Two Vignettes from Optimization

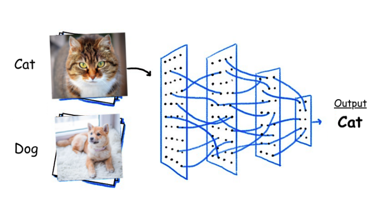
Qijia Jiang

April 2021

Stanford University

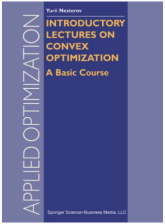
Motivation (for Studying Optimization)

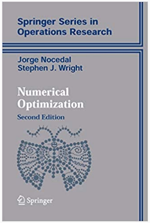
- Workhorse behind machine learning empirical success stories; applications across science & engineering domains – better understanding informs practice (e.g., hyper-parameter tuning)



Motivation (for Studying Optimization)

- Workhorse behind machine learning empirical success stories; applications across science & engineering domains – better understanding informs practice (e.g., hyper-parameter tuning)
- Modern computational model adds new dimension to the traditional setup (distributed, communication, synchrony, robustness etc.) – invite exploration for different trade-offs





Motivation (for Studying Optimization)

- Workhorse behind machine learning empirical success stories; applications across science & engineering domains – better understanding informs practice (e.g., hyper-parameter tuning)
- Modern computational model adds new dimension to the traditional setup (distributed, communication, synchrony, robustness etc.) – invite exploration for different trade-offs
- Tools can be brought to bear in a broader context – connections to MCMC sampling, online learning etc.

Outline

Goal: Complexity-theoretic investigations for various function classes under (non-classical) oracle models

- Higher Derivative Oracle $\Rightarrow$ Highly smooth functions
- Parallel Oracle $\Rightarrow$ Non-smooth functions

Higher-Order Acceleration for Highly-Smooth Functions

Smooth Function Minimization: Gradient Descent

Setting

- Assumption: (1) Lipschitz gradient, i.e., $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$; (2) $f : \mathbb{R}^d \to \mathbb{R}$ is convex & differentiable
- Gradient Oracle: access to $f(\cdot)$ and $\nabla f(\cdot)$ at any query point x
- Goal: find ϵ-optimal point, i.e., $f(x_k) - f^* \leq \epsilon$ (unconstrained)

Smooth Function Minimization: Gradient Descent

Assumption

(1) Lipschitz gradient, i.e., $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$; (2) $f : \mathbb{R}^d \to \mathbb{R}$ is convex & differentiable.

Gradient Descent

At each iteration k,

$$\begin{aligned} x_{k+1} &= x_k - \frac{1}{L}\nabla f(x_k) \\ &= \arg\min_y \left\{ f(x_k) + \langle \nabla f(x_k), y - x_k \rangle + \frac{L}{2}\|y - x_k\|^2 \right\} \end{aligned}$$

Error decreases as $\mathcal{O}(1/k)$, dimension-free.

Accelerated Gradient Descent

Accelerated Gradient Descent [Nesterov '83, Zhu & Orecchia '17]

1. At each iteration k, compute

$$a_{k+1} = \frac{1}{2}\left(1/L + \sqrt{1/L^2 + 4A_k/L}\right) \quad \text{and} \quad A_{k+1} = A_k + a_{k+1}$$

2. Set

$$\tilde{x}_k = \frac{A_k}{A_{k+1}} y_k + \frac{a_{k+1}}{A_{k+1}} x_k$$

3. Gradient Descent step:

$$y_{k+1} = \tilde{x}_k - \frac{1}{L}\nabla f(\tilde{x}_k) = \arg\min_y \left\{ \langle \nabla f(\tilde{x}_k), y - \tilde{x}_k \rangle + \frac{L}{2}\|y - \tilde{x}_k\|^2 \right\}$$

4. Mirror Descent step:

$$x_{k+1} = x_k - a_{k+1}\nabla f(\tilde{x}_k) = \arg\min_x \left\{ \langle a_{k+1}\nabla f(\tilde{x}_k), x - x_k \rangle + \frac{1}{2}\|x - x_k\|^2 \right\}$$

Error decrease as $\mathcal{O}(\frac{1}{k^2})$, although not a descent method.

