Uniform Optimality for Convex and Nonconvex Optimization

Guanghui (George) Lan

School of Industrial and Systems Engineering
Georgia Institute of Technology
Joint work with Tianjiao Li, Yuyuan Ouyang and Zhe Zhang

Optimization Seminar
Statistics and Data Science
University of Pennsylvania
December 7, 2023

- Background
- Smooth convex optimization
    - Small function value
    - Small (projected) gradient
- Strongly convex problems
- Nonconvex problems
- Summary

Problem of Interest

Consider

$$f^* := \min_{x \in X} f(x)$$

- x: decision variable
- $X \subseteq \mathbb{R}^n$: feasible set
- f: objective function
- For simplicity, assume $X = \mathbb{R}^n$

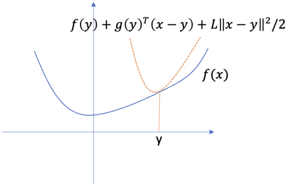
Smoothness of f: upper curvature

L-smooth (our focus)

f differentiable, for some $L > 0$:
$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + L\|x - y\|^2/2, \forall x, y.$

(α, L_α)-weakly smooth

f differentiable, for some $\alpha \in [0, 1)$ and $L_\alpha > 0$,
$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + L_\alpha\|x - y\|^{1+\alpha}/2, \forall x, y.$



Regularity of f: lower curvature

Convex

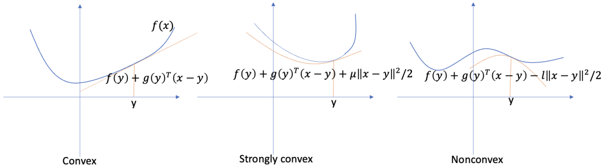
$f(x) - f(y) - \langle \nabla f(y), x - y \rangle \geq 0.$

μ-strongly convex

for some $\mu > 0$, $f(x) - f(y) - \langle \nabla f(y), x - y \rangle \geq \mu \|x - y\|^2/2$

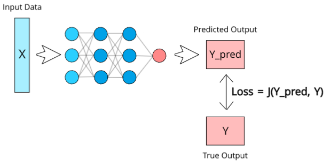
l-nonconvex

for some $l \in (0, L)$, $f(x) - f(y) - \langle \nabla f(y), x - y \rangle \geq -l \|x - y\|^2/2$



First-order methods

- Iterative methods working with $\nabla f(x)$ and $f(x)$ only
- Wide applications in machine learning and data science
  - Each iteration is cheap
  - No need for high accuracy
- Accuracy measure
  - $f(\hat{x}) - f^* \leq \varepsilon$ (for convex problems only)
  - $\|\nabla f(\hat{x})\| \leq \varepsilon$ (for both convex and nonconvex problems)
- Fundamental questions
  - How many gradient evaluations (gradient complexity)?
  - How much problem information?



Uniform Optimality

Definition (Lan 10, 11,13 (15))

First-order methods that can achieve the best possible gradient complexity without the input of any problem parameters.

- Problems parameters: L, α, L_α, μ, l, $\|x_0 - x^*\|$
- Defined over a global scope, hard to estimate
- Conservative estimation slows down the algorithm
- Gaps between theory and practice
  - Nonsmooth methods perform better than smooth ones
  - Non-accelerated methods run faster than accelerated ones
- A lot of tuning required for first-order methods

What has been done?

Focused on smooth convex optimization, and small function value

- Accelerated prox-level method (Lan 10, 11, 13))
  - Uniformly optimal for smooth, weakly smooth and nonsmooth problems
  - Extended for unbounded case (Chen et. al. 14)
  - Require projection over X plus one linear constraint
- Fast gradient method (Nesterov 13)
  - Uniformly (universally) optimal for smooth, weakly smooth and nonsmooth problems
  - Simple subproblem, can deal with unbounded sets
  - Require a line search procedure
  - Require the input of target accuracy

A forgotten paper (Chen, Lan, Ouyang, Zhang 14)

- Achieved the best complexity among parameter-free algorithms for unconstrained nonsmooth optimization
  - Fierce discussions in online learning and ML communities
- A Matlab implementation can beat Lapack for solving underdetermined linear systems!

