Introduction to Optimization Theory

Lecture #3 - 9/22/20
MS&E 213 / CS 2690

Aaron Sidford
sidford@stanford.edu
Lecture Plan

**Recap**
- Oracles, minimization, efficiency, and iterative methods
- Continuity, smoothness, and critical points

**Material**
- Continuity, $\epsilon$-nets, and lower bounds

**Thursday**
- Smoothness revisited
- Convexity
Recap

**Goal**
- Objective function \( f: \mathbb{R}^n \rightarrow \mathbb{R} \)
- Constraint set \( S \subseteq \mathbb{R}^n \)
  (Next few lectures, unconstrained \( S = \mathbb{R}^n \))
- Optimize

\[
\min_{x \in S \subseteq \mathbb{R}^n} f(x)
\]

provably efficiently with few assumptions

**Access to f?**
- Through an “oracle”

query
e.g. \( x \in \mathbb{R}^n \)

↓

**oracle**

↓

output
e.g. \( f(x) \in \mathbb{R} \) [value]
e.g. \( \nabla f(x) \) [gradient]
Recap

Minimize? Progress Measure?

**ε-(sub)optimal point** or a point with **ε-additive function error**:

- \( x \in S \text{ s.t. } f(x) \leq f_* + \epsilon \text{ where } f_* = \min_{x \in S} f(x) \)

**ε-critical point**:

- \( x \in S \text{ s.t. } \| \nabla f(x) \|_2 \leq \epsilon \text{ where } \| y \|_2 = \sqrt{\sum_{i \in [n]} y_i^2} \)

Efficency?

- Oracle complexity = # calls to oracle
- Runtime = # oracle calls \( \times \) (average computational cost per call)
Recap

Iterative Method Approach
• Start at initial point $x_0$
• For $t = 0, \ldots, T - 1$
  • Query oracle
  • Take “local step” to obtain $x_{t+1}$
  • Repeat
• Output aggregation of the $x_t$

e.g.
• Last iterate: $x_{T-1}$
• Average iteration: $\frac{1}{T} \sum_{k \in [T-1]} x_k$

Analysis
• Oracle complexity = # iterations
• Runtime = # iterations * cost per iteration (iteration complexity)
Recap: setting #0: impossible

- \( f : \mathbb{R} \to \mathbb{R} \) (one dimensional)
- Have evaluation oracle (can compute \( f(x) \) with 1 query)
- Promised \( \exists x* \in [0,1] \) such that \( f(x) = f_* = \inf_{y \in \mathbb{R}} f(y) \)
- Promised \( f(x) \in [0,1] \) for all \( x \in \mathbb{R} \)
- Goal: compute 1/2-optimal point
  - i.e. compute \( x \) with \( f(x) \leq f(x*) + 1/2 \)

**Question:** what oracle complexity achievable?

**Answer:** \( \infty \) is optimal

We will discuss this lower bound more formally today.
Recap

**Problem**: oracle gives only pointwise information, no local information.

**Solution**:
- This is a class on *continuous* optimization
- **Today**: assume more structure and analyze a working method

Last class discussed how continuity is not enough and will prove today.
Recap: assuming more structure

\[ f \text{ is } L_1\text{-Lipschitz w.r.t. } \| \cdot \| \]
\[
|f(x) - f(y)| \leq \|x - y\|
\]
for all \( x, y \in \mathbb{R}^n \)

\[ f \text{ is } L_2\text{-Lipschitz} \]
\[
\|\nabla f(x) - \nabla f(y)\|_2 \leq L_2 \|x - y\|_2
\]
for all \( x, y \in \mathbb{R}^n \)

(bounded slope)

(bounded 1st derivatives)

(bounded curvature)

(bounded 2nd derivative)
Recap: Gradient Descent Method for Critical Points

**Algorithm / Method** (for $L$-smooth $f$)

- Initial point: $x_0 \in \mathbb{R}^n$
- For $k = 0, 1, 2, \ldots$
  - $x_{k+1} = x_k - \frac{1}{L} \nabla f(x_k)$
  - If $\|\nabla f(x_k)\|_2 \leq \epsilon$ then output $x_k$

