How hard is this function to optimize?

John Duchi

Based on joint work with
Sabyasachi Chatterjee, John Lafferty, Yuancheng Zhu

Stanford University

West Coast Optimization Rumble – October 2016

Problem

$$\begin{aligned} &\text{minimize } f(x) := \mathbb{E}[F(x;\xi)] \\ &\text{subject to } x \in X. \end{aligned} \tag{1}$$

where $x \mapsto F(x;\xi)$ is convex, X is closed convex set

Problem

$$\begin{aligned} &\text{minimize } f(x) := \mathbb{E}[F(x;\xi)] \\ &\text{subject to } x \in X. \end{aligned} \tag{1}$$

where $x \mapsto F(x;\xi)$ is convex, X is closed convex set

Two questions:

1. How hard is it to solve problem (1) for *that* f
2. Can an algorithm(s) do as well as possible for each f?

Outline

- **Part 0** Complexity of problems
- **Part I** Complexity lower bounds
  - General lower bounds
  - Super-efficiency
- **Part II** Toward achievability?
- **Part III** Problem geometry and dimensionality

Problem Complexity

Large literature on guarantees of optimality

- Wald 1939, "Contributions to the theory of statistical estimation and testing hypotheses" (minimax complexity)
- Nemirovski and Yudin 1983, "Problem Complexity and Method Efficiency in Optimization" (information-based complexity)

Problem Complexity

Large literature on guarantees of optimality

- Wald 1939, "Contributions to the theory of statistical estimation and testing hypotheses" (minimax complexity)
- Nemirovski and Yudin 1983, "Problem Complexity and Method Efficiency in Optimization" (information-based complexity)

Three main considerations:

i. Information oracle (how we get information on problem)
ii. Problem class (what problems must the algorithm solve)
iii. How to measure error

Minimax error and oracles

Information oracle

- How algorithm/procedure receives information about problem
  - Optimization: function value $f(x)$ (zero-order), gradient $\nabla f(x)$ (first-order), Hessian $\nabla^2 f(x)$ (second-order)
  - Statistics: observations ξ_i from probability distribution P

Minimax error and oracles

Information oracle

- How algorithm/procedure receives information about problem
  - Optimization: function value $f(x)$ (zero-order), gradient $\nabla f(x)$ (first-order), Hessian $\nabla^2 f(x)$ (second-order)
  - Statistics: observations ξ_i from probability distribution P

Minimax principle Develop algorithm that has best *worst case* performance (risk) for problem class $\mathcal{F}$

$$R_N(\mathcal{F}) := \inf_{A \in \mathcal{A}_N} \sup_{f \in \mathcal{F}} \mathsf{error}(A, f)$$

where $\mathcal{A}_N$ is algorithms with N queries, $\mathcal{F}$ is problem class

Example

- $\mathcal{F}$ consist of 1-Lipschitz convex functions on $\mathbb{R}^d$
- Oracle returns $\nabla f(x) + \varepsilon$, where $\mathbb{E}[\varepsilon] = 0$ and $\|\varepsilon\| \leq 1$
- Domain X containts ℓ_∞-ball

Example

- $\mathcal{F}$ consist of 1-Lipschitz convex functions on $\mathbb{R}^d$
- Oracle returns $\nabla f(x) + \varepsilon$, where $\mathbb{E}[\varepsilon] = 0$ and $\|\varepsilon\| \leq 1$
- Domain X containts ℓ_∞-ball

Minimax rate: let $\widehat{x}_N$ be output of algorithm A after N steps

$$\inf_{A \in \mathcal{A}_N} \sup_{f \in \mathcal{F}} \mathbb{E}[f(\widehat{x}_N) - f(x^\star)] \asymp \frac{\sqrt{d}}{\sqrt{N}}.$$

(Agarwal et al. 12, Nemirovski & Yudin 83)