TABLE 5.3
Comparison to Matlab solver

Matrix A:$m \times n$	Matlab $A\backslash b$		FAPL method		
	Time	Acc.	Iter.	Time	Acc.
Uniform 2000 × 4000	4.41	5.48e-24	204	3.59	6.76e-23
Uniform 2000 × 6000	7.12	9.04e-24	155	4.10	9.73e-23
Uniform 2000 × 8000	9.80	9.46e-24	135	4.45	9.36e-23
Uniform 2000 × 10000	12.43	1.04e-23	108	4.23	7.30e-23
Gaussian 3000 × 5000	11.17	5.59e-25	207	6.25	7.18e-23
Gaussian 3000 × 6000	13.96	1.43e-24	152	5.50	9.59e-23
Gaussian 3000 × 8000	19.57	1.66e-24	105	4.83	8.17e-23
Gaussian 3000 × 10000	25.18	1.35e-24	95	5.43	5.81e-23

Plan for this talk

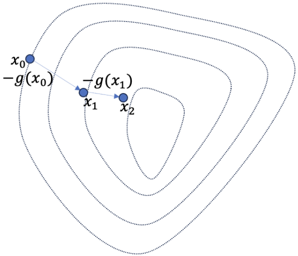
- Smooth convex optimization: Small function value
  - Novel method: Simple subproblem, line search free
- Smooth convex optimization: Small gradient
  - Novel method, parameter-free
- Strongly convex optimization
  - New complexity, parameter-free
- Nonconvex optimization
  - New complexity, parameter-free

Overview of results

	Termination	Algorithms	Complexity	Parameter free	Line search free	Easy subproblem
Convex	$f(x) - f^* \le \epsilon$	APL (Lan 10)	$\sqrt{LD^2/\epsilon}$	Yes	Yes	Not in general
Convex	$f(x) - f^* \le \epsilon$	FGM (Nesterov 13)	$\sqrt{LD^2/\epsilon}$	Yes	No	Yes
Convex	$f(x) - f^* \le \epsilon$	AC-FGM (Li and Lan 23)	$\sqrt{LD^2/\epsilon}$	Yes	Yes	Yes
Convex	$\|\nabla f(x)\| \le \epsilon$	AR (Lan, Ouyang and Zhang 23)	$\sqrt{LD/\epsilon}$	Yes	backtracking	Yes
Strongly convex	$\|\nabla f(x)\| \le \epsilon$	SCAR (Lan, Ouyang and Zhang 23)	$\sqrt{\frac{L}{\mu}} log\frac{\|\nabla f(x_0)\|}{\epsilon}$ (new)	Yes	backtracking	Yes
Nonconvex	$\|\nabla f(x)\| \le \epsilon$	NASCAR (Lan, Ouyang and Zhang 23)	$\frac{\sqrt{Ll}\,[f(x_0)-f^*]}{\epsilon^2}$ (new)	Yes	backtracking	Yes

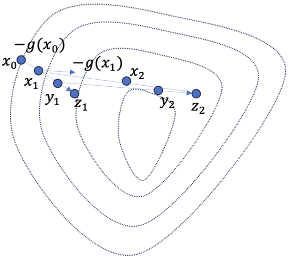
Gradient descent

- $x_t = \arg\min_{z \in X} \left\{ \eta_t \langle \nabla f(x_{t-1}), z \rangle + \frac{\|x_{t-1} - z\|^2}{2} \right\}$
- Minimize linear model at x_{t-1}, not moving far from x_{t-1}
- η_t depends on the (local) Lipschitz constant: $\frac{1}{\eta_t} \geq L_t := \frac{2[f(x_{t-1}) - f(x_t) - \langle g(x_t), x_{t-1} - x_t \rangle]}{\|g(x_t) - g(x_{t-1})\|_*^2}$
- But L_t is unknown when selecting η_t
  - require global estimate of L or line search
- Can η_t be determined based on $L_1, \ldots, L_{t-1}$?