Lower Bound for First-Order Oracle Model

Lower Bound [Nemirovski & Yudin '83]

Let $\{x_k\}$ be the sequence of points generated by any first-order method satisfying $x_k \in x_0 + \text{span}\{\nabla f(x_0), \cdots, \nabla f(x_{k-1})\}\ \forall k \geq 1$ and assume $f(\cdot)$ is L-smooth, convex, we have

$$f(x_k) - f^* \geq \Omega\left(\frac{L}{k^2}\right).$$

Higher Order Oracle Model

Setting

- Assumption: (1) f convex & differentiable; (2) p-th order smooth

$$\|\nabla^p f(x) - \nabla^p f(y)\| := \max_{\|v\|=1} \left|\nabla^p f(x)[v]^p - \nabla^p f(y)[v]^p\right| \leq L_p\|x - y\|$$

- p-th Order Oracle: access to $\{f(x), \nabla f(x), \cdots, \nabla^p f(x)\}$
- Goal: find ϵ-optimal point, i.e., $f(x_k) - f^* \leq \epsilon$
- Denote p-th order Taylor Expansion around x as

$$f_p(y; x) := f(x) + \sum_{i=1}^{p} \frac{1}{i!}\nabla^i f(x)[y - x]^i$$

Higher Order Oracle Model

Setting

- Assumption: (1) f convex & differentiable; (2) p-th order smooth
- p-th Order Oracle: access to $\{f(x), \nabla f(x), \cdots, \nabla^p f(x)\}$
- Goal: find ϵ-optimal point, i.e., $f(x_k) - f^* \leq \epsilon$

Lower Bound [Agarwal & Hazan '18, Nesterov '18]

Let $\{x_k\}$ be the sequence of points generated by some tensor method of degree p that satisfies mild assumption, have

$$\min_{0 \leq t \leq k} f(x_t) - f^* \geq \Omega\left(\frac{L_p}{k^{\frac{3p+1}{2}}}\right).$$

Higher Order Oracle Model

Setting

- Assumption: (1) f convex & differentiable; (2) p-th order smooth
- p-th Order Oracle: access to $\{f(x), \nabla f(x), \cdots, \nabla^p f(x)\}$
- Goal: find ϵ-optimal point, i.e., $f(x_k) - f^* \leq \epsilon$

Lower Bound [Agarwal & Hazan '18, Nesterov '18]

Let $\{x_k\}$ be the sequence of points generated by some tensor method of degree p that satisfies mild assumption, have

$$\min_{0 \leq t \leq k} f(x_t) - f^* \geq \Omega\left(\frac{L_p}{k^{\frac{3p+1}{2}}}\right).$$

Accelerated Rate [Nesterov & Polyak '08, Baes '09, Nesterov '18]

There is an algorithm that achieves convergence rate $\mathcal{O}(\frac{1}{k^{p+1}})$.

Coincide when $p = 1$. For $p = 2$: Accel Cubic-Regularized Newton.

The Iteration-Complexity Optimal Algorithm

1. Compute $\lambda_{k+1} > 0$ and $y_{k+1} \in \mathbb{R}^d$ such that

$$y_{k+1} = \arg\min_y \left\{ f_p(y; \tilde{x}_k) + \frac{L_p}{p!} \|y - \tilde{x}_k\|^{p+1} \right\} \tag{1}$$

$$\frac{1}{2} \le \lambda_{k+1} \frac{L_p \cdot \|y_{k+1} - \tilde{x}_k\|^{p-1}}{(p-1)!} \le \frac{p}{p+1} \tag{2}$$

   where

$$a_{k+1} = \frac{1}{2}\left(\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1} A_k}\right), \quad A_{k+1} = A_k + a_{k+1}$$

$$\tilde{x}_k = \frac{A_k}{A_{k+1}} y_k + \frac{a_{k+1}}{A_{k+1}} x_k \tag{3}$$

2. Update

$$x_{k+1} = x_k - a_{k+1} \nabla f(y_{k+1})$$

Convergence Guarantee [BJLLS, COLT'19]

Error decrease as $\tilde{\mathcal{O}}(k^{-\frac{3p+1}{2}})$, i.e., after $\tilde{\mathcal{O}}(\epsilon^{-\frac{2}{3p+1}})$ calls to tensor minimization blackbox that solves (1), we find an ϵ-optimal point.

The Iteration-Complexity Optimal Algorithm

1. Compute $\lambda_{k+1} > 0$ and $y_{k+1} \in \mathbb{R}^d$ such that

$$y_{k+1} = \arg\min_y \left\{ f_p(y; \tilde{x}_k) + \frac{L_p}{p!} \|y - \tilde{x}_k\|^{p+1} \right\} \tag{1}$$

$$\frac{1}{2} \leq \lambda_{k+1} \frac{L_p \cdot \|y_{k+1} - \tilde{x}_k\|^{p-1}}{(p-1)!} \leq \frac{p}{p+1} \tag{2}$$

where

$$a_{k+1} = \frac{1}{2}\left(\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1} A_k}\right), \quad A_{k+1} = A_k + a_{k+1}$$

$$\tilde{x}_k = \frac{A_k}{A_{k+1}} y_k + \frac{a_{k+1}}{A_{k+1}} x_k \tag{3}$$

Remark 1:
Approximate joint optimization over “interpolation parameter” and next query point y_{k+1}, i.e.,

$$\mathrm{prox}_f^p(y_k, x_k - y_k) := \arg\min_{y \in \mathbb{R}^d, \tau \in \mathbb{R}} \; f(y) + C \cdot \|y - (y_k + \tau(x_k - y_k))\|^{p+1}$$

implemented by p-th order Taylor expansion + 1D binary search.

