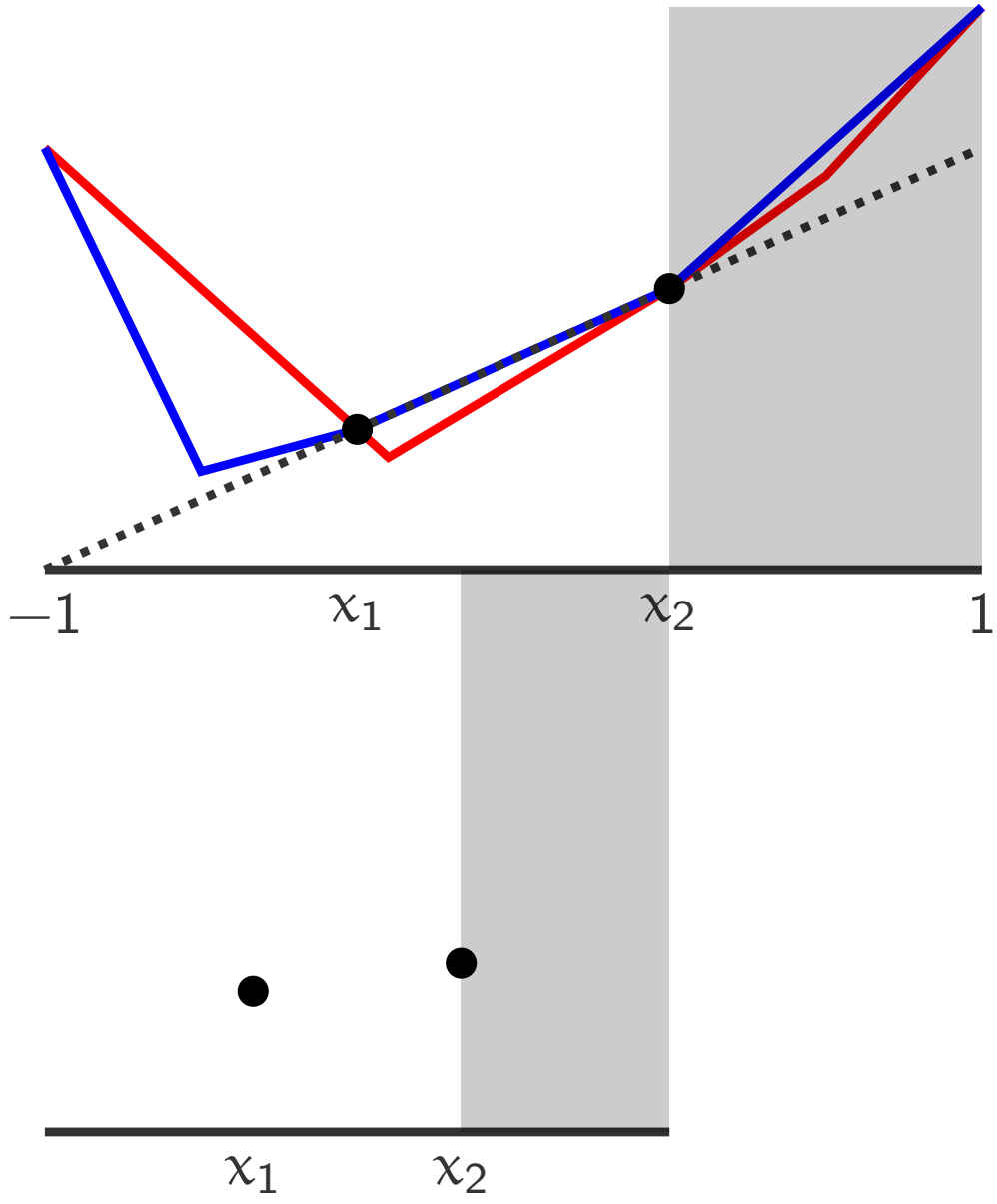


A SIMPLE ALGORITHM





A SIMPLE ALGORITHM



A SIMPLE ALGORITHM

```
left = -1, right = 1
for k = 0,...
   width = right - left
   middle1 = left + width / 3
   middle2 = left + 2 * width / 3
   if f(middle1) > f(middle2)
      left = middle1
   else
      right = middle2
```

ANALYSIS – WHAT'S THE GOAL?

Find a near minimiser, fast!

ANALYSIS – WHAT'S THE GOAL?

- **SAMPLE COMPLEXITY** Number of interactions n with oracle to find x such that

$$f(x) \leqslant f(x_\star) + \varepsilon$$

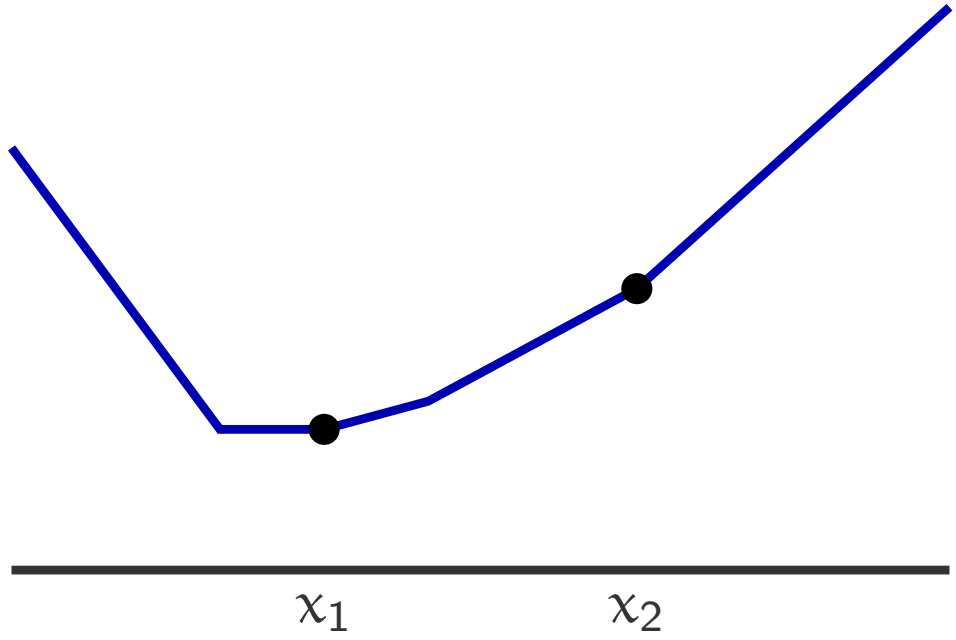
- **REGRET** The regret over n interactions is

$$\sum_{t=1}^{n} (f(x_t) - f(x_\star))$$

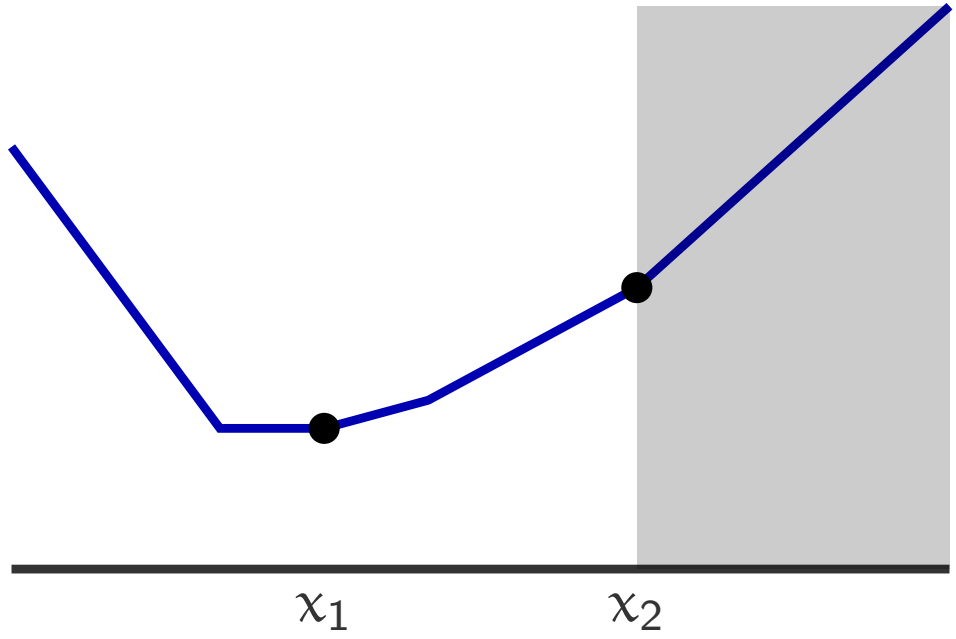
- **REGRET $\Rightarrow$ SAMPLE-COMPLEXITY** By Jensen's

$$f\left(\frac{1}{n}\sum_{t=1}^{n} x_t\right) - f(x_\star) \leqslant \frac{1}{n}\left[\sum_{t=1}^{n} (f(x_t) - f(x_\star))\right]$$