**Theorem**

$\epsilon$-critical point in $\leq 2L[f(x_0) - f_*]/\epsilon^2$

steps / queries for $f_* = \inf_{x \in \mathbb{R}^n} f(x)$

**Today:** $\epsilon$-(sub)optimal points
Lecture Plan

Recap
- Oracles, minimization, efficiency, and iterative methods
- Continuity, smoothness, and critical points

Material
- Continuity, $\epsilon$-nets, and lower bounds

Thursday
- Smoothness revisited
- Convexity
Setting #1: 1d-Lipschitz Function Minimization

- \( f : \mathbb{R} \to \mathbb{R} \) (one dimensional)
- Have evaluation oracle (can compute \( f(x) \) with 1 query)
- \( \exists x_* \in [0,1] \) such that \( f(x) = f_* = \inf_{y \in \mathbb{R}} f(y) \)
- \( f(x) \in [0,1] \) for all \( x \in \mathbb{R} \)
- \( f \) is \( L \)-Lipschitz with respect to \( \ell_\infty \)
- **Goal**: compute \( \epsilon \)-optimal point for \( \epsilon \in (0,1) \)

- **Question #1**: what oracle complexity achievable?
- **Question #0**: what does \( L \)-Lipschitz mean? Imply?
L-Lipschitz Function

$f$ is $L$-Lipschitz w.r.t. $\| \cdot \|$ if $|f(x) - f(y)| \leq L\|x - y\|$ for all $x, y \in \mathbb{R}^n$

- $\Leftrightarrow -L\|x - y\| \leq f(y) - f(x) \leq L\|x - y\|$ for all $x, y \in \mathbb{R}^n$
- $\Leftrightarrow f(x) - L\|x - y\| \leq f(y) \leq f(x) + L\|x - y\|$ for all $x, y \in \mathbb{R}^n$
- If $n = 1$ and $\| \cdot \| = \| \cdot \|_p$ (i.e. $\|x\| = \|x\|_p = (|x|^p)^{1/p} = |x|$) then
  $\Leftrightarrow f(x) - L|d| \leq f(x + d) \leq f(x) + L|d|$ (slope at most $L$)

Value of $f$ lies in this range
Setting #1: 1d-Lipschitz Function Minimization

- $f : \mathbb{R} \to \mathbb{R}$ (one dimensional)
- Have evaluation oracle (can compute $f(x)$ with 1 query)
- $\exists x^* \in [0,1]$ such that $f(x) = f^* = \inf_{y \in \mathbb{R}} f(y)$
- $f(x) \in [0,1]$ for all $x \in \mathbb{R}$
- $f$ is $L$-Lipschitz with respect to $\ell_\infty$
- Goal: compute $\epsilon$-optimal point for $\epsilon \in (0,1)$

- Question #1: what oracle complexity achievable?
- Question #0: what does $L$-Lipschitz mean? Imply?
Setting #1:

Algorithm
• Pick $k \in \mathbb{Z}_{\geq 0}$
• For $i \in [k] = \{1, \ldots, k\}$
  • Let $x_i = \frac{i}{k}$
  • Query $f(x_i)$ for all $i \in [k]$
• Return $x_{\text{out}} = \arg\min_{x_i} f(x_i)$

Theorem: there is method with query complexity $\lceil L/\epsilon \rceil$ for setting #1
Setting #1:

Algorithm

- Pick $k \in \mathbb{Z}_{\geq 0}$
- For $i \in [k] = \{1, \ldots, k\}$
  - Let $x_i = \frac{1}{k}$
  - Query $f(x_i)$ for all $i \in [k]$
- Return $x_{\text{out}} = \arg \min_{x_i} f(x_i)$

Theorem: there is method with query complexity $[L/\epsilon]$ for setting #1

Analysis

- $x_\star \in \left[\frac{i-1}{k}, \frac{i}{k}\right]$ for some $i \in [k]$
- $\exists i_\star \in [k]$ s.t. $\left|x_\star - \frac{i_\star}{k}\right| \leq \frac{1}{k}$
- $|f(x_{i_\star}) - f(x_\star)| \leq L \left\|x_\star - \frac{i_\star}{k}\right\|_\infty \leq \frac{L}{k}$
- $f(x_{i_\star}) \leq f_\star + \frac{L}{k}$
- $f(x_{\text{out}}) \leq f(x_{i_\star})$
- $k \geq L/\epsilon \Rightarrow f(x_{\text{out}})$ is $\epsilon$-optimal