The Iteration-Complexity Optimal Algorithm

1. Compute $\lambda_{k+1} > 0$ and $y_{k+1} \in \mathbb{R}^d$ such that

$$y_{k+1} = \arg\min_y \left\{ f_p(y; \tilde{x}_k) + \frac{L_p}{p!} \|y - \tilde{x}_k\|^{p+1} \right\} \tag{1}$$

$$\frac{1}{2} \leq \lambda_{k+1} \frac{L_p \cdot \|y_{k+1} - \tilde{x}_k\|^{p-1}}{(p-1)!} \leq \frac{p}{p+1} \tag{2}$$

where

$$a_{k+1} = \frac{1}{2} \left(\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1} A_k} \right), \quad A_{k+1} = A_k + a_{k+1}$$

$$\tilde{x}_k = \frac{A_k}{A_{k+1}} y_k + \frac{a_{k+1}}{A_{k+1}} x_k \tag{3}$$

2. Update

$$x_{k+1} = x_k - a_{k+1} \nabla f(y_{k+1})$$

Remark 2:
Proof based on refinement of estimate sequence technique.
[Monteiro & Svaiter '13]

Highly Parallel Oracle for Nonsmooth Functions

Setting

- Assumption: (1) f convex; (2) 1-Lipschitz: $|f(x) - f(y)| \leq \|x - y\|$
- First Order Oracle: query $x \in \mathbb{R}^d \Rightarrow$ return $f(x), \nabla f(x)$
- Goal: find ϵ-optimal point, i.e., $f(x_k) - f^* \leq \epsilon$

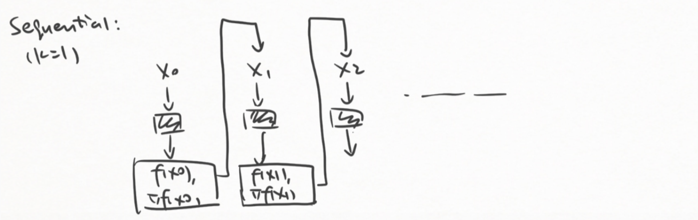
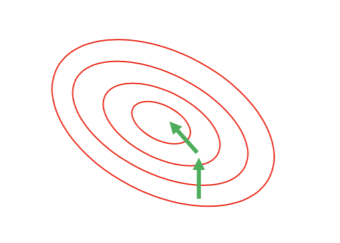
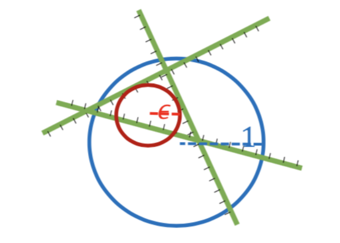


Figure 1: Schematic for Sequential Setup

Nonsmooth Optimization – Classical Model





Subgradient Descent
At iteration k,

$$x_{k+1} \leftarrow x_k - \eta \nabla f(x_k)$$

Output: $\bar{x}_K = \frac{1}{K} \sum x_k$

$\mathcal{O}(\frac{1}{\epsilon^2})$ queries suffice

Cutting Plane Methods
High-dimensional binary search (separation oracle implementable by subgradient oracle thanks to convexity)

$\mathcal{O}(d \log(\frac{1}{\epsilon}))$ queries suffice

Parallel Oracle

Allowed to submit K queries in parallel [Nemirovski '94].

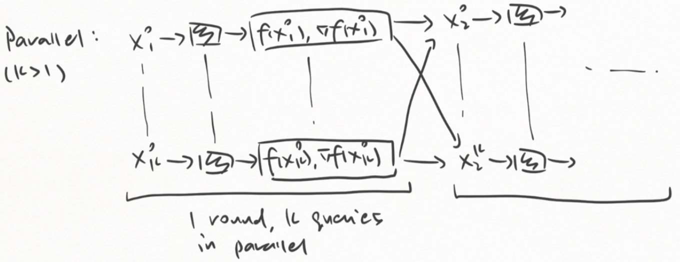


Figure 1: Schematic for Parallel Setup

Depth # queries to parallel oracle

Work total # gradients computed / functions evaluated

$$\text{work} = K \times \text{depth}$$

Parallel Oracle

Allowed to submit K queries in parallel [Nemirovski '94].