Auto-conditioned Fast Gradient Method (AC-FGM)

- Trust linear model at x_{t-1}, not far from prox-center y_{t-1}:
  $z_t = \arg\min_{z \in X} \left\{ \eta_t \langle g(x_{t-1}), z \rangle + \frac{1}{2}\|y_{t-1} - z\|^2 \right\}$
- Update prox-center: $y_t = (1 - \beta_t) y_{t-1} + \beta_t z_t$
- Update output: $x_t = (z_t + \tau_t x_{t-1})/(1 + \tau_t)$



Difference from Accelerated Gradient Descent

- In contrast to Nesterov's AGD (84):
  $$y_t = (1-\alpha_t)x_{t-1} + \alpha_t z_{t-1},$$
  $$z_t = \arg\min_{z \in X} \left\{\eta_t \langle g(y_t), z\rangle + \tfrac{1}{2}\|z_{t-1} - z\|^2\right\},$$
  $$x_t = (1-\alpha_t)x_{t-1} + \alpha_t z_t.$$
- AGD uses $\{z_t\}$ as prox-centers, while AC-FGM uses the sequence $\{y_t\}$, a moving average of $\{z_t\}$ as prox-centers.
- AGD builds model at $\{y_t\}$ rather than the output solutions $\{x_t\}$, while AC-FGM computes model at $\{x_t\}$.
- Interpretation of AGD
  - Earlier AGD with nice geometric interpretation: Nemirovski and Yudin, 79(83a), 83b
  - Game interpretation: Lan and Zhou, 2015

Game interpretation of AC-FGM

- A buyer-seller game $\min_x \max_g \{\langle x, g\rangle - f^*(g)\}$:
  - Buyer: to determine order quantity to minimize cost
  - Seller: to determine price to maximize profit, f^* being the production cost
- Buyer determines order z_t, based on price g_{t-1}, but not too far away from y_{t-1} (i.e., a moving average of z_t).
  - $z_t = \arg\min_{z\in X} \left\{\eta_t\langle g_{t-1}, z\rangle + \frac{1}{2}\|y_{t-1} - z\|^2\right\}$
  - $y_t = (1-\beta_t)y_{t-1} + \beta_t z_t$
- Seller determines the prize g_t, based on the demand z_t, but not too far away from the previous price g_{t-1}.
  - $g_t = \arg\max_g \{\langle z_t, g\rangle - f^*(g) - \tau_t V(g_{t-1}, g)\}$
  - $V(g_{t-1}, g) := f^*(g) - [f^*(g_{t-1}) + \langle [f^*]'(g_{t-1}), g - g_{t-1}\rangle]$
  - Reduces to compute $\nabla f(x_t)$ at $x_t = (z_t + \tau_t x_{t-1})/(1+\tau_t)$

Convergence rate of AC-FGM

Theorem. Suppose $\tau_1 = 0, \ \tau_t = \frac{t}{2}$ for $t \geq 2$, $\beta \in (0, 1 - \frac{\sqrt{3}}{2}]$, and the stepsize η_t follows the rule:

$$\eta_t = \min\{\tfrac{t}{t-1}\eta_{t-1}, \tfrac{\beta(t-1)}{8L_{t-1}}\} \ t \geq 4,$$

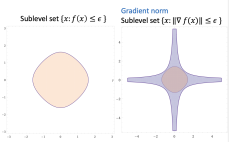
with η_t, $t \leq 3$, being properly specified. Then we have for $t \geq 2$,

$$f(x_k) - f(x^*) \leq \tfrac{\mathcal{O}(1)L}{k(k+1)} \|z_0 - x^*\|^2.$$

Note: (a) η_t only depends on $L_1, \ldots, L_{t-1}$, no need for line search; (b) Optimal rate of convergence.