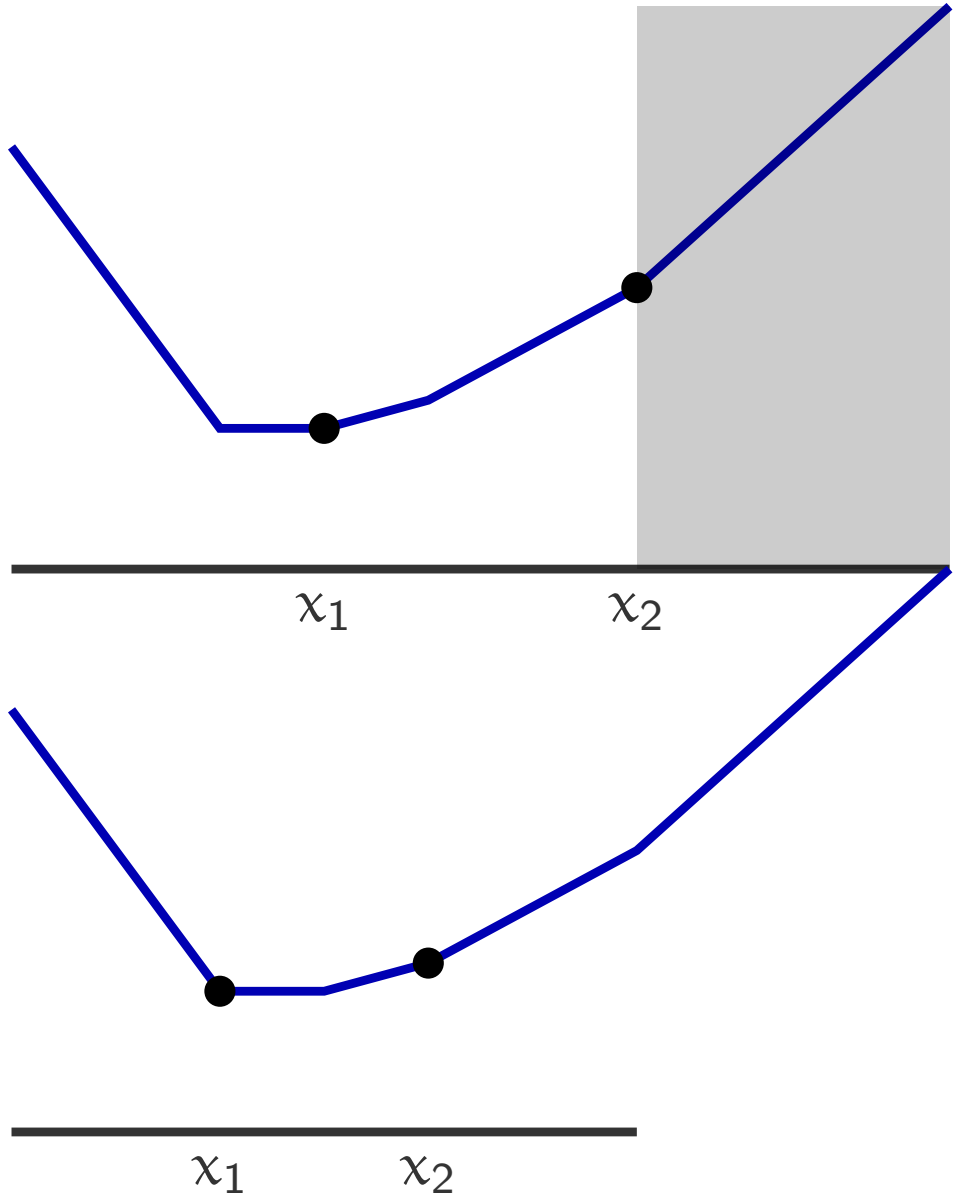
A SIMPLE ALGORITHM



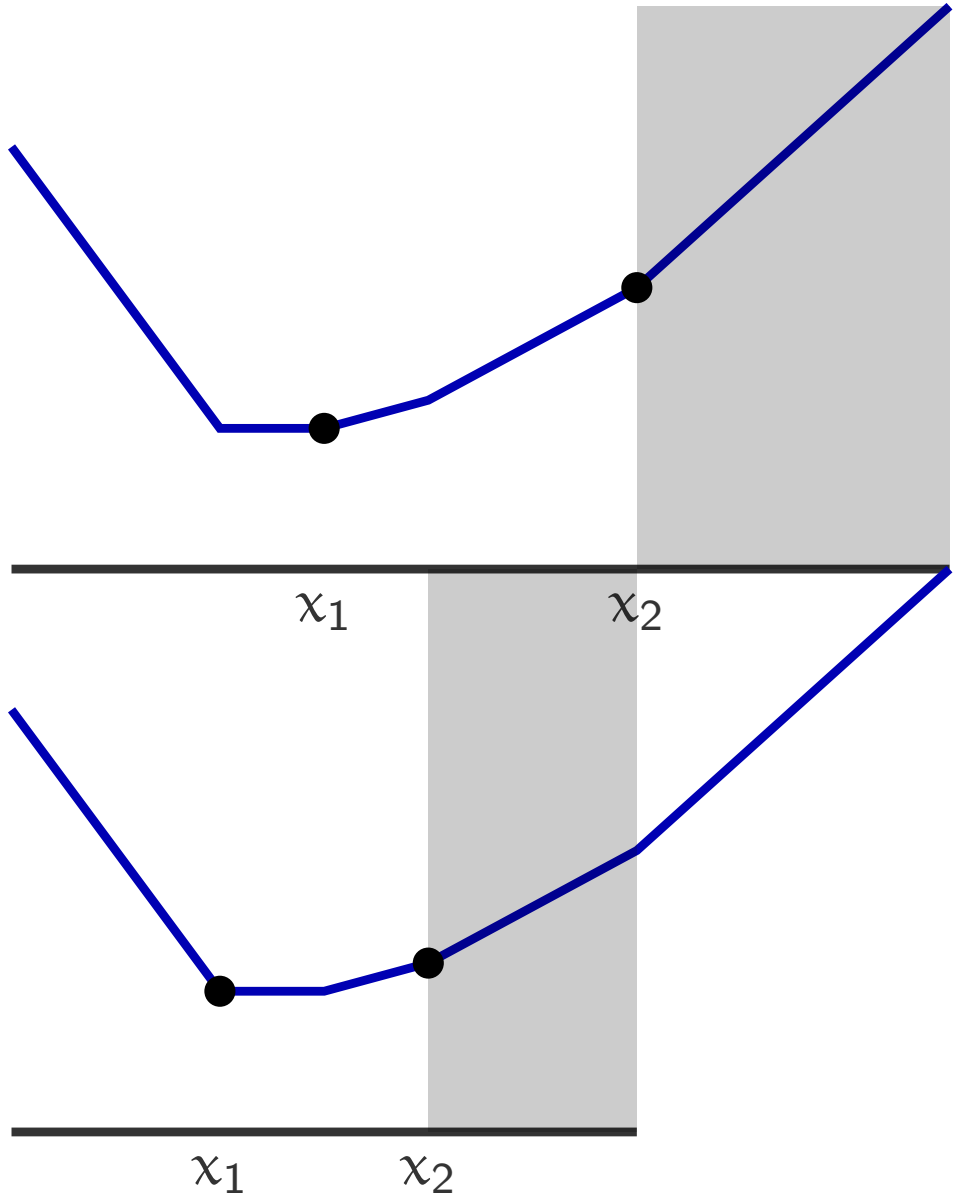
A SIMPLE ALGORITHM



A SIMPLE ALGORITHM



A SIMPLE ALGORITHM



ANALYSIS

Let $\texttt{width}_k, \texttt{left}_k, \texttt{right}_k$ denote corresponding variables in iteration k

BOUND ON WIDTH $\texttt{width}_k \leqslant 2 \cdot \left(\frac{2}{3}\right)^k$

CORRECTNESS $x_\star \in [\texttt{left}_k, \texttt{right}_k]$ for all k

BOUND ON LOSS $f(\texttt{left}_k) \leqslant f(x_\star) + \texttt{width}_k$

After $2n$ queries the algorithm has identified a point with loss at most $2 \cdot \left(\frac{2}{3}\right)^n \leqslant 2\exp(-0.405n)$

Sample complexity is $O(\log(1/\varepsilon))$

$$f(x) - f(y) \leqslant |x - y|$$

COMPARISON TO NEWTON'S METHOD

- Newton's method computes

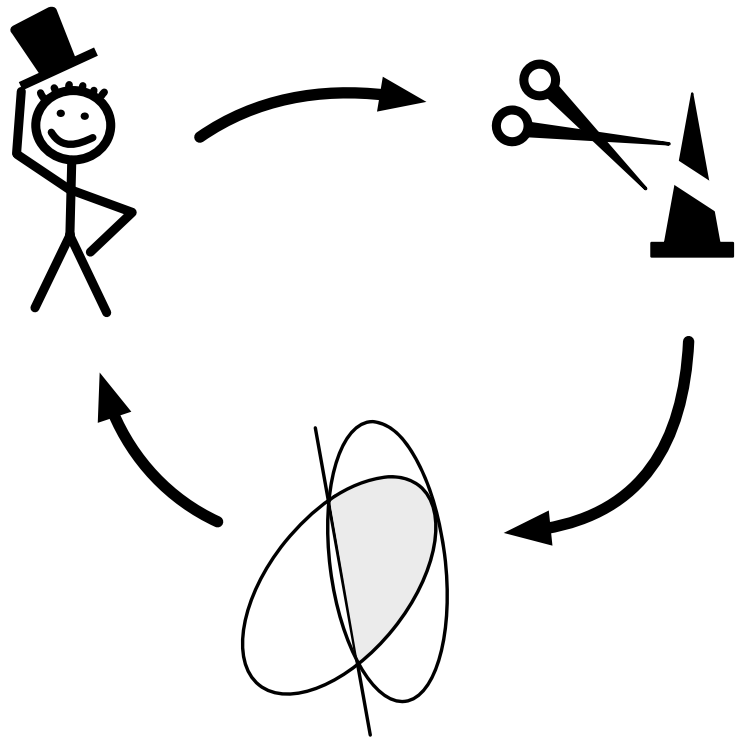
$$x_{t+1} = x_t - \nabla^2 f(x_t)^{-1} \nabla f(x_t)$$

- For **nice enough** functions converges quadratically

$$f(x_t) - f(x_\star) = O\left(\left(\frac{1}{2}\right)^{2^k}\right)$$

- Sample complexity is $\log\log(1/\varepsilon)$ [that's insane...]
- Not as robust. Hard to use in zeroth-order setting

GENERALISATION TO HIGHER DIMENSIONS AND STOCHASTIC SETTING



Agarwal et al. 2013 construct a complicated algorithm with sample complexity $\tilde{O}\left(\frac{d^{32}}{\varepsilon^2}\right)$

(STOCHASTIC) GRADIENT DESCENT

$x_1 = 0$

for $t = 1$ to n

Play x_t and observe $\widehat{\nabla} f(x_t)$

Update $x_{t+1} = x_t - \eta \widehat{\nabla} f(x_t)$

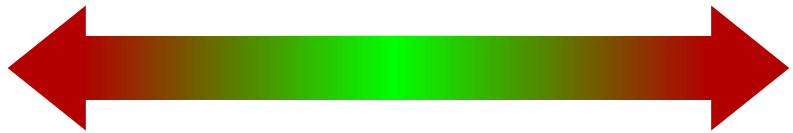
learning rate

THEORY

$$\mathbb{E}\left[\sum_{t=1}^{n}\left(f(x_t)-f(x_\star)\right)\right] \leqslant \frac{\|x_\star\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{n}\mathbb{E}\left[\|\widehat{\nabla} f(x_t)\|^2\right]$$

Does it make sense?

Algorithm too slow



Algorithm unstable

Learning rate, η

Excellent references by Bubeck [2015], Hazan [2016]

THEORY

$$\|x_\star\| \leqslant 1$$
$$\eta = \frac{1}{G\sqrt{n}}$$

$$\mathbb{E}\left[\sum_{t=1}^{n}\left(f(x_t) - f(x_\star)\right)\right] \leqslant \frac{\|x_\star\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{n}\mathbb{E}\left[\|\widehat{\nabla} f(x_t)\|^2\right]$$

$$\leqslant G\sqrt{n} \quad \text{[if } \mathbb{E}[\|\widehat{\nabla} f(x_t)\|^2] \leqslant G^2\text{]}$$

Does it make sense?

Algorithm too slow

Algorithm unstable

Learning rate, η

Excellent references by Bubeck [2015], Hazan [2016]

THEORY

$$\|x_\star\| \leqslant 1$$
$$\eta = \frac{1}{G\sqrt{n}}$$

$$\mathbb{E}\left[\sum_{t=1}^{n}(f(x_t) - f(x_\star))\right] \leqslant \frac{\|x_\star\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{n}\mathbb{E}\left[\|\widehat{\nabla} f(x_t)\|^2\right]$$
$$\leqslant G\sqrt{n} \quad \text{[if } \mathbb{E}[\|\widehat{\nabla} f(x_t)\|^2] \leqslant G^2\text{]}$$

$$\mathbb{E}\left[f\left(\frac{1}{n}\sum_{t=1}^{n} x_t\right)\right] - f(x_\star) \leqslant G\sqrt{\frac{1}{n}} \leqslant \varepsilon$$

when $n \geqslant \frac{G^2}{\varepsilon^2}$ = **SAMPLE COMPLEXITY**

Excellent references by Bubeck [2015], Hazan [2016]

GRADIENTS WITHOUT GRADIENTS

How can you estimate gradients using a noisy zeroth-order oracle?

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x-h)}{2h}$$

GRADIENTS WITHOUT GRADIENTS

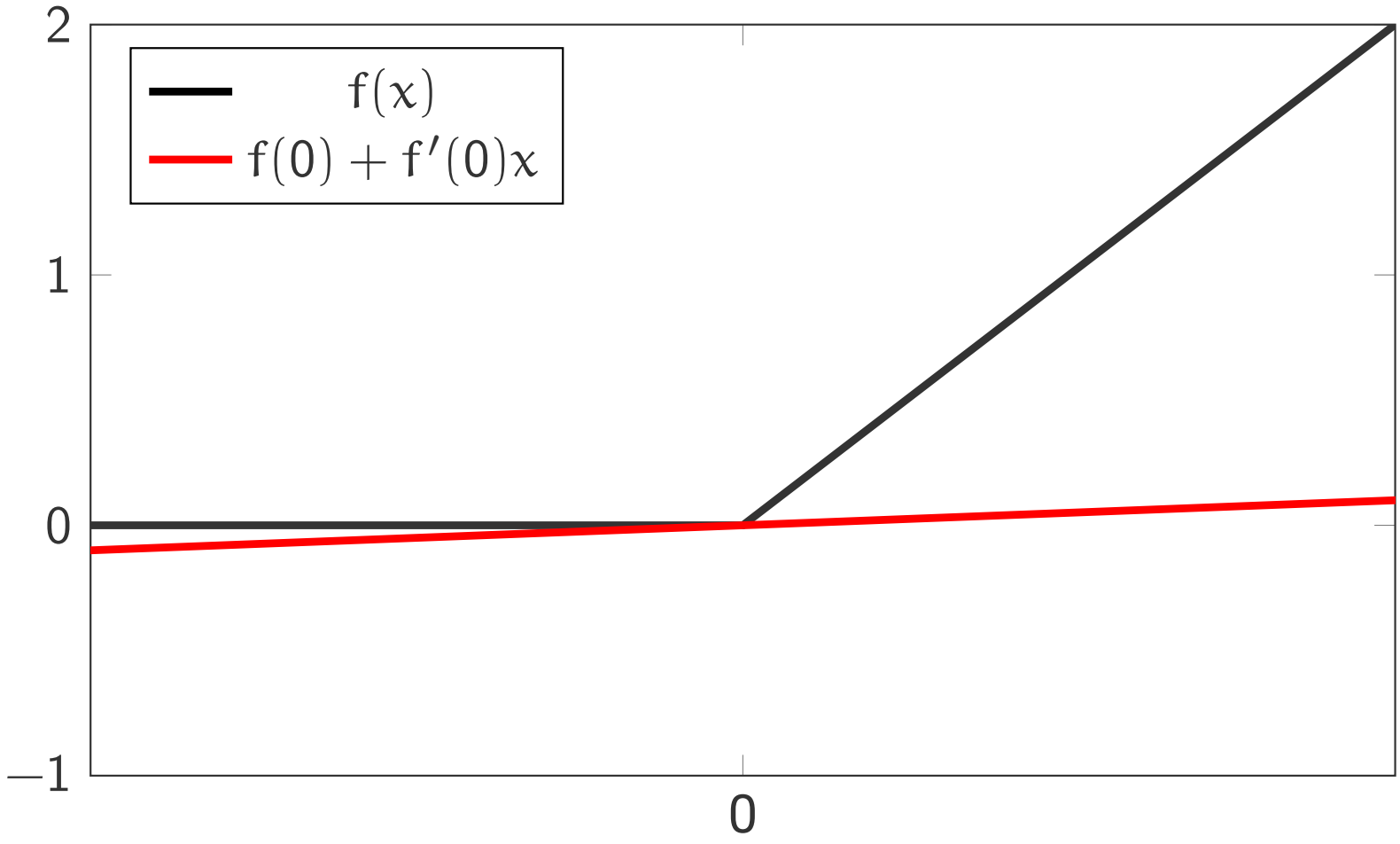
How can you estimate gradients using a noisy zeroth-order oracle?