Example

- $\mathcal{F}$ consist of 1-Lipschitz convex functions on $\mathbb{R}^d$
- Oracle returns $\nabla f(x) + \varepsilon$, where $\mathbb{E}[\varepsilon] = 0$ and $\|\varepsilon\| \leq 1$
- Domain X containts ℓ_∞-ball

Minimax rate: let $\widehat{x}_N$ be output of algorithm A after N steps

$$\inf_{A \in \mathcal{A}_N} \sup_{f \in \mathcal{F}} \mathbb{E}[f(\widehat{x}_N) - f(x^\star)] \asymp \frac{\sqrt{d}}{\sqrt{N}}.$$

(Agarwal et al. 12, Nemirovski & Yudin 83)
More generally optimality guarantee: L-Lipschitz convex functions on sets X with diameter D, minimax rate

$$LD/\sqrt{N}$$

An optimal? algorithm

Algorithm: At iteration t

- Choose random ξ, set

$$g_t = \nabla F(x_t; \xi_i)$$

- Update

$$x_{t+1} = x_t - \alpha_t g_t$$

An optimal? algorithm

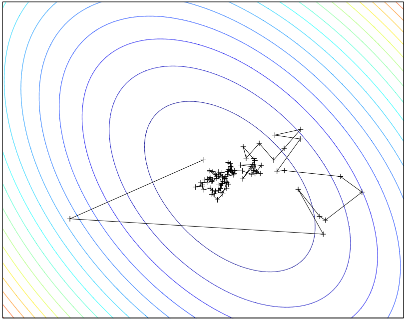
Algorithm: At iteration t

- Choose random ξ, set

$$g_t = \nabla F(x_t; \xi_i)$$

- Update

$$x_{t+1} = x_t - \alpha_t g_t$$



An optimal? algorithm

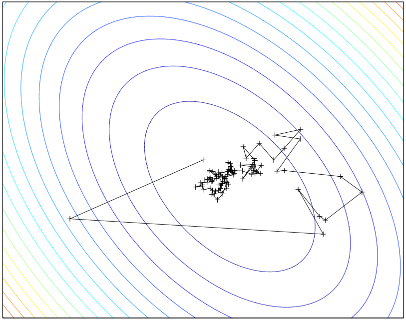
Algorithm: At iteration t

- Choose random ξ, set

$$g_t = \nabla F(x_t; \xi_i)$$

- Update

$$x_{t+1} = x_t - \alpha_t g_t$$



Theorem (Russians / Hungarians): Let $\widehat{x}_N = \frac{1}{N}\sum_{t=1}^{N} x_t$ and assume $D \geq \|x^\star - x_1\|_2$, $L^2 \geq \mathbb{E}[\|g_t\|_2^2]$. Then

$$\mathbb{E}[f(\widehat{x}_N) - f(x^\star) \leq \frac{LD}{\sqrt{N}}$$

Stochastic gradient descent

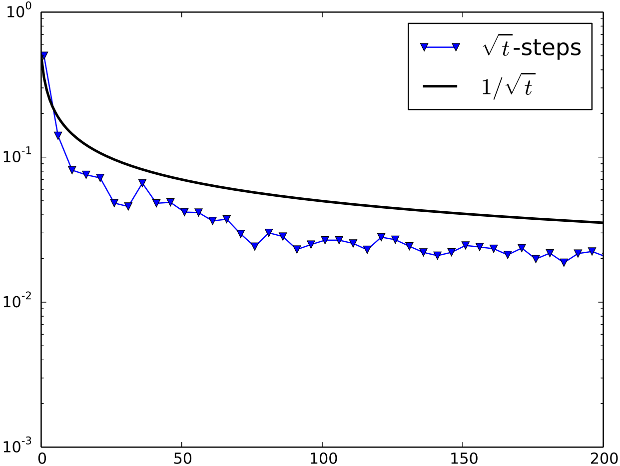
Let's use stochastic gradient descent to solve

$$\underset{x}{\text{minimize}} \;\; f(x) = \frac{1}{2}x^2 \quad \text{subject to} \;\; x \in [-1, 1]$$