Improvements? Lower bound?
Theorem: there is method with query complexity $1 + \lceil L/2\epsilon \rceil$ for setting #1

### Setting #1:

**Algorithm**
- Pick $k \in \mathbb{Z}_{\geq 0}$
- For $i \in \{0, 1, \ldots, k\}$
  - Let $x_i = \frac{i}{k}$
  - Query $f(x_i)$ for all $i \in [k]$
- Return $x_{\text{out}} = \arg\min f(x_i)$

**Analysis**
- $x_* \in \left[\frac{i-1}{k}, \frac{i}{k}\right]$ for some $i \in [k]$
- $\exists i_* \in \{0, \ldots, k\}$ s.t. $\left|x_* - \frac{i_*}{k}\right| \leq \frac{1}{2k}$
- $|f(x_{i_*}) - f(x_*)| \leq L \left\| x_* - \frac{i_*}{k} \right\|_\infty \leq \frac{1}{2k}$
- $f(x_{i_*}) \leq f_* + \frac{L}{2k}$
- $f(x_{\text{out}}) \leq f(x_{i_*})$
- $k \geq \frac{L}{2\epsilon} \Rightarrow f(x_{\text{out}})$ is $\epsilon$-optimal

Improvements? Lower bound?
Lower bound proof strategy

**Arbitrary Algorithm**
- For \( k = 1, \ldots, K \)
  - Compute point \( x_k \) based on previous oracle output (and randomness)
  - Query oracle at \( x_k \)
  - Output a point \( x_{\text{out}} \) based on previous points, oracle

**Lower Bound Strategy**
- From oracle output at \( x_1, \ldots, x_{k-1} \)
  - Specify oracle output at \( x_k \).
- Show that there are two valid functions \( f_1 \) and \( f_2 \) consistent with oracle output on \( x_1, \ldots, x_{k-1} \) with no common valid output point.

Any algorithm must take at least \( K \) steps.

Why?

Algorithm outputs incorrect answer on either \( f \) or \( g \).
Setting #0

Arbitrary Algorithm

- For $k = 1, \ldots, K$
  - Compute point $x_k$ based on previous oracle output (and randomness)
  - Query oracle at $x_k$
  - Output a point $x_{out}$ based on previous points, oracle

Any algorithm must take at least $K$ steps.

Candidate $f_i$

- For all $z \in [0,1]$ let
  \[
  f_z(x) = \begin{cases} 
  1 & x \neq z \\
  0 & x = z 
  \end{cases}
  \]

- Note: $f_{z_1}$ and $f_{z_2}$ have disjoint $\frac{1}{2}$-optimal points for $z_1 \neq z_2$

Lower Bound Strategy

- From oracle output at $x_1, \ldots, x_{K-1}$ specify oracle output at $x_k$.
  - Output $= 1$

- Show that there are two valid functions $f_1$ and $f_2$ consistent with oracle output on $x_1, \ldots, x_{K-1}$ with no common valid output point.

- $f_{z_1}$ and $f_{z_2}$ for any $z_1 \neq z_2$ with $z_1, z_2 \notin \{x_1, \ldots, x_k\}$

Since holds for all $K$, an infinite number of steps are needed.
Setting #1:

• $f : \mathbb{R} \to \mathbb{R}$ via evaluation oracle
• $\exists x_* \in [0,1]$ such that $f(x) = f_*$
• $f(x) \in [0,1]$ for all $x \in \mathbb{R}$
• $f$ is $L$-Lipschitz w.r.t $\| \cdot \|_\infty$
• Goal: compute $\epsilon$-optimal point

$f_{z,\alpha}(x) = \min\{1, -\alpha + L|x - z|\}$

Claims
• $x'$ is $\epsilon$-optimal for $f_{z,\alpha}$ for $\alpha > \epsilon$ if and only if $|x' - z| \leq L/\epsilon$
• $f_{z,\alpha}$ is $L$-Lipschitz w.r.t $\| \cdot \|_\infty$