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x-h)}{2h}$$

OBVIOUS IDEA choose h small and use

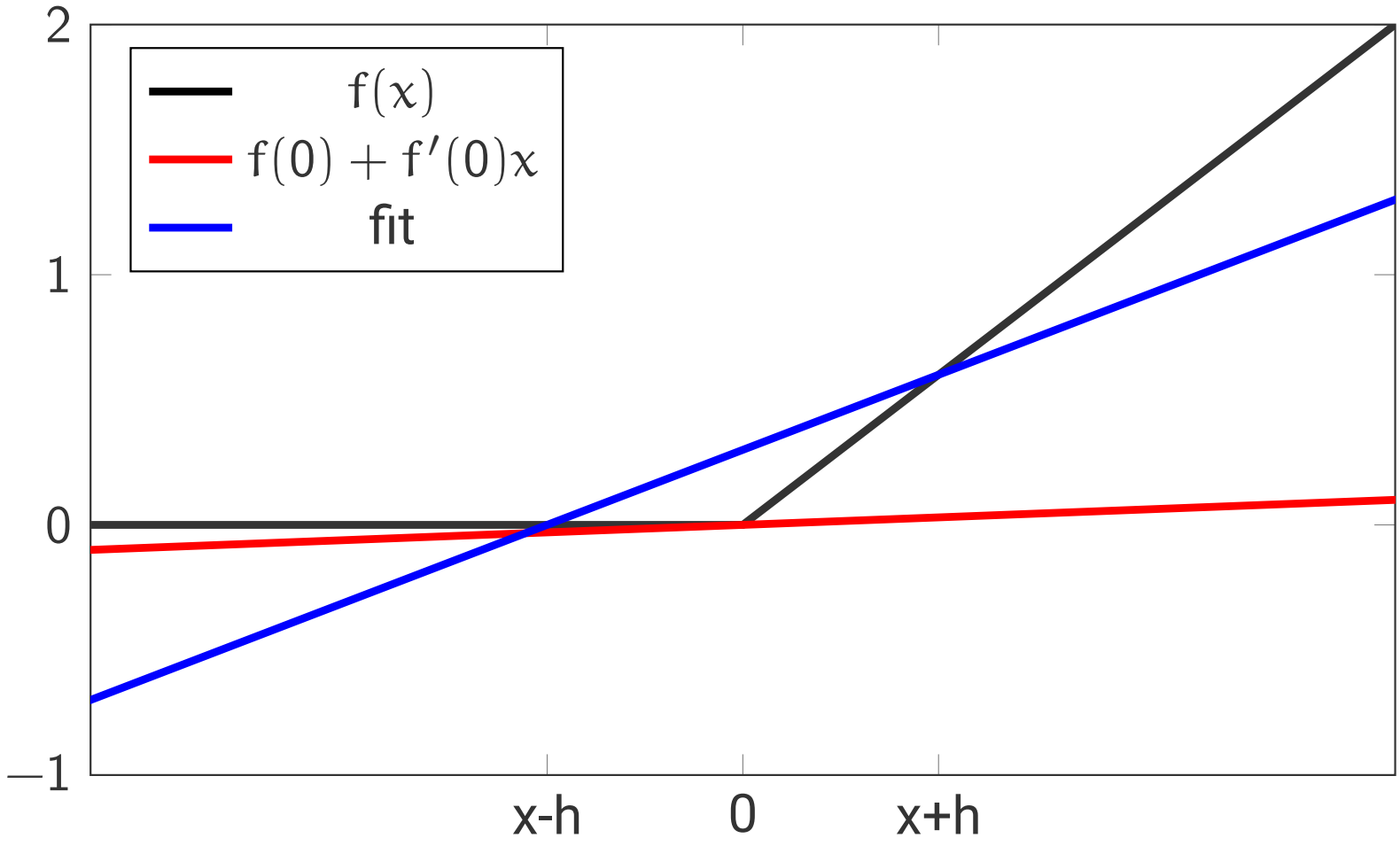
$$\widehat{f}'(x) = \frac{\widehat{f}(x+h) - \widehat{f}(x-h)}{2h}$$

ESTIMATING THE GRADIENT



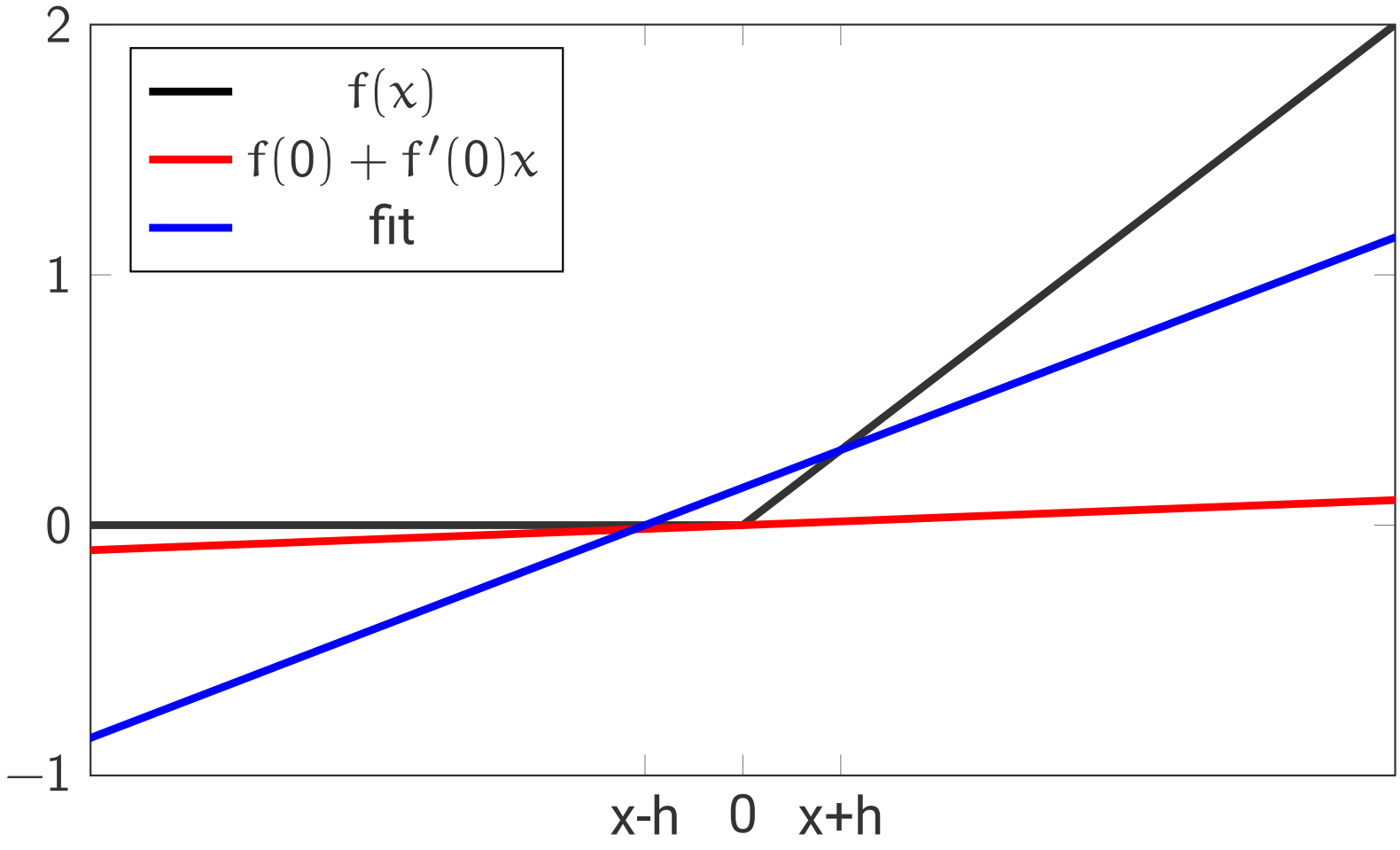
$$f'(x) \overset{?}{\approx} \frac{f(x+h) - f(x-h)}{2h}$$

ESTIMATING THE GRADIENT



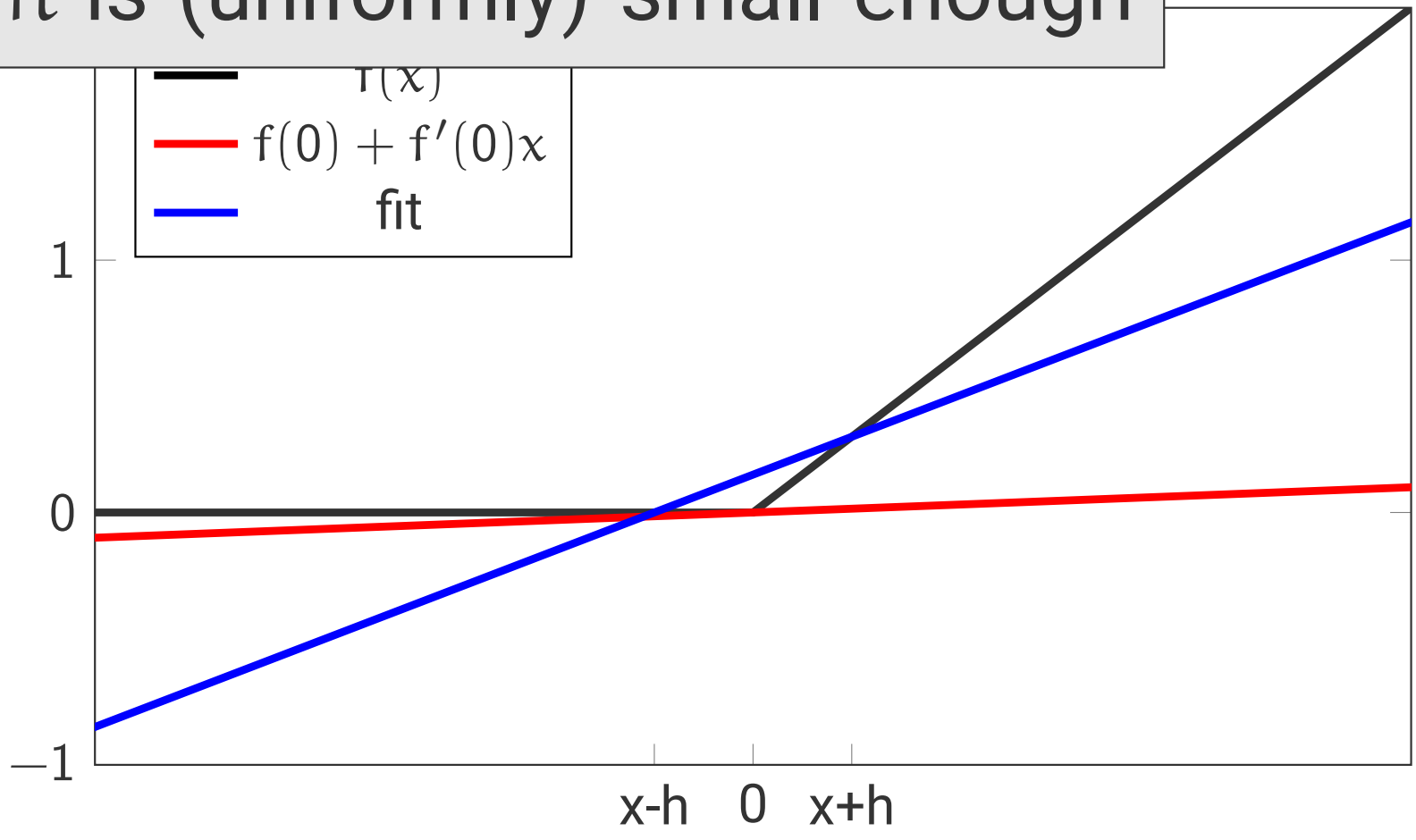
$$f'(x) \stackrel{?}{\approx} \frac{f(x+h) - f(x-h)}{2h}$$