Why should we care about gradients

- Previous studies focuses on termination criterion $f(\hat{x}) - f(x^*) \leq \varepsilon$
  - f^* unknown, difficult to check
- Easy to check whether $\|\nabla f(\hat{x})\| \leq \varepsilon$
- $\|\nabla f(\hat{x})\| \leq \varepsilon$ is a stronger criterion: by $\|\nabla f(\hat{x})\|^2/(2L) \leq f(\hat{x}) - f^* \leq \|\nabla f(\hat{x})\|\|\hat{x} - x^*\|$,
  - ε-gradient implies ε-function gap.
  - ε-function gap implies $\sqrt{\varepsilon}$-gradient.
- Turns out to be very important to design uniformly optimal algorithms for strongly convex and nonconvex problems



What is the status to drive gradients small?

- For a long period of time, only exist suboptimal methods
  - Worse than the lower gradient complexity bound $\mathcal{O}(\sqrt{L\|x_0 - x^*\|/\varepsilon})$ by a logarithmic factor.
- This lower bound is recently achieved by an optimized gradient method (Kim and Fessler 2021, Nesterov et. al. 2021, Diakonikolas et. al. 2022, Lee at. al. 2021).
  - Computer assisted algorithm design, empirically “solving” a nonconvex semidefinite programming
  - Combining two algorithms: the first one computes small function value and the second one drives gradient small
  - Lack intuitive interpretation
  - Require total number of iterations N given in advance. Do not actually use $\|\nabla f(\hat{x})\| \le \epsilon$ to terminate the algorithm
- No existence of parameter-free methods

A blackbox reduction to make gradient small

Algorithm Accumulative regularization for gradient minimization

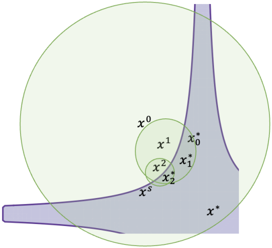
Input: strictly increasing $\{\sigma_s\}_{s=0}^S$ with $\sigma_0 = 0$; $\overline{x}_0 := x_0$.
for $s = 1, \dots, S$ **do**
Set $\overline{x}_s = (1 - \gamma_s)\overline{x}_{s-1} + \gamma_s x_{s-1}$ with $\gamma_s = 1 - \sigma_{s-1}/\sigma_s$.
Compute an approximate solution x_s of

$$x_s^* := \arg\min_{x \in \mathbb{R}^n} \{f_s(x) := f(x) + \frac{\sigma_s}{2}\|x - \overline{x}_s\|^2\}$$

by running an optimal algorithm $\mathcal{A}$ for smooth convex optimization (e.g., AC-FGM).
end for

Convergence for accumulative regularization

- Adaptive selection of prox-centers (again!) in the proximal point method.
- Sublinear convergence of $\mathcal{A}$, i.e., $f_s(x_s) - f_s(x_s^*) \leq \frac{c_{\mathcal{A}} \hat{L}}{k_s^2} \|x_{s-1} - x_s^*\|^2$ after k_s steps. Here $c_{\mathcal{A}}$ is a universal constant, and $\hat{L} \leq L$.



Theorem. If $\sigma_s = 4^{s-2}\varepsilon/D$ and $S = 1 + \lceil \log_4(LD/\varepsilon) \rceil$, where $D \geq \min_{x^* \in X^*} \|x_0 - x^*\|$, then the number of gradient evaluations to have $\|\nabla f(x_S)\| \leq \varepsilon$ is bounded by $\mathcal{O}(1)\sqrt{LD/\varepsilon}$.