Stochastic gradient descent

Let's use stochastic gradient descent to solve

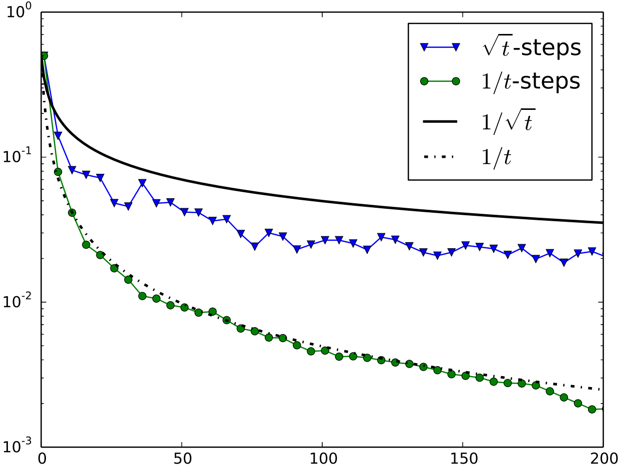
$$\underset{x}{\text{minimize}}\ \ f(x) = \frac{1}{2}x^2 \quad \text{subject to}\ \ x \in [-1, 1]$$



Stochastic gradient descent

Let's use stochastic gradient descent to solve

$$\underset{x}{\text{minimize}} \;\; f(x) = \frac{1}{2}x^2 \quad \text{subject to} \;\; x \in [-1, 1]$$



A local notion of complexity

Minimax complexity

$$R_N(\mathcal{F}) := \inf_{A \in \mathcal{A}_N} \sup_{f \in \mathcal{F}} \mathsf{error}(A, f)$$

A local notion of complexity

Minimax complexity

$$R_N(\mathcal{F}) := \inf_{A \in \mathcal{A}_N} \sup_{f \in \mathcal{F}} \mathsf{error}(A, f)$$

Local minimax complexity: Fix function f, and look for *hardest local alternative*

$$R_N(f; \mathcal{F}) := \sup_{g \in \mathcal{F}} \inf_{A \in \mathcal{A}_N} \max \left\{ \mathsf{error}(A, f), \mathsf{error}(A, g) \right\}$$

(Related ideas in statistics: Donoho & Liu 1987, 1991, Cai & Low 2015)

Local minimax complexity for stochastic optimization

- Noisy subgradient oracle: $\xi \sim \mathsf{N}(0, \sigma^2 I_{d\times d})$, return $f'(x) + \xi$
- Function class $\mathcal{F}$: convex functions (can restrict)
- Algorithm $\mathcal{A}_N$: all algorithms with N noisy subgradient queries
- Error metric $\mathsf{err} : X \times \mathcal{F} \to \mathbb{R}$

Local minimax complexity for stochastic optimization

- Noisy subgradient oracle: $\xi \sim \mathsf{N}(0, \sigma^2 I_{d\times d})$, return $f'(x) + \xi$
- Function class $\mathcal{F}$: convex functions (can restrict)
- Algorithm $\mathcal{A}_N$: all algorithms with N noisy subgradient queries
- Error metric $\mathsf{err} : X \times \mathcal{F} \to \mathbb{R}$

Local minimax complexity ($\widehat{x}_A$ is output of algorithm)

$$R_N(f; \mathcal{F}) := \sup_{g \in \mathcal{F}} \inf_{A \in \mathcal{A}_N} \max \left\{ \mathbb{E}_f \left[\mathsf{err}(\widehat{x}_A, f) \right], \mathbb{E}_g \left[\mathsf{err}(\widehat{x}_A, g) \right] \right\}.$$

Distances on functions and moduli of continuity

Distance for solutions Error must to satisfy *exclusion inequality*

$$\mathsf{err}(x, f) \leq d(f, g) \ \text{ implies } \ \mathsf{err}(x, g) \geq d(f, g).$$