ESTIMATING THE GRADIENT



$$f'(x) \overset{?}{\approx} \frac{f(x+h) - f(x-h)}{2h}$$

Unless gradients are Lipschitz, no h is (uniformly) small enough



$$f'(x) \overset{?}{\approx} \frac{f(x+h) - f(x-h)}{2h}$$

SMOOTHING TO THE RESCUE

$$\frac{d}{dx}\underbrace{\frac{1}{2h}\int_{-h}^{h} f(x+u)\,du}_{f_h(x)} = \underbrace{\frac{f(x+h)-f(x-h)}{2h}}_{f_h'(x)}$$

Estimate the gradient of the smoothed function

$$f_h(x) = \frac{1}{2h}\int_{-h}^{h} f(x+u)\,du \qquad f_h \to f \text{ as } h \to 0$$

ESTIMATION, BIAS AND VARIANCE

$$f_h(x) = \frac{1}{2h}\int_{-h}^{h} f(x+u)\,du$$

$$f_h'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

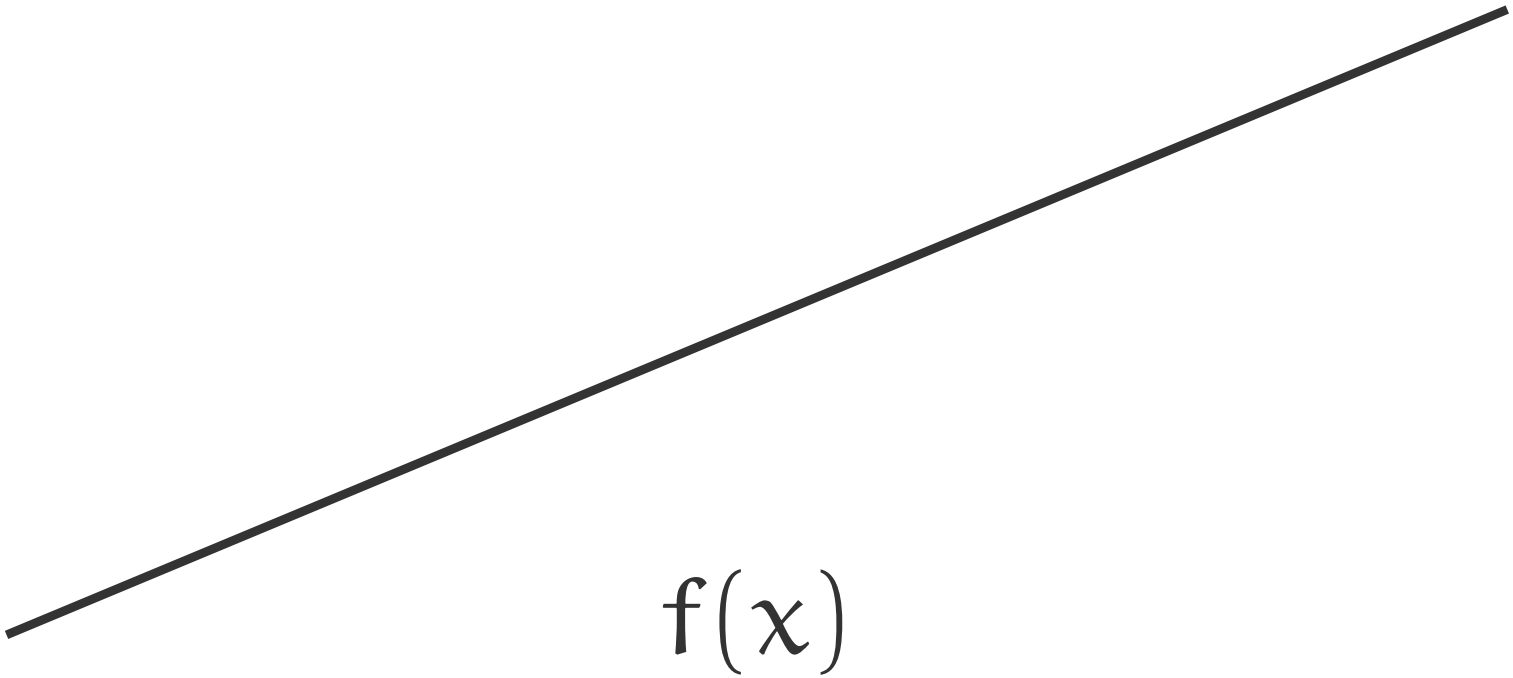
Estimate $f_h'(x)$ by

$$\widehat{f_h'}(x) = \frac{\widehat{f}(x+h) - \widehat{f}(x-h)}{2h}$$

$$\mathbb{E}\left[\widehat{f_h'}(x)\right] = f_h'(x) \qquad \mathbb{E}\left[(\widehat{f_h'}(x))^2\right] = \Theta\left(\frac{1}{h^2}\right)$$

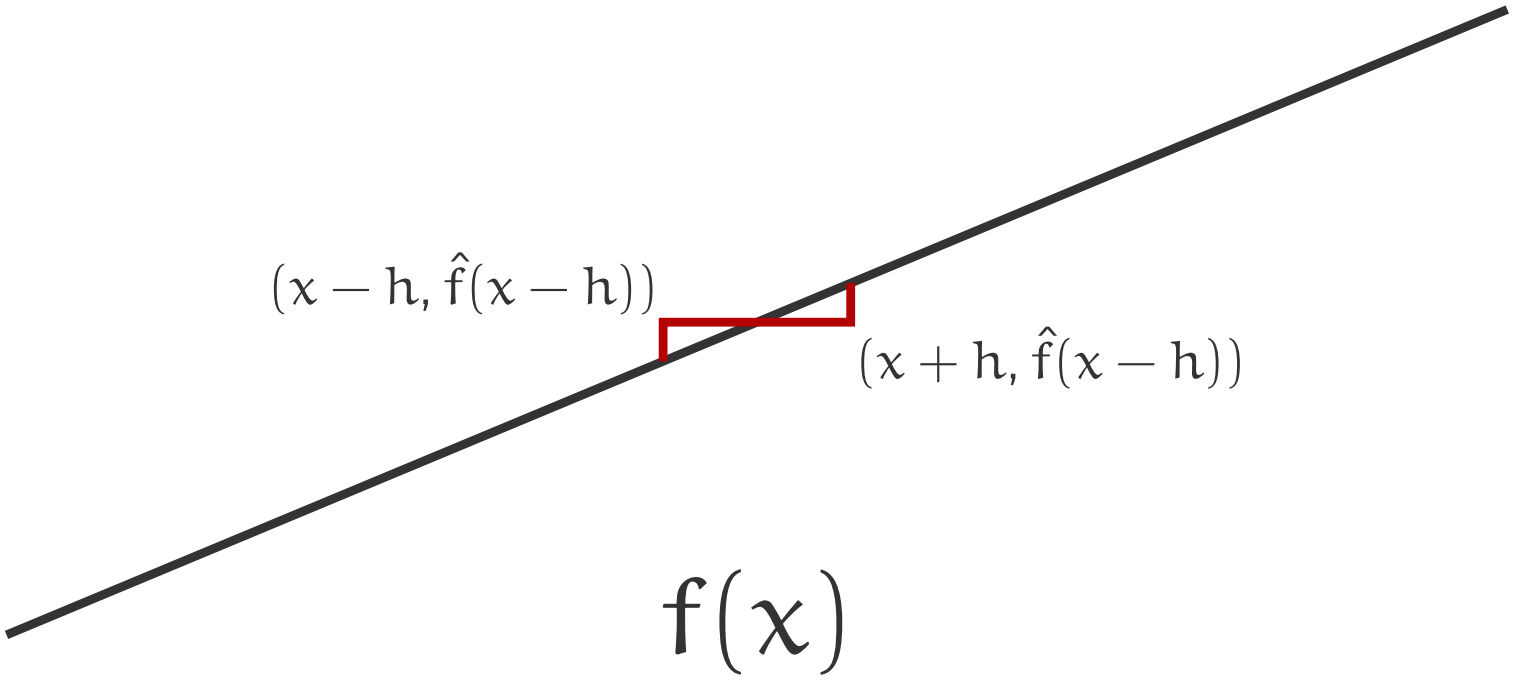
VARIANCE

Variance of gradient estimators is large when h is small



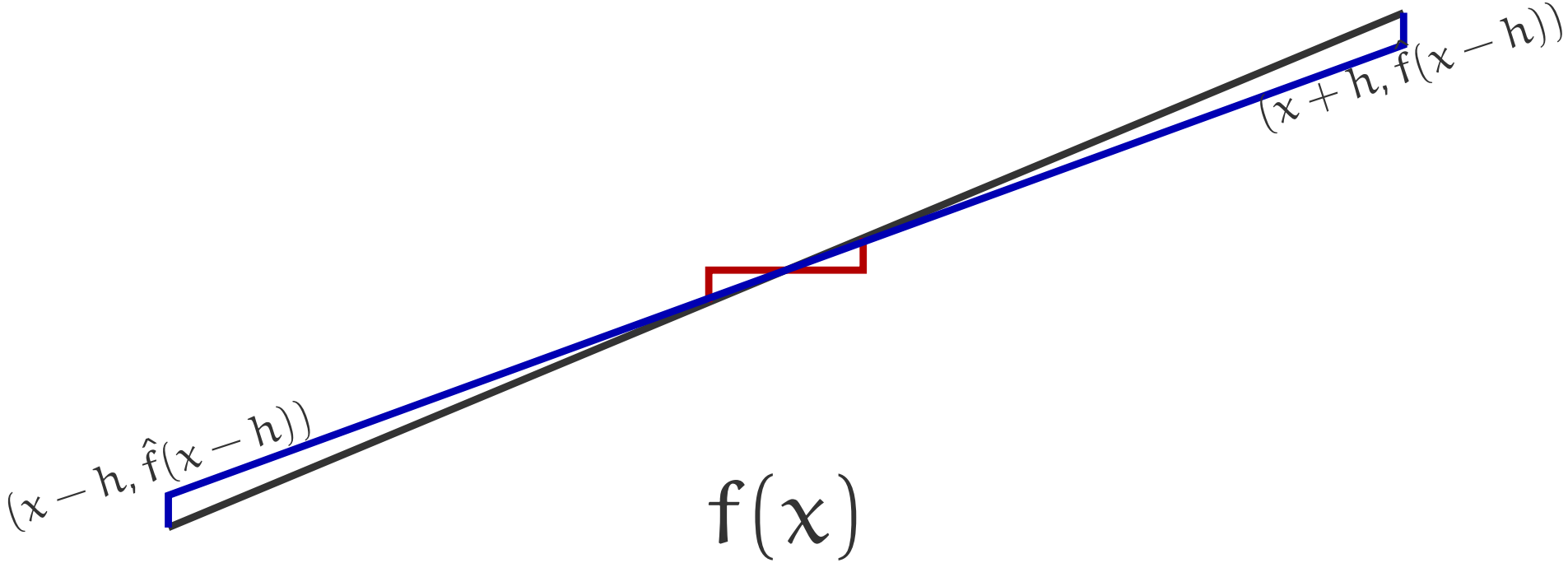
VARIANCE

Variance of gradient estimators is large when h is small



VARIANCE

Variance of gradient estimators is large when h is small



RANDOMISATION TRICK

$$\widehat{f}'_h(x) = \frac{\widehat{f}(x+h) - \widehat{f}(x-h)}{2h}$$

Queries the oracle twice

RANDOMISATION TRICK

$$\widehat{f}'_h(x) = \frac{\widehat{f}(x+h) - \widehat{f}(x-h)}{2h}$$

Queries the oracle twice

Sample u uniformly from $\{\pm h\}$

$$\widehat{f}'_h(x) = \frac{u\widehat{f}(x+u)}{h^2}$$

ALGORITHM

input: smoothing h and learning rate η

set $x_1 = 0$

for $t = 1$ to n:

  sample u from $\{\pm h\}$

  estimate gradient: $g_t = u\hat{f}(x_t + u)/h^2$

  update: $x_{t+1} = x_t - \eta g_t$

Flaxman et al. 2005, *Online Convex Optimization in the Bandit Setting: Gradient Descent Without a Gradient*

PLUGGING INTO THE THEORY

$$\begin{aligned}
\sum_{t=1}^{n} (f(x_t) - f(x_\star)) &\leqslant \sum_{t=1}^{n} (f_h(x_t) - f_h(x_\star)) + O(nh) \\
&\leqslant \frac{\|x_\star\|^2}{2\eta} + \frac{\eta}{2}\mathbb{E}\left[\sum_{t=1}^{n} \|\widehat{\nabla f_h}(x_t)\|^2\right] + O(nh) \\
&= \frac{\|x_\star\|^2}{2\eta} + O\left(\frac{\eta n}{h^2}\right) + O(nh) \\
&= O(n^{3/4})
\end{aligned}$$