Parameters L and D in accumulative regularization

- D can be handled by a doubling trick, but not needed for strongly convex and nonconvex problems.
- Subproblems are solved by a uniformly optimal method.
- Only need local Lipschitz constant of f at x_s

Algorithm M=Backtracking(h, σ, x, M_0)

for $j = 0, 1, \ldots,$ **do**

Set $x^{++} = x - (1/(2(M_j + \sigma)))\nabla g(x)$.

If $h(x^{++}) - h(x) - \langle \nabla h(x), x^{++} - x\rangle \le \frac{M_j+\sigma}{2}\|x^{++} - x\|^2$, then **terminate** with $M = M_j$.

Otherwise, set $M_{j+1} = 2M_j$.

end for

Parameter-free accumulative regularization

Algorithm Accumulative regularization (AR) without input of L

function $(\hat{x}, M) = \mathrm{AR}(f, x_0, \sigma_1, M_0)$
  for $s = 1, 2, \ldots$ **do**
    Compute an approximate solution x_s of the proximal subproblem by running $\mathcal{A}$ with initial point x_{s-1}.
    Set $M_s = \text{Backtracking}\ (f_s,\ \sigma_s,\ x_s,\ M_{s-1}/2)$.
    If $\sigma_s \geq M_s$, then **terminate** with $\hat{x} = x_s$ and $M = M_s$.
  end for
end function

Convergence: A similar gradient complexity bound as before, in addition to $\log_4(M/M_0)$ function evaluations in backtracking.

What is the current status?

- AGD finds $\|\nabla f(\hat{x})\| \leq \varepsilon$ within $\mathcal{O}(1)\sqrt{L/\mu}\log(L/(\mu\varepsilon))$ gradient evaluations.
- The strong convexity modulus μ defined over a global scope is notoriously hard to estimate.
- Can we improve the gradient complexity to an optimal one: $\mathcal{O}(1)\sqrt{L/\mu}\log(1/\varepsilon)$?
- Can we achieve such complexity without the input of μ?

Basic Ideas

- For any input argument $\sigma_1 > 0$, the AR method outputs a point $\hat{x}$ with $\|\nabla f(\hat{x})\| \leq 5\sigma_1\|x_0 - x^*\|$.
- Setting $\sigma_1 = \mu/10$ and using the strong convexity of f, we have $\|\nabla f(\hat{x})\| \leq \mu\|x_0 - x^*\|/2 \leq \|\nabla f(x_0)\|/2$.
- The gradient norm is now reduced by half and we may restart the AR method.
- This results in an $\mathcal{O}(1)\sqrt{L/\mu}\log(1/\varepsilon)$ optimal complexity.
- When μ is not available, set $\sigma_1 = \tilde{\mu}/10$ with a guess $\tilde{\mu}$.
- A guess-and-check implementation to search correct $\tilde{\mu}$ since $\|\nabla f(\hat{x})\|$ can be computed.

Parameter-free optimal algorithm

Algorithm Strongly convex accumulative regularization (SCAR)

function $(\hat{x}, \hat{M})$ = SCAR$(f, \varepsilon, y_0, \mu_0, M_0)$
  for $t = 1, 2, \ldots$ **do**
    Set $(y_t, M_t) = \text{AR}(f, y_{t-1}, \frac{\mu_{t-1}}{10}, M_{t-1})$.
    If $\|\nabla f(y_t)\| > \frac{\|\nabla f(y_{t-1})\|}{2}$ then $\mu_t = \frac{\mu_{t-1}}{4}$ and $y_t = y_{t-1}$.
    If $\|\nabla f(y_t)\| \le \varepsilon$, **terminate** with $\hat{x} = y_t$ and $\hat{M} = M_t$.
  end for
end function

Initial selection: $\mu_0 = M_0 = \|\nabla f(y_0) - \nabla f(z_0)\| / \|y_0 - z_0\|$
Complexity: $\mathcal{O}(1)\left\{\sqrt{L/\mu}\log(\|\nabla f(x_0)\|/\varepsilon) + \log(\mu_0/\mu)\sqrt{L/\mu}\right\}$