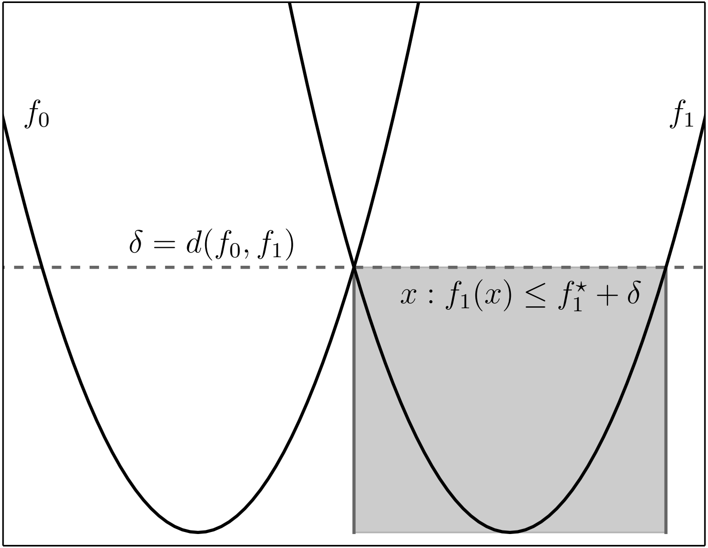
Example: how well one or other can be optimized

$$d(f_0, f_1) := \sup \left\{ \delta \geq 0 : \begin{array}{ll} f_1(x) \leq f_1^\star + \delta \ \text{ implies } \ f_0(x) \geq f_0^\star + \delta \\ f_0(x) \leq f_0^\star + \delta \ \text{ implies } \ f_1(x) \geq f_1^\star + \delta \end{array} \right\},$$

error

$$\mathsf{err}(x, f) = f(x) - f^\star$$

Separation



Other error metrics

Distance for solutions Error must to satisfy *exclusion inequality*

$$\mathsf{err}(x,f) \le d(f,g) \quad \text{implies} \quad \mathsf{err}(x,g) \ge d(f,g).$$

Example: distance to optimality

$$X_f^\star := \operatorname*{argmin}_{x\in X} f(x), \qquad X_g^\star := \operatorname*{argmin}_{x\in X} g(x)$$

$$d(f,g) = \mathsf{dist}(X_f^\star, X_g^\star)$$

Distances on functions

We study first-order methods, so define

$$\kappa(f, g) := \sup_{x \in X} \|f'(x) - g'(x)\|$$

Modulus of continuity

Given solution metric d and function metric κ, *modulus of continuity of d with respect to κ at f* is

$$\omega_f(\epsilon) := \sup_g \{d(f,g) : \kappa(f,g) \leq \epsilon\}$$

Modulus of continuity

Given solution metric d and function metric κ, *modulus of continuity of d with respect to κ at f* is

$$\omega_f(\epsilon) := \sup_g \{d(f,g) : \kappa(f,g) \le \epsilon\}$$

Examples: with

$$d(f,g) = |x_f^\star - x_g^\star|$$

Modulus of continuity

Given solution metric d and function metric κ, *modulus of continuity of d with respect to κ at f* is

$$\omega_f(\epsilon) := \sup_g \{d(f,g) : \kappa(f,g) \le \epsilon\}$$

Examples: with

$$d(f,g) = |x_f^\star - x_g^\star|$$

- Quadratic $f(x) = \frac{1}{2}x^2$,
  $\omega_f(\epsilon) = \epsilon$

Modulus of continuity

Given solution metric d and function metric κ, *modulus of continuity of d with respect to κ at f* is

$$\omega_f(\epsilon) := \sup_g \{d(f,g) : \kappa(f,g) \le \epsilon\}$$

Examples: with

$$d(f,g) = |x_f^\star - x_g^\star|$$

- Quadratic $f(x) = \frac{1}{2}x^2$,
  $\omega_f(\epsilon) = \epsilon$
- Power $f(x) = \frac{1}{k}|x|^k$,
  $\omega_f(\epsilon) = \epsilon^{\frac{1}{k-1}}$