SAMPLE COMPLEXITY $O\left(\frac{1}{\varepsilon^4}\right)$

ZEROTH-ORDER (STOCHASTIC) OPTIMISATION

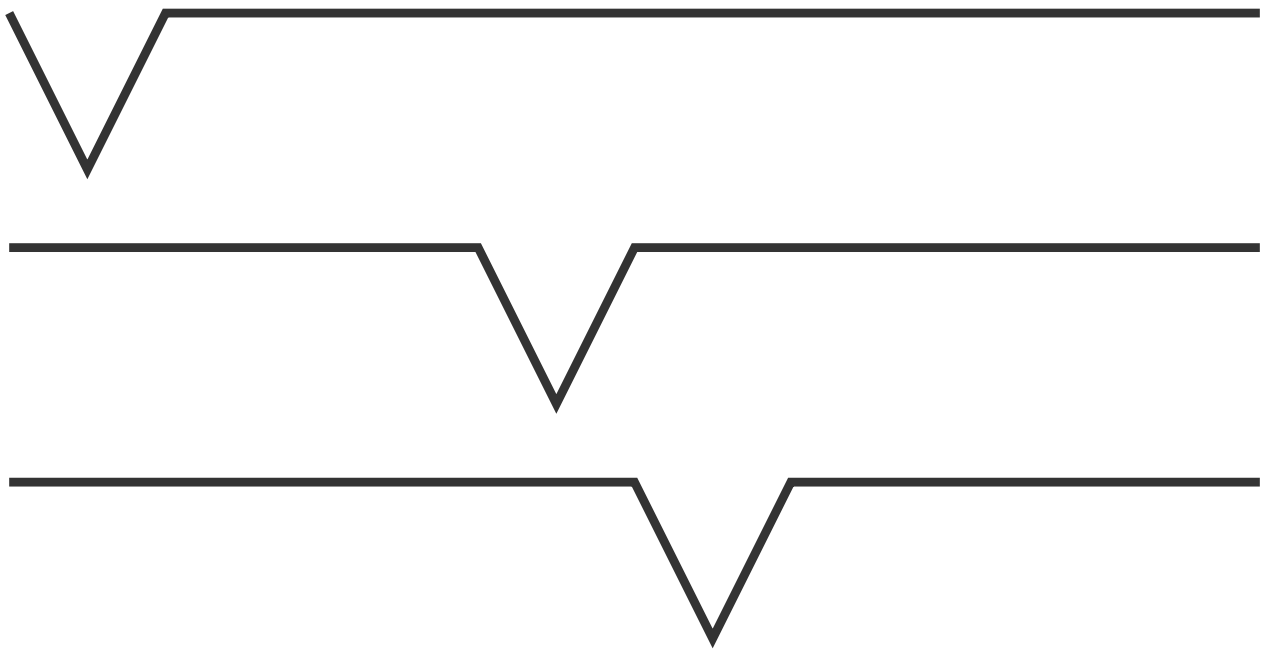
PART TWO

Tor Lattimore

Google DeepMind, London

LIFE WITHOUT CONVEXITY

- With convexity and noise free zeroth-order access we had $\log(1/\varepsilon)$ sample complexity
- What if we drop convexity?



- Sample complexity is $\mathrm{O}(1/\varepsilon)$

CURSE OF DIMENSIONALITY

Situation is **much worse** when d is large

Sample complexity for minimising Lipschitz loss functions without convexity is $\Omega((1/\varepsilon)^d)$

(STOCHASTIC) GRADIENT DESCENT

$x_1 = 0$

for $t = 1$ to n

Play x_t and observe $\widehat{\nabla} f(x_t)$

Update $x_{t+1} = x_t - \eta \widehat{\nabla} f(x_t)$

learning rate

ALGORITHM FOR ONE DIMENSION

input: smoothing h and learning rate η

set $x_1 = 0$

for $t = 1$ to n:

  sample u from $\{\pm h\}$

  estimate gradient: $g_t = u\hat{f}(x_t + u)/h^2$

  update: $x_{t+1} = x_t - \eta g_t$

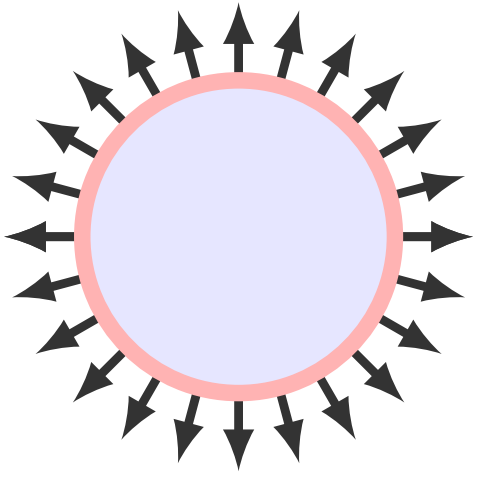
Flaxman et al. 2005, *Online Convex Optimization in the Bandit Setting: Gradient Descent Without a Gradient*

Stoke's theorem

Provided that $X \subset \mathbb{R}^d$ is nice enough,

$$\underbrace{\int_X \nabla f(x)\, dx}_{\text{volume integral}} = \underbrace{\int_{\partial X} f(x)\eta(x)\, dx}_{\text{surface integral}}$$

with $\eta(x)$ the outward-facing normal to ∂X at x



$$\int_a^b f'(x)\, dx = f(b) - f(a)$$

GENERALISING TO HIGHER DIMENSIONS

$$\mathbb{B}_h^d = \{u \in \mathbb{R}^d : \|u\| \leqslant h\} \quad \mathbb{S}_h^{d-1} = \{u \in \mathbb{R}^d : \|u\| = h\}$$

$$f_h(x) = \frac{1}{|\mathbb{B}_h^d|} \int_{\mathbb{B}_h^d} f(x+u)\, du$$

$$\nabla f_h(x) = \frac{1}{h|\mathbb{B}_h^d|} \int_{\mathbb{S}_h^{d-1}} u f(x+u)\, du \quad \text{[Stoke's theorem]}$$

Estimate $\nabla f_h(x)$ by sampling u uniformly from $\mathbb{S}_h^{d-1}$ and

$$\widehat{\nabla f_h}(x) = \frac{|\mathbb{S}_h^{d-1}|}{h|\mathbb{B}_h^d|} u f(x+u)$$

ALGORITHM FOR ONE DIMENSION

input: smoothing h and learning rate η

set $x_1 = 0$

for $t = 1$ to n:

  sample u from $\mathbb{S}^{d-1}(h)$

  estimate gradient: $g_t = u\hat{f}(x_t + u)\frac{|\mathbb{S}_h^{d-1}|}{h|\mathbb{B}_h^d|}$

  update: $x_{t+1} = x_t - \eta g_t$

Flaxman et al. 2005, *Online Convex Optimization in the Bandit Setting: Gradient Descent Without a Gradient*

SAMPLE COMPLEXITY

EXERCISE

$$\mathbb{E}\left[\sum_{t=1}^{n}(f(x_t)-f(x_\star))\right]=\tilde{O}(\sqrt{d}n^{3/4})$$

hint Use bound on regret of gradient descent and the fact that $\frac{|\mathbb{S}_h^{d-1}|}{|\mathbb{B}_h^d|}=\frac{d}{h}$

Sample complexity is $\tilde{O}\left(\frac{d^2}{\varepsilon^4}\right)$

NOTES

- Polynomial sample complexity
- Efficient and simple to implement
- Poor dependence on ε
- Various improvements to analysis and variants: [Saha and Tewari, 2011, Hazan and Levy, 2014, Ito, 2020]

A "SECOND-ORDER" ALGORITHM WITHOUT GRADIENTS OR HESSIANS

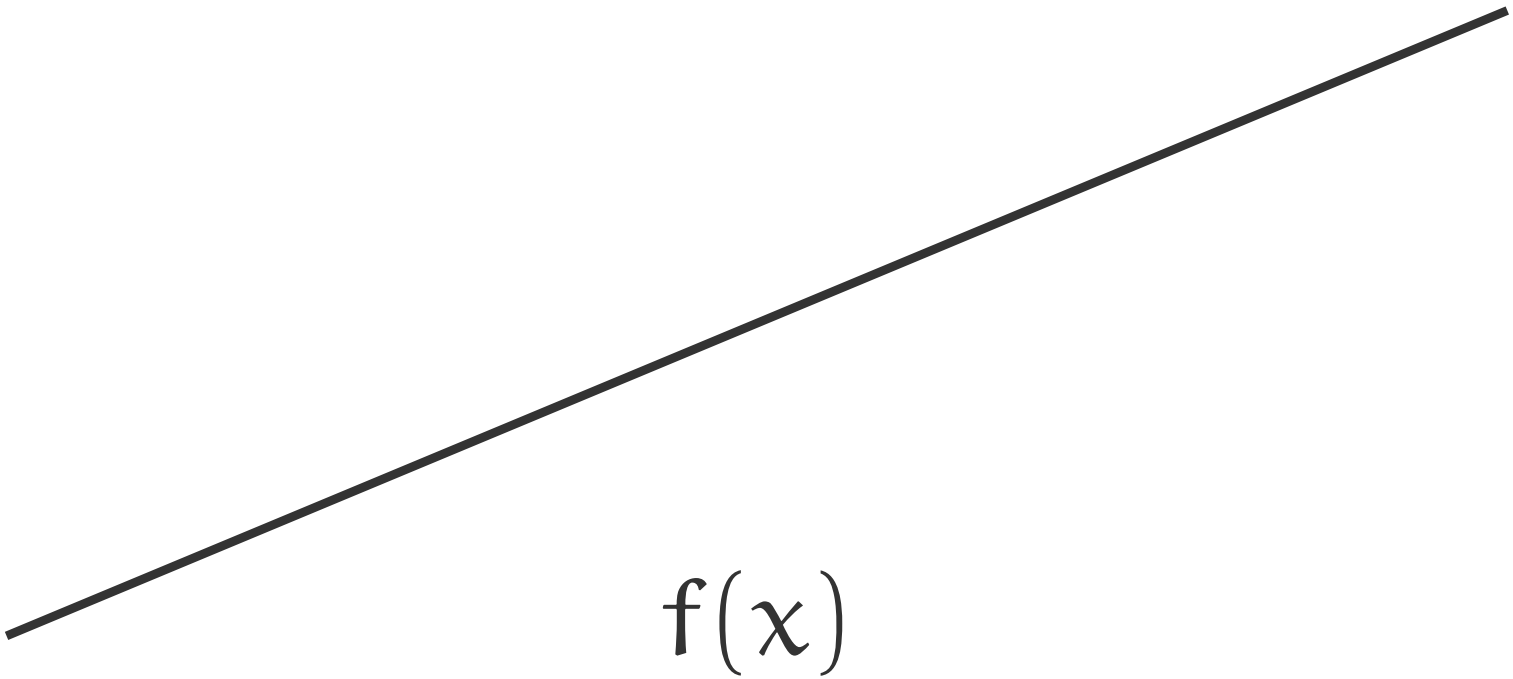
Curvature makes estimating gradients harder