Lower bound (for K = 1):

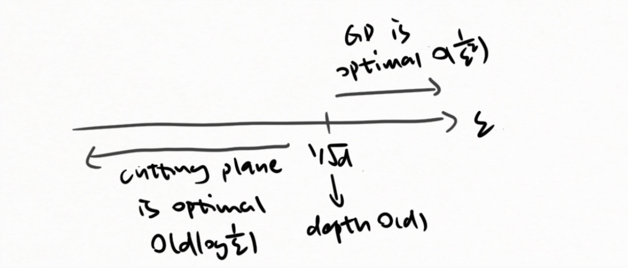


Figure 1: Upper & Lower Bound for Sequential

Question (for $K > 1$):
For $K = \text{poly}(d)$, best possible depth? Power of adaptive information for convex optimization?

Our Result

Parallel Complexity [BJLLS, NeurIPS '19]

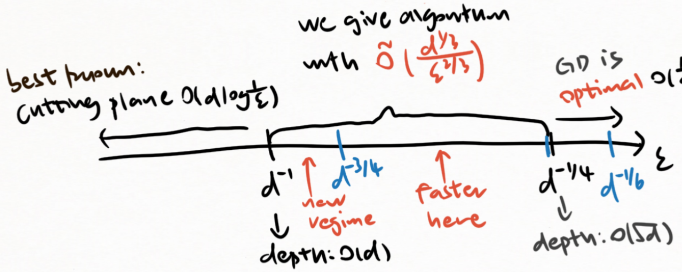


Figure 2: Upper & Lower Bound for Parallel

- $d^{-1/6}$ [Balanski & Singer '18]: GD is optimal for $\epsilon = \tilde{\omega}(d^{-1/6})$
- $d^{-3/4}$ [Duchi, Bartlett, Wainwright '12]: Accelerated stochastic method with depth $\mathcal{O}(\frac{d^{1/4}}{\epsilon})$
  $\rightarrow$ improve on sequential for $\epsilon \in [d^{-3/4}, d^{-1/4}]$, depth $\in [\sqrt{d}, d]$

Our Result

How?

1. Convolve the nonsmooth function f with Gaussian kernel γ_r, i.e., $g(y) = \int f(y-x)\gamma_r(x)\,dx$. Tradeoff between approximation vs. smoothness

$$|g(y) - f(y)| \leq r\sqrt{d} \quad \textit{and} \quad \|\nabla^2 g(y)\|_{\mathrm{op}} \leq \frac{1}{r}$$

2. Build inexact gradient oracle via sampling[1], use to minimize the "prox" step approximately in depth=1

$$\tilde{\nabla} g(y) = \sum_{x_i} \gamma_r(y - x_i)\nabla f(x_i) \quad \textit{and} \quad \min_y \, g(y) + Q(\|y - c\|)$$

3. Apply highly smooth acceleration result on the smoothed $g(\cdot)$

[1] K ends up being $\tilde{\mathcal{O}}(d/\epsilon^2)$

A More General Framework Emerges

Extend to more general "prox" function than $\|\cdot\|^p$ – access to a subroutine that can approximately minimize

$$\arg\min_y \; f(y) + Q(\|x-y\|)$$

gives a systematic way of getting accelerated rate for the fcn class.

- Approximate Proximal Point: $\approx \arg\min_y f(y) + L/2\|x-y\|^2$
- p-th order Acceleration: $= \arg\min_y f_p(y;x) + L_p/p!\|x-y\|^{p+1}$
- Highly Parallel: $\approx \arg\min_y g(y) + C\|x-y\|^{p+1}$
- Ball-oracle [NeurIPS '20]: $\approx \arg\min_{y:\|x-y\|\le r} f(y)$
  efficiently implementable by trust-region for Hessian-stable functions, i.e., $c^{-1}\nabla^2 f(y) \preceq \nabla^2 f(x) \preceq c\nabla^2 f(y)$ for all $\|x-y\| \le r$

A More General Framework Emerges

Extend to more general "prox" function than $\|\cdot\|^p$ – access to a subroutine that can approximately minimize

$$\arg\min_y \; f(y) + Q(\|x - y\|)$$

gives a systematic way of getting accelerated rate for the fcn class.

- Approximate Proximal Point: $\approx \arg\min_y f(y) + L/2\|x - y\|^2$
- p-th order Acceleration: $= \arg\min_y f_p(y; x) + L_p/p!\|x - y\|^{p+1}$
- Highly Parallel: $\approx \arg\min_y g(y) + C\|x - y\|^{p+1}$
- Ball-oracle [NeurIPS '20]: $\approx \arg\min_{y:\|x-y\|\le r} f(y)$
  efficiently implementable by trust-region for Hessian-stable functions, i.e., $c^{-1}\nabla^2 f(y) \preceq \nabla^2 f(x) \preceq c\nabla^2 f(y)$ for all $\|x - y\| \le r$

Deeper implication for complexity theory – Better iteration complexity implemented by lower-order method.

Thanks!