Illustration of modulus of continuity

If $X \subset \mathbb{R}$,

$$\omega_f(\epsilon) = \sup_x \left\{ |x - x_f^\star| : |f'(x)| \le \epsilon \right\}$$

Main Theorem I

Theorem (Chatterjee, D., Lafferty, Zhu): Under Gaussian σ^2-noise, for (almost) any f

$$c\,\omega_f\left(\frac{\sigma}{\sqrt{N}}\right) \leq R_N(f;\mathcal{F}) \leq C\,\omega_f\left(\frac{\sigma}{\sqrt{N}}\right).$$

Main Theorem I

Theorem (Chatterjee, D., Lafferty, Zhu): Under Gaussian σ^2-noise, for (almost) any f

$$c\,\omega_f\left(\frac{\sigma}{\sqrt{N}}\right) \leq R_N(f;\mathcal{F}) \leq C\,\omega_f\left(\frac{\sigma}{\sqrt{N}}\right).$$

Modulus of continuity *precisely* determines rate of convergence

Proof intuition

If P_f = distribution when f is true, P_g = distribution when g is true

$$\max\left\{\mathbb{E}_f[\mathsf{err}(\widehat{x}, f)], \mathbb{E}_g[\mathsf{err}(\widehat{x}, g)]\right\} \geq \frac{1}{2}\mathbb{E}_f[\mathsf{err}(\widehat{x}, f)] + \frac{1}{2}\mathbb{E}_g[\mathsf{err}(\widehat{x}, g)]$$

Proof intuition

If $P_f =$ distribution when f is true, $P_g =$ distribution when g is true

$$\max\left\{\mathbb{E}_f[\mathsf{err}(\widehat{x}, f)], \mathbb{E}_g[\mathsf{err}(\widehat{x}, g)]\right\}$$
$$\geq \frac{d(f,g)}{2}\left[1 + P_g(\mathsf{err}(\widehat{x}, f) \geq d(f,g)) - P_f(\mathsf{err}(\widehat{x}, g) \geq d(f,g))\right]$$

Is this real?

$$c\,\omega_f\left(\frac{\sigma}{\sqrt{N}}\right) \le R_N(f;\mathcal{F}) \le C\,\omega_f\left(\frac{\sigma}{\sqrt{N}}\right).$$

Is this real?

$$c\,\omega_f\left(\frac{\sigma}{\sqrt{N}}\right) \le R_N(f;\mathcal{F}) \le C\,\omega_f\left(\frac{\sigma}{\sqrt{N}}\right).$$

- Is it possible to have a faster algorithm? (Were we too adversarial?)
- Is it too easy? (Were we not adversarial enough?)

You probably cannot be faster

Theorem (Chatterjee, D., Lafferty, Zhu): Let an algorithm $\widehat{x}_N$ satisfy

$$\mathbb{E}_f\left[\mathsf{err}(\widehat{x}_N, f)\right] \leq \delta\, \omega_f\left(\frac{\sigma}{\sqrt{N}}\right).$$

Then for

$$\epsilon_N = \sigma\sqrt{\frac{\log\frac{1}{\delta}}{N}}$$

and g one of $g_1(x) = f(x) - \epsilon_N x$, $g_{-1}(x) = f(x) + \epsilon_N x$,

$$\mathbb{E}_g\left[\mathsf{err}(\widehat{x}_N, g)\right] \geq c\omega_g\left(\sigma\sqrt{\frac{\log\frac{1}{\delta}}{N}}\right).$$

Is the local problem too easy?

Not in 1-dimension!