Lower bound idea
• If oracle outputs 1 and not enough queries, consistent with two $f_{z,\alpha}$

Valid functions with disjoint $\epsilon$-optimal points.
Setting #1

- \( f : \mathbb{R} \rightarrow \mathbb{R} \) via evaluation oracle
- \( \exists x_* \in [0,1] \) such that \( f(x) = f_* \)
- \( f(x) \in [0,1] \) for all \( x \in \mathbb{R} \)
- \( f \) is \( L \)-Lipschitz w.r.t \( \| \cdot \|_\infty \)
- Goal: compute \( \epsilon \)-optimal point

\[ f_{z,\alpha}(x) = \min\{1, -\alpha + L|x - z|\} \]

Claims

- \( x' \) is \( \epsilon \)-optimal for \( f_{z,\alpha} \) for \( \alpha > \epsilon \) if and only if \( |x' - z| \leq L/\epsilon \)
- \( f_{z,\alpha} \) is \( L \)-Lipschitz w.r.t \( \| \cdot \|_\infty \)

Lower bound idea

- If oracle outputs 1 and not enough queries, consistent with two \( f_{z,\alpha} \)

Lower bound proof

- Algorithm makes \( K \)-queries
- Can partition \([0,1]\) with \( \leq K + 1 \) intervals so points are on boundary
- At least one interval is length at least \( 1/(k + 1) \)
- If length is \( > 4\epsilon/L \) then there are two \( f_{z,\alpha} \) consistent with disjoint \( \epsilon \)-optimal points
- \( \Rightarrow k + 1 > L/4\epsilon \)

Upper bound was \( \frac{L}{2\epsilon} + 1 \). Can we improve?

Lower Bound
At least \( \frac{L}{4\epsilon} - 2 \) queries are needed
Improve

- Algorithm also fails if there are two disjoint intervals of length \( > \frac{2\epsilon}{L} \)
- To succeed the total length of the intervals (1) satisfies
  \[ < k \left( \frac{2\epsilon}{L} \right) + \frac{4\epsilon}{L} \]
- \( k \geq \frac{L}{2\epsilon} - 2 \)
- Correct answer up to an additive 3!!!

Lower bound proof

- Algorithm makes \( K \)-queries
- Can partition \([0,1]\) with \( \leq K + 1 \) intervals so points are on boundary
- At least one interval is length at least \( 1/(k + 1) \)
- If length is \( > \frac{4\epsilon}{L} \) then there are two \( f_{z,\alpha} \) consistent with disjoint \( \epsilon \)-optimal points
- \( \Rightarrow k + 1 \geq L/4\epsilon \)
Setting #2: Higher Dimensions

Algorithm (\(\epsilon\)-net)

- Pick \(k \in \mathbb{Z}_{\geq 0}\)
- Query \(\left(\frac{i_1}{k}, \frac{i_2}{k}, \ldots, \frac{i_k}{k}\right)^T\) for all possible \(i_j \in [k]\)
- Return point of minimum value

Analysis

- \(\forall i \in [n], \exists j \in [k] \text{ s.t. } \left|x^*_i(i) - \frac{i}{k}\right| \leq \frac{1}{k}\)
- \(\exists q \text{ queried s.t. } ||x^* - q||_{\infty} \leq \frac{1}{k}\)
- \(f(q) \leq f(x^*_i) + \frac{L}{k}\)
- Point output is \(\frac{L}{k}\)-optimal
- \(k^n\) queries are made
- \(\left\lceil\frac{L}{\epsilon}\right\rceil^n\)-queries suffice

How do we avoid this large dependence on dimension?

Optimal up to constants! \(((cL/\epsilon)^n\) queries are needed)
Lecture Plan

Recap
- Oracles, minimization, efficiency, and iterative methods
- Continuity, smoothness, and critical points

Material
- Continuity, $\epsilon$-nets, and lower bounds

Thursday
- Lipschitzness and smoothness elaborated / revisited
- Convexity