Current status in nonconvex optimization

Let l be lower curvature. Starting with $x^0 \in \mathbb{R}^n$, set

$$x^i = \arg\min_{x \in \mathbb{R}^n} \{F_i(x) := f(x) + l\|x - x^{i-1}\|^2\}.$$

- By optimality condition: $F_i(x_i) + \frac{l}{2}\|x_{i-1} - x_i\|^2 \le F_i(x_{i-1})$, implying $f(x^{i-1}) - f(x^i) \ge 3\|\nabla f(x^i)\|^2/(8l)$.
- Telescopic sum: $\min_{i=1,\ldots,N} \|\nabla f(x^i)\|^2 \le \frac{8l(f(x^0)-f^*)}{3N}$.
- But x^i can only be computed approximately (e.g., by AGD).
- Find $\|\nabla f(\hat{x})\| \le \varepsilon$ within $\mathcal{O}(1)\frac{\sqrt{Ll}(f(x^0)-f^*)}{\epsilon^2} \log \frac{L}{l\varepsilon}$ gradient evaluations.
- Can we improve further the gradient complexity?
- Can we achieve such complexity without the input of l?

Suppose l is given

- Apply SCAR to $\min F_i(x)$. If $\|\nabla F^{(i)}(x^i)\| \leq \frac{\varepsilon}{4}$ but $\|\nabla f(x^i)\| \geq \varepsilon$, then $\|\nabla f(x^i)\|^2 \leq 10l[f(x^{i-1}) - f(x^i)]$.
- To bound the total number of gradient evaluations, formulate an optimization problem (with $y_i = \|\nabla f(x^i)\|$):

$$\max_{y_1,\ldots,y_N \in \mathbb{R}} \left\{ \sum_{i=1}^{N} \log_2 \frac{y_i}{\varepsilon} : \sum_{i=1}^{N} y_i^2 \leq \Delta;\ y_i \geq \varepsilon,\ \forall i \right\}.$$

- Obtain the desired $\mathcal{O}(1)\frac{\sqrt{Ll}}{\epsilon^2}[f(x^0) - f(x^*)]$ gradient complexity, the best-known complexity that has not been achieved before.

What if l is unknown?

- F_i may be nonconvex if l is underesitmated
- We need to modify SCAR to handle plausible strong convexity modulus $\tilde{\mu}$ (SCAR-PM).
- Subroutine $\mathcal{A}$ in AR terminates when $k \geq 8\sqrt{2L_s^k/\sigma_s}$.

---

function $(\hat{x}, \hat{M}, \textsc{Err})$=SCAR-PM$(f, \varepsilon, y_0, \tilde{\mu}, M_0)$
  for $t = 1, 2, \ldots$ **do**
    Set $(y_t, M_t) = \text{AR}(f, y_{t-1}, \tilde{\mu}/10, M_{t-1})$.
    If $\|\nabla f(y_t)\| > \|\nabla f(y_{t-1})\|/2$, then **terminate** with $\hat{x} = y_0$, $\hat{M} = M_t$, and ERR=TRUE.
    If $\|\nabla f(y_t)\| \leq \varepsilon$, then **terminate** with $\hat{x} = y_t$, $\hat{M} = M_t$ and ERR=FALSE.
  end for
end function

Nonconvex acceleration through strongly convex accumulative regularization (NASCAR)

function $\hat{x} = \text{NASCAR}(x^0, \varepsilon, M_0)$
    Set $M_0 = \|\nabla f(x^0) - \nabla f(z^0)\| / \|x^0 - z^0\|$, $l_0 = \textit{Initialize}(M_0)$.
    for $i = 1, \ldots,$ **do**
        Set $F^{(i)}(x) := f(x) + l_{i-1}\|x - x^{i-1}\|^2$.
        (x^i, M_i, ERR_i) = SCAR-PM$(F^{(i)}, \varepsilon/4, x^{i-1}, l_{i-1}, M_{i-1})$.
        If $\|\nabla f(x^i)\| \le \varepsilon$, then **terminate** with $\hat{x} := x^i$.
        If ERR_i =TRUE or $\|\nabla f(x^i)\|^2 > 10 l_i(f(x^{i-1}) - f(x^i))$,
then replace l_i and x^i by $4l_i$ and x^{i-1}, respectively.
    end for
end function