Sign-based binary search From interval $[a_0, b_0]$, for $k = 1, 2, \ldots$

- Query T points at $x_k = \frac{1}{2}(a+b)$ for gradients $G_1, \ldots, G_T$
- If $\sum_{t=1}^{T} G_t > 0$, set $b = x_k$ otherwise $a = x_k$

Proposition (Chatterjee, D., Lafferty, Zhu): After k epochs, define

$$\mathcal{I}_k := \left\{ x : |f'(x)| \le \frac{\sigma\sqrt{\log(k/\delta)}}{\sqrt{T}} \right\}$$

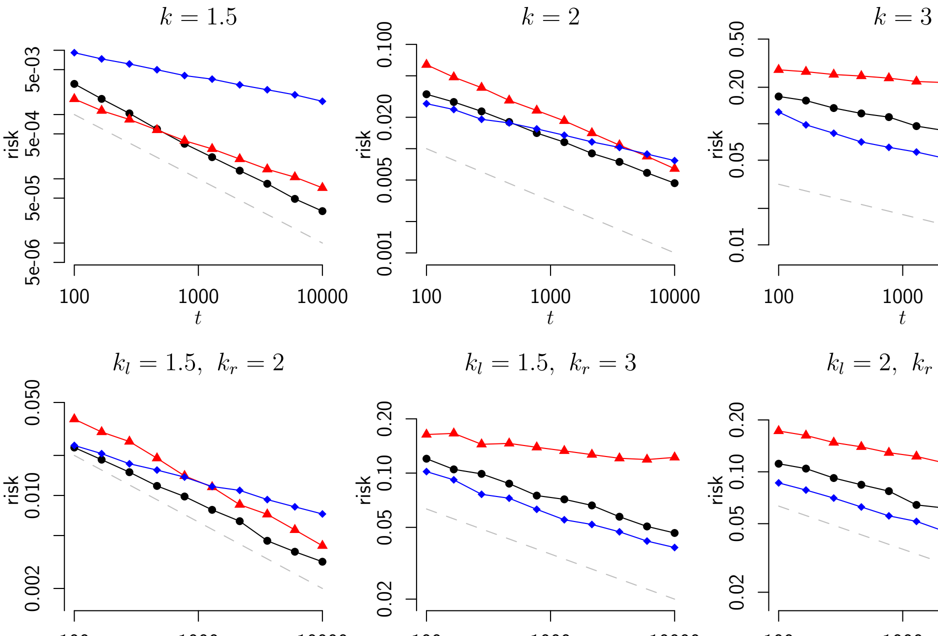
Then w.p. $\ge 1 - \delta$,

$$\mathsf{dist}(x_k, \mathcal{I}_k) \le 2^{-k}|b_0 - a_0| \le \widetilde{O}(1)\omega_f\left(\frac{\sigma}{\sqrt{N}}\right)$$

Intuition

Recall “flat set” $\{x : |f'(x)| \leq \epsilon\}$

Simulations



More achievability

Problem: In higher dimensions, we have some issues

More achievability

Problem: In higher dimensions, we have some issues
Note: an epoch-based stochastic gradient descent procedure of Juditsky and Nesterov (2014) is nearly adaptive (result coming soon)

More achievability

Problem: In higher dimensions, we have some issues
Note: an epoch-based stochastic gradient descent procedure of Juditsky and Nesterov (2014) is nearly adaptive (result coming soon)

Intuition: for $k = 1, 2, \ldots$,

- Perform stochastic gradient descent for $T_k \propto 2^k$ iterations with constant stepsize α_k
- Halve stepsize, double iteration length $T_{k+1} = 2T_k$, run again

More achievability

Problem: In higher dimensions, we have some issues
Note: an epoch-based stochastic gradient descent procedure of Juditsky and Nesterov (2014) is nearly adaptive (result coming soon)

Intuition: for $k = 1, 2, \ldots,$

- Perform stochastic gradient descent for $T_k \propto 2^k$ iterations with constant stepsize α_k
- Halve stepsize, double iteration length $T_{k+1} = 2T_k$, run again

One of these will be "correct" stepsize, and everything later will not allow any movement anyway

Part III: Problem Geometry

Coming soon... just a handwritten teaser if time.

Summary

- Local notions of minimax risk
  - Worst single alternative
  - Shrinking neighborhoods (suitably or poorly) defined
- Some optimal algorithms, but work to be done!

Some here: https://arxiv.org/abs/1605.07596, more soon...