Given gradients, curvature makes optimisation easier

Based on L and György [2023] appearing at COLT this year

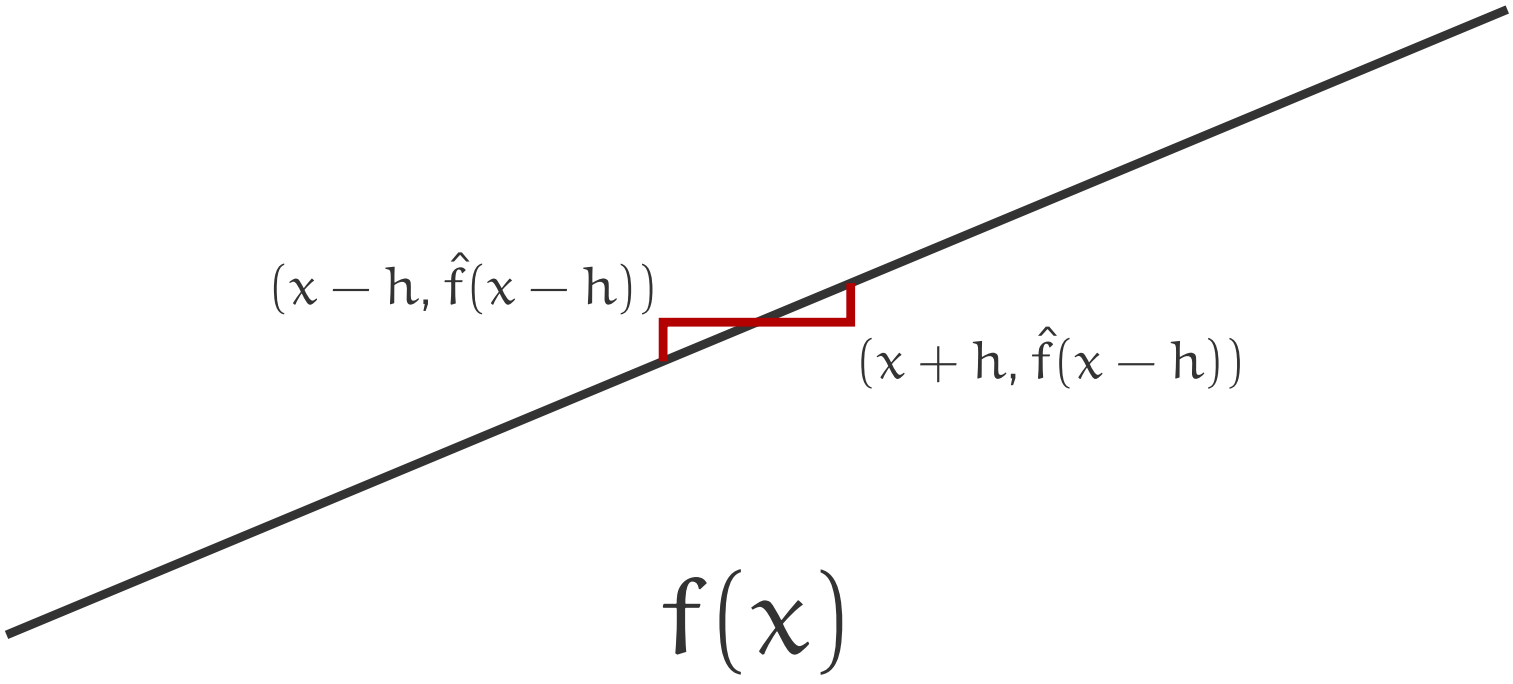
VARIANCE REVISITED

Variance of gradient estimators is large when h is small



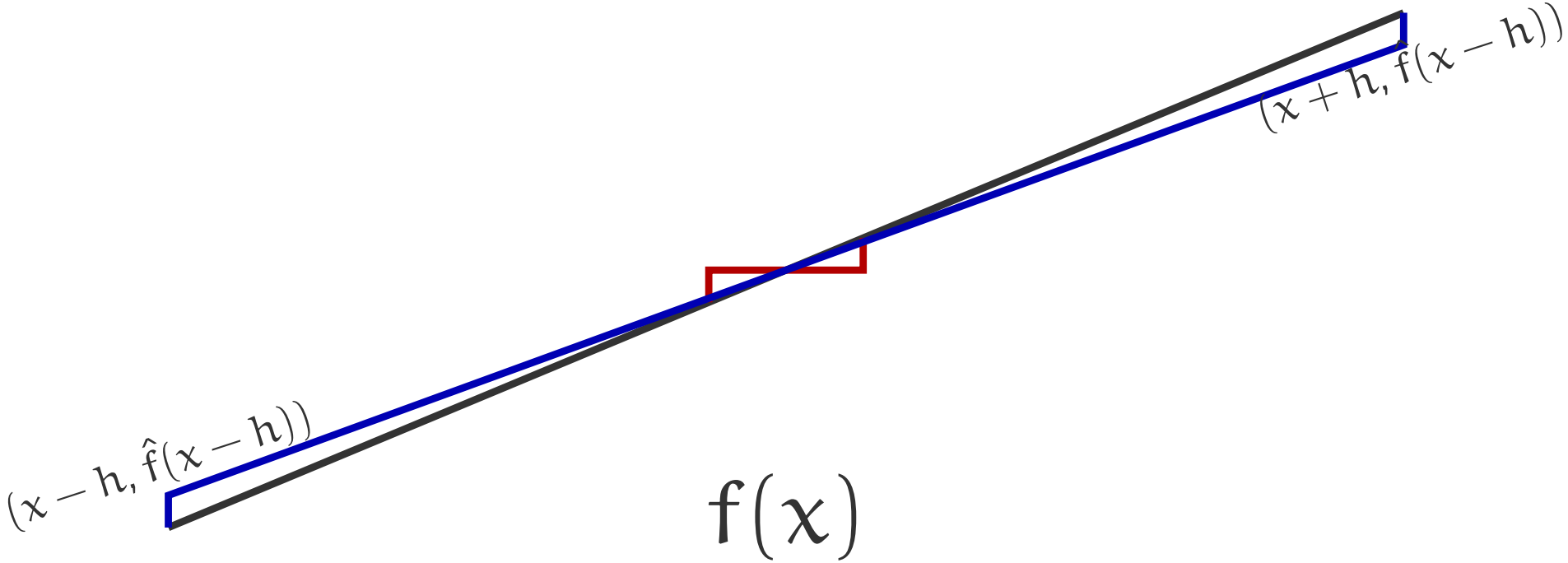
VARIANCE REVISITED

Variance of gradient estimators is large when h is small



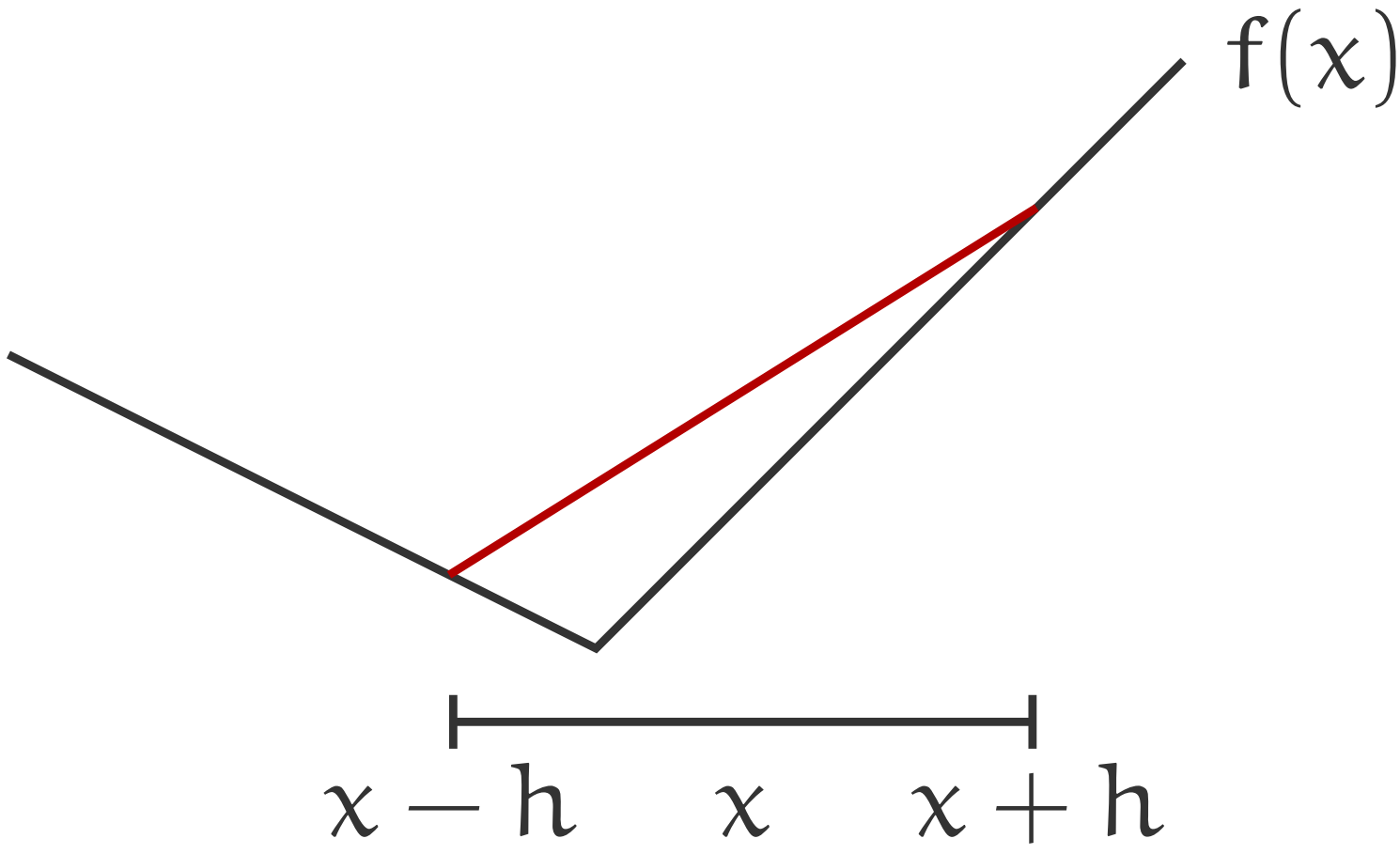
VARIANCE REVISITED

Variance of gradient estimators is large when h is small



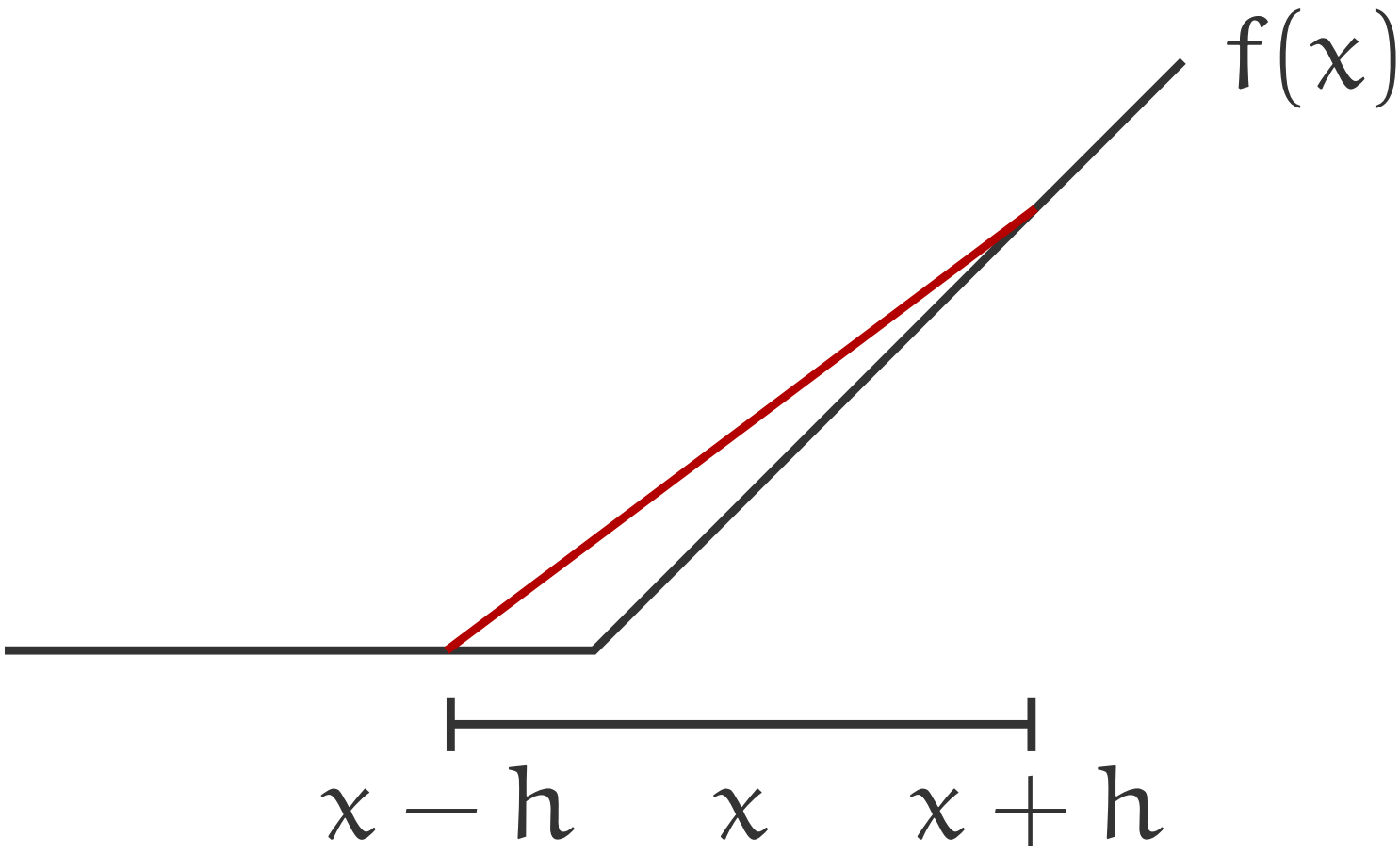
THE CURSE OF CURVATURE

When f has a lot of curvature, the bias of gradient estimator can be large

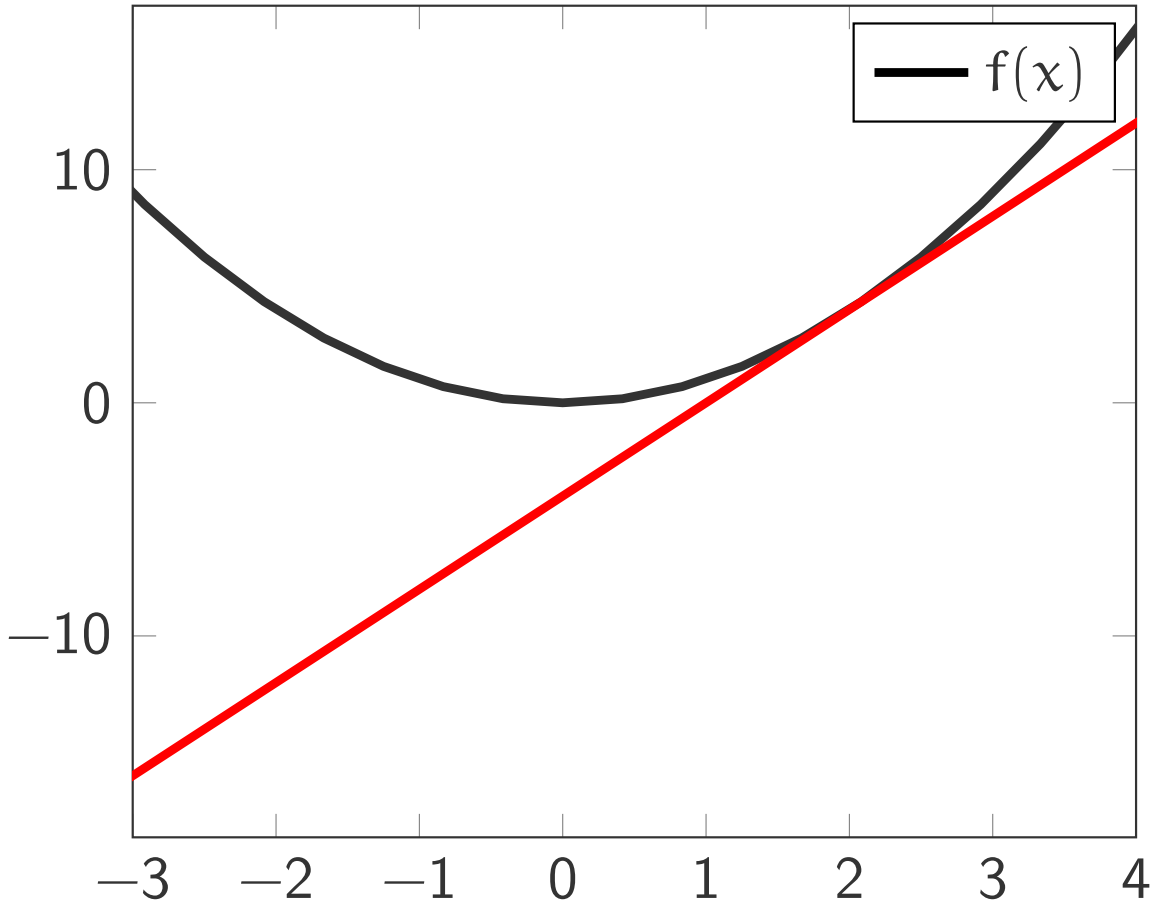


THE CURSE OF CURVATURE

When f has a lot of curvature, the bias of gradient estimator can be large

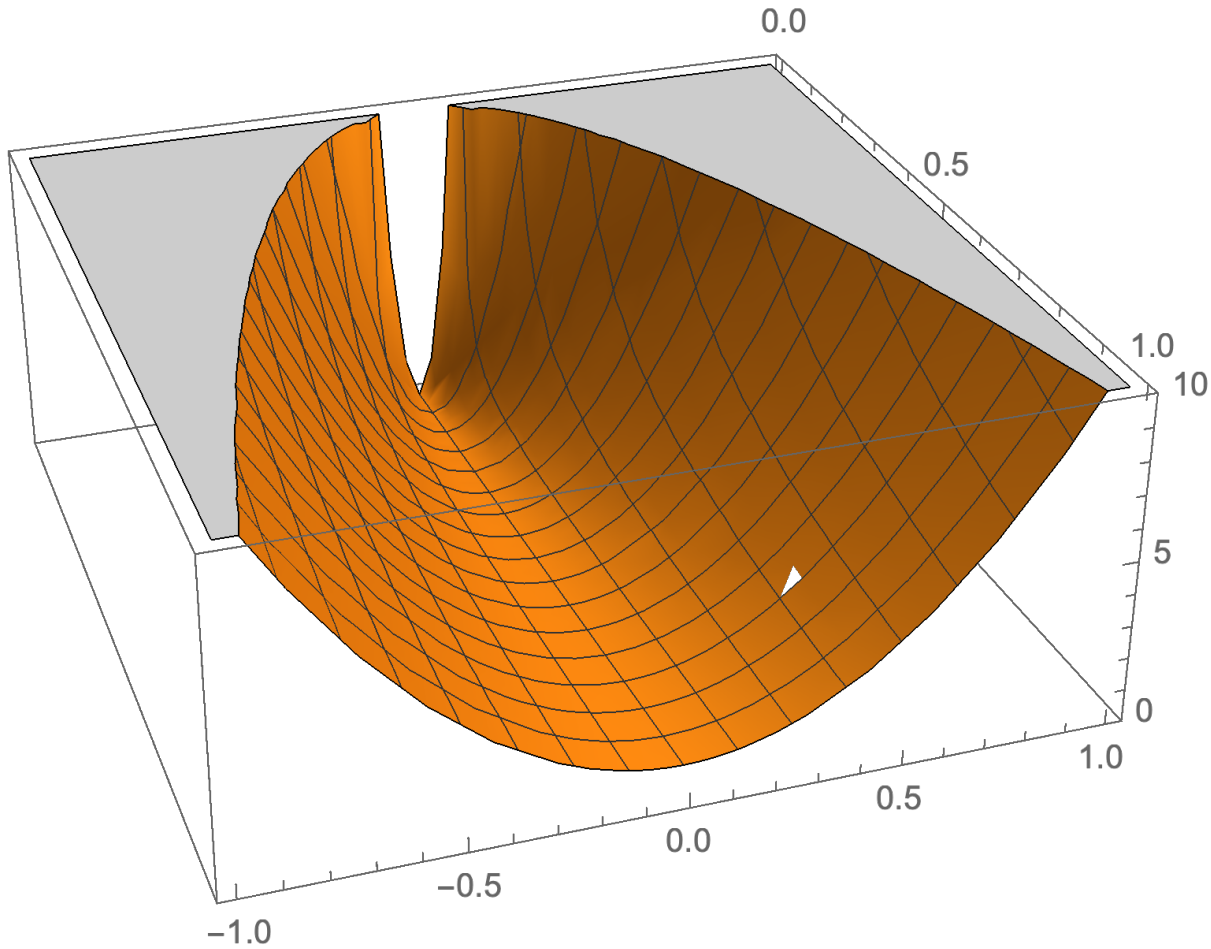


THE BLESSING OF CURVATURE



Loss is much smaller than linearisation suggests

CURVATURE IS COMPLICATED IN HIGH DIMENSIONS



ALGORITHMIC IDEA

- Smooth on an ellipsoidal *focus region*
- Estimate curvature
- Decrease focus region in directions where curvature is large

ALGORITHM

$\Sigma_t = 1$ and $\mu_t = 0$

for $t = 1$ to n:

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

Play x_t and observe $\hat{f}(x_t)$

$g_t = \hat{f}(x_t)\Sigma_t^{-1}(x_t - \mu_t)$

$H_t = \hat{f}(x_t)\Sigma_t^{-1}((x_t - \mu_t)(x_t - \mu_t)^\top \Sigma_t^{-1} - 1)$

$\Sigma_{t+1}^{-1} = \Sigma_t^{-1} + \lambda H_t$

$\mu_{t+1} = \mu_t - \eta \Sigma_{t+1}^{-1} g_t$

Smoothed gradient: $\mathbb{E}[g_t] = \mathbb{E}[\nabla f(x_t)]$

Smoothed Hessian: $\mathbb{E}[H_t] = \mathbb{E}[\nabla^2 f(x_t)]$

DEMO

THREE INTERPRETATIONS

GRADIENT DESCENT ON REPARAMETERISATION

- **STANDARD VIEW** Algorithm plays actions in $\mathbb{R}^d$. Gradient of loss with respect to parameter $x \in \mathbb{R}^d$ is $\nabla f(x)$ and gradient descent is

$$x_{t+1} = x_t - \eta \nabla f(x_t)$$

- **ALTERNATIVE** Algorithm samples actions from $\mathcal{N}(\mu_t, \Sigma_t)$. Play (stochastic) mirror descent on *parameters* μ and Σ with entropy regularisation

GRADIENT DESCENT ON REPARAMETERISATION

Gaussian density

$$p_{\mu,\Sigma}(x) = \left(\frac{1}{2\pi}\right)^{d/2} \sqrt{\det(\Sigma^{-1})} \exp\left(-\frac{1}{2}\|x-\mu\|^2_{\Sigma^{-1}}\right)$$

Gradient with respect mean

$$\nabla_\mu \int_{\mathbb{R}^d} f(x) p_{\mu,\Sigma}(x)\, dx = \int_{\mathbb{R}^d} f(x)\Sigma^{-1}(x-\mu) p_{\mu,\Sigma}(x)\, dx$$

Gradient with respect to inverse covariance

$$\nabla_{\Sigma^{-1}} \int_{\mathbb{R}^d} f(x) p_{\mu,\Sigma}(x)\, dx = -\int_{\mathbb{R}^d} f(x)\Sigma^{-1}((x-\mu)(x-\mu)^\top \Sigma^{-1} - \mathbf{1}) p_{\mu,\Sigma}(x)\, dx$$