Note $\|\nabla F^{(i)}(x^i)\| \le \varepsilon/4$ but $\|\nabla f(x^i)\|^2 > 10\tilde{l}(f(x^{i-1}) - f(x^i))$ implies our guess $l_i < l$.

Initial estimation of l_0

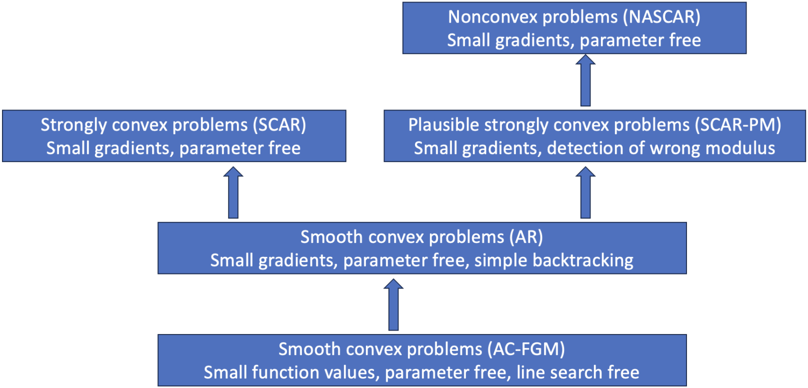
Algorithm Find an estimation of $l_0 \leq l$ or terminate NASCAR

function $\tilde{l} = \text{INITIALIZE}(\varepsilon, M_0)$
  Set $\tilde{l} = M_0$.
  for $i = 1, \ldots,$ **do**
    Set $F^{(0)}(x) := f(x) + \tilde{l}\|x - x^0\|^2$.
    $(\tilde{x}^0, M_1, \text{ERR})$ = SCAR-PM$(F_0^{(i)}, \varepsilon/4, x^0, \tilde{l}, M_0)$.
    If ERR=TRUE or $\|\nabla f(x^1)\|^2 > 10\tilde{l}(f(x^0) - f(x^1))$ then
  terminate with $\tilde{l}$.
    If $\|\nabla f(x^1)\| \leq \varepsilon$, then **terminate** NASCAR.
    Set $\tilde{l} = \tilde{l}/2$.
  end for
end function

Complexity of NASCAR

- Number of gradient evaluations in Initialization:
  $\mathcal{O}(1)\frac{\sqrt{L(f(x^0)-f^*)}}{\epsilon}\log\frac{M_0}{\epsilon}$.
- Number of gradient evaluations in main algorithm:
  $\mathcal{O}(1)\frac{\sqrt{Ll(f(x^0)-f(x^*))}}{\varepsilon^2}$.

Algorithm Tree



Summary

- AC-FGM: uniformly optimal without line search
  - An intuitive game interpretation
- AR: parameter-free optimal method to drive gradient small
  - Simple black-box reduction, no computer-aided design
- SCAR: parameter-free optimal method for strongly convex problems
  - New complexity bounds reported
- NASCAR: parameter-free method for nonconvex problems
  - New complexity bounds reported

References

- T. Li and G. Lan, A simple uniformly optimal method without line search for convex optimization, arXiv preprint arXiv:2310.10082, 10/2023.
- G. Lan, Y. Ouyang, and Z. Zhang, Optimal and parameter-free gradient minimization methods for convex and nonconvex optimization , arXiv preprint arXiv:2310.12139, 10/2023.