ALGORITHM

$\Sigma_t = 1$ and $\mu_t = 0$

for $t = 1$ to n:

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

Play x_t and observe $\hat{f}(x_t)$

$g_t = \hat{f}(x_t)\Sigma_t^{-1}(x_t - \mu_t)$

$H_t = \hat{f}(x_t)\Sigma_t^{-1}((x_t - \mu_t)(x_t - \mu_t)^\top \Sigma_t^{-1} - 1)$

$\Sigma_{t+1}^{-1} = \Sigma_t^{-1} + \lambda H_t$

$\mu_{t+1} = \mu_t - \eta \Sigma_{t+1}^{-1} g_t$

Smoothed gradient: $\mathbb{E}[g_t] = \mathbb{E}[\nabla f(x_t)]$

Smoothed Hessian: $\mathbb{E}[H_t] = \mathbb{E}[\nabla^2 f(x_t)]$

CONTINUOUS EXPONENTIAL WEIGHTS AND ONLINE NEWTON STEP

- Continuous exponential weights samples x_t from p_t

$$p_t(x) \propto p_0(x) \exp\left(-\eta \sum_{s=1}^{t-1} \hat{f}_s(x)\right)$$

  where $p_0(x)$ is a standard Gaussian density

- Equivalent to previous algorithm when $\hat{f}_s$ is a quadratic approximation of f
- Algorithm is also equivalent to online Newton step with a quadratic approximation [Hazan et al., 2007]
- All explained in beautiful paper by van der Hoeven et al. [2018]

COMPUTATION COMPLEXITY

$\Sigma_t = 1$ and $\mu_t = 0$

for $t = 1$ to n:

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

Play x_t and observe $\hat{f}(x_t)$

$g_t = \hat{f}(x_t)\Sigma_t^{-1}(x_t - \mu_t)$

$H_t = \hat{f}(x_t)\Sigma_t^{-1}((x_t - \mu_t)(x_t - \mu_t)^\top \Sigma_t^{-1} - 1)$

$\Sigma_{t+1}^{-1} = \Sigma_t^{-1} + \lambda H_t$

$\mu_{t+1} = \mu_t - \eta \Sigma_{t+1}^{-1} g_t$

Memory is $O(d^2)$

Sampling from x_t	$O(d^3)$
Computing g_t and H_t	$O(d^2)$
Updating μ_t and Σ_t^{-1}	$O(d^2)$

THEORY

THEOREM Let

$$\eta = \tilde{\Theta}\left(\sqrt{\frac{d}{n}}\right) \qquad \lambda = \tilde{\Theta}\left(\frac{\eta}{d}\right)$$

Then with high probability

$$\sum_{t=1}^{n} \left(f(x_t) - f(x_\star)\right) = \tilde{O}\left(d^{1.5}\sqrt{n}\right)$$

Sample complexity is $\tilde{O}\left(\frac{d^3}{\varepsilon^2}\right)$

THOUGHTS

PROS

- The sample complexity is now $\mathrm{O}(1/\varepsilon^2)$
- Reasonable dimension-dependence

CONS

- Computation complexity is now $\mathrm{O}(\mathrm{d}^3)$
- Memory complexity is $\mathrm{O}(\mathrm{d}^2)$

COMPUTE VS SAMPLE COMPLEXITY

Algorithm	S.C.	Compute
Flaxman et al. [2005]	$\frac{d^2}{\varepsilon^4}$	d
L and György [2023]	$\frac{d^3}{\varepsilon^2}$	d^2

L and György [2023] $>$ Flaxman et al. [2005]

$$\iff \frac{d^5}{\varepsilon^2} \leqslant \frac{d^3}{\varepsilon^4}$$

$$\iff \varepsilon \leqslant \frac{1}{d}$$

CONSTRAINED SETTING

So far we only talked about $\mathcal{K} = \mathbb{R}^d$

How to relax this?

NEW SETTING

$\mathcal{K} \subset \mathbb{R}^d$ is closed and convex

Algorithm plays $x_t \in \mathcal{K}$, observes $\hat{f}(x_t)$

CONSTRAINED SETTING

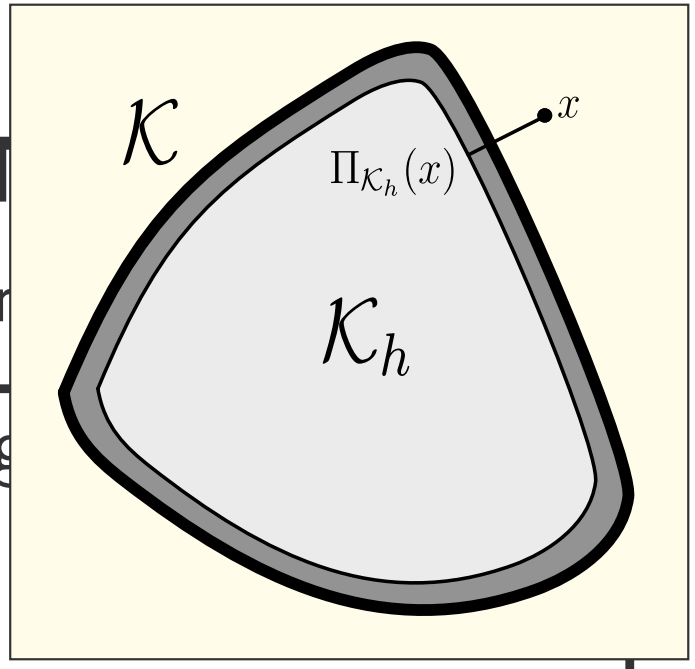
Algorithm of Flaxman et al. [2005] for the constrained setting:

input: smoothing h and learning rate η
set $x_1 = 0$
for $t = 1$ to n:
  sample u from $\mathbb{S}^{d-1}(h)$
  estimate gradient: $g_t = u d \hat{f}(x_t + u)/h^2$
  update: $x_{t+1} = \Pi_{\mathcal{K}_h}(x_t - \eta g_t)$

$$\mathcal{K}_h = \{x \in \mathcal{K} : x + \mathbb{B}^d(h) \in \mathcal{K}\} \quad \Pi_{\mathcal{K}_h}(x) = \operatorname*{arg\,min}_{y \in \mathcal{K}_h} \|x - y\|$$

CONSTRAINED SET

Algorithm of Flaxman et al. [2005] for the constr



input: smoothing h and learning

set $x_1 = 0$

for $t = 1$ to n:

  sample u from $\mathbb{S}^{d-1}(h)$

  estimate gradient: $g_t = u d \hat{f}(x_t + u)/h^2$

  update: $x_{t+1} = \Pi_{\mathcal{K}_h}(x_t - \eta g_t)$

$$\mathcal{K}_h = \{x \in \mathcal{K} : x + \mathbb{B}^d(h) \in \mathcal{K}\} \quad \Pi_{\mathcal{K}_h}(x) = \operatorname*{arg\,min}_{y \in \mathcal{K}_h} \|x - y\|$$

ADVERSARIAL SETTING

Adversary chooses $f_1, \ldots, f_n : \mathcal{K} \to [0, 1]$

Learner plays $x_t \in \mathcal{K}$ and observes $\hat{f}_t(x_t)$

Usual goal is to make the regret small

$$\sum_{t=1}^{n} (f_t(x_t) - f_t(x_\star))$$

with $x_\star = \arg\min_{x \in \mathcal{K}} \sum_{t=1}^{n} f_t(x)$

EXAMPLE – ONLINE LINEAR REGRESSION

Dataset of size n

$$\mathcal{D} = (x_t, y_t)_{t=1}^{n}$$

$x_t \in \mathbb{R}^d$ and $y_t \in \mathbb{R}$

$$f_t(\theta) = (\langle x_t, \theta \rangle - y_t)^2$$

MIRACLE

MIRACLE Algorithm of Flaxman et al. [2005] still works!

$$\mathbb{E}\left[\sum_{t=1}^{n}\left(f_t(x_t)-f_t(x_\star)\right)\right]=O(\sqrt{d}n^{3/4})$$

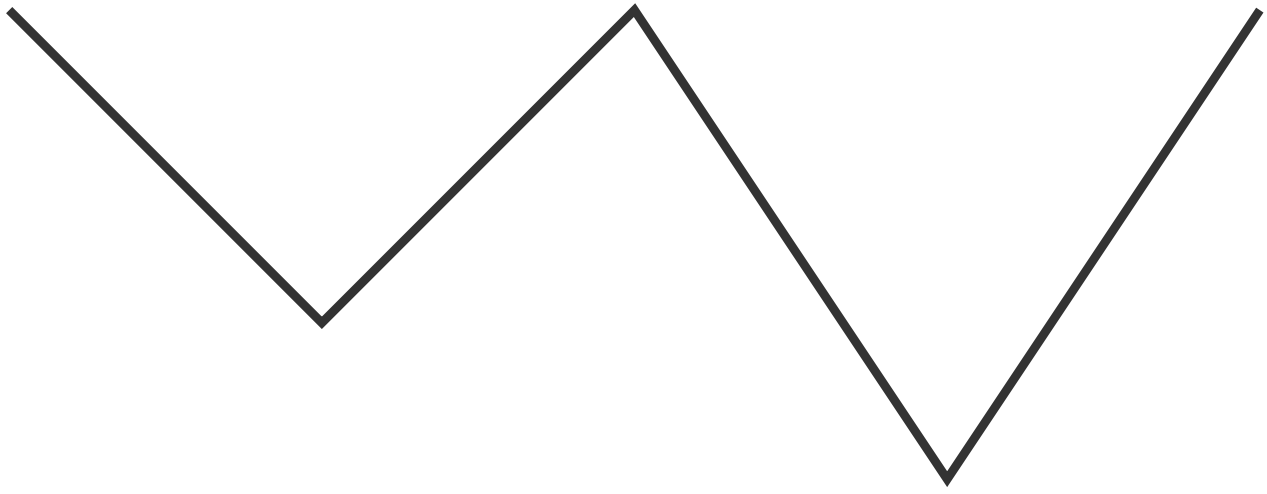
CONSTRAINED ADVERSARIAL SETTING

Algorithm of Flaxman et al. [2005] for the constrained adversarial setting:

input: smoothing h and learning rate η
set $x_1 = 0$
for $t = 1$ to n:
  sample u from $\mathbb{S}^{d-1}(h)$
  estimate gradient: $g_t = u d \hat{f}_t(x_t + u)/h^2$
  update: $x_{t+1} = \Pi_{\mathcal{K}_h}(x_t - \eta g_t)$

$$\mathcal{K}_h = \{x \in \mathcal{K} : x + \mathbb{B}^d(h) \in \mathcal{K}\} \quad \Pi_{\mathcal{K}_h}(x) = \operatorname*{arg\,min}_{y \in \mathcal{K}_h} \|x - y\|$$

NON-CONVEX PROBLEMS



Many gradient-based algorithms can find **local minima** efficiently [Roy et al., 2022]

WHAT'S MISSING?

- Really performant algorithms
- Lower bounds don't match upper bounds
- Understanding the resource/performance pareto frontier
- Adaptive algorithms

CURRENT STATE

Alg.	Setting	Constrained	S.C.	Compute
Flaxman et al. [2005]	Adv.	Yes	$\frac{d^2}{\varepsilon^4}$	$O(d)$
Bubeck et al. [2017]	Adv.	Yes	$\frac{d^{21}}{\varepsilon^2}$	polynomial
L [2020]	Adv.	Yes	$\frac{d^5}{\varepsilon^2}$	exponential
L and György [2021]	Stoch.	Yes	$\frac{d^9}{\varepsilon^2}$	$O(d)$
L and György [2023]	Stoch.	No	$\frac{d^3}{\varepsilon^2}$	$O(d^3)$

Wide range of results with more structure: [Kleinberg, 2005, Agarwal et al., 2010, Saha and Tewari, 2011, Hazan and Levy, 2014, Belloni et al., 2015, Ito, 2020, Luo et al., 2022, Suggala et al., 2021]

Best lower bound is $\Omega\left(\frac{d^2}{\varepsilon^2}\right)$ by Dani et al. [2008]

Questions and resources

- Open problems:
  - Do algorithms with $O(1/\varepsilon^2)$ sample complexity need $\Omega(d^2)$ memory?
  - What is the minimax sample complexity? $\in [\frac{d^2}{\varepsilon^2}, \frac{d^3}{\varepsilon^2}]$
- Books on bandits:
  - *Bandit algorithms*. L & Szepesvári
  - *Introduction to multi-armed bandits*. Slivkins
  - *Regret analysis of stochastic and nonstochastic multi-armed bandit problems*. Bubeck and Cesa-Bianchi
- Books on online learning
  - *A modern introduction to online learning*. Orabona
  - *Online learning*. Hazan
  - *Prediction, learning and games*. Cesa-Bianchi and Lugosi

A. Agarwal, O. Dekel, and L. Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *Colt*, pages 28–40. Citeseer, 2010.

A. Agarwal, D. P. Foster, D. Hsu, S. M. Kakade, and A. Rakhlin. Stochastic convex optimization with bandit feedback. *SIAM Journal on Optimization*, 23(1):213–240, 2013.

A. Belloni, T. Liang, H. Narayanan, and A. Rakhlin. Escaping the local minima via simulated annealing: Optimization of approximately convex functions. In *Conference on Learning Theory*, pages 240–265, 2015.

S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

S. Bubeck and R. Eldan. Exploratory distributions for convex functions. *Mathematical Statistics and Learning*, 1(1):73–100, 2018.

S. Bubeck, Y.T. Lee, and R. Eldan. Kernel-based methods for bandit convex optimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 72–85, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4528-6.

V. Dani, T. P. Hayes, and S. M. Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Conference on Learning Theory*, pages 355–366, 2008.

A Flaxman, A Kalai, and HB McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *SODA'05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394, 2005.

Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

E. Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

E. Hazan and K. Levy. Bandit convex optimization: Towards tight bounds. In *Advances in Neural Information Processing Systems*, pages 784–792, 2014.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69 (2-3):169–192, 2007.

S. Ito. An optimal algorithm for bandit convex optimization with strongly-convex and smooth loss. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2229–2239. PMLR, 26–28 Aug 2020.

R. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in Neural Information Processing Systems*, pages 697–704. MIT Press, 2005.

T. L. Improved regret for zeroth-order adversarial bandit convex optimisation. *Mathematical Statistics and Learning*, 2(3/4):311–334, 2020.

T. L and A. György. Improved regret for zeroth-order stochastic convex bandits. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 2938–2964. PMLR, 15–19 Aug 2021.

T. L and A. György. A second-order method for stochastic bandit convex optimisation. *arxiv*, 2023.

H. Luo, M. Zhang, and P. Zhao. Adaptive bandit convex optimization with heterogeneous curvature. In Po-Ling Loh and Maxim Raginsky, editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 1576–1612. PMLR, 02–05 Jul 2022.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. *arXiv preprint arXiv:2305.17333*, 2023.

Abhishek Roy, Krishnakumar Balasubramanian, Saeed Ghadimi, and Prasant Mohapatra. Stochastic zeroth-order optimization under nonstationarity and nonconvexity. *Journal of Machine Learning Research*, 23(64):1–47, 2022.

A. Saha and A. Tewari. Improved regret guarantees for online smooth convex optimization with bandit feedback. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 636–642, 2011.

A. Suggala, P. Ravikumar, and P. Netrapalli. Efficient bandit convex optimization: Beyond linear losses. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 4008–4067. PMLR, 15–19 Aug 2021.

D. van der Hoeven, T. van Erven, and W. Kot lowski. The many faces of exponential weights in online learning. In *Proceedings of*