Testing Approximate Stationarity Concepts for Piecewise Affine (PA) Functions

Lai Tian

Department of Systems Engineering and Engineering Management

The Chinese University of Hong Kong (CUHK)

Joint Work with

Anthony Man-Cho So

January 13, 2025

ACM-SIAM Symposium on Discrete Algorithms (SODA), 2025

Travel support for this presentation was provided by IBM

Outline

Gradient Method for Nonconvex Functions

Gradient method for a β-smooth, nonconvex, lower bounded f:

$$\boldsymbol{x}_{t+1} := \boldsymbol{x}_t - \beta^{-1} \cdot \nabla f(\boldsymbol{x}_t).$$

Let $\Delta := f(\boldsymbol{x}_0) - \inf f < \infty$.

Gradient Method for Nonconvex Functions

Gradient method for a β-smooth, nonconvex, lower bounded f:

$$\boldsymbol{x}_{t+1} := \boldsymbol{x}_t - \beta^{-1} \cdot \nabla f(\boldsymbol{x}_t).$$

Let $\Delta := f(\boldsymbol{x}_0) - \inf f < \infty$. By descent lemma, we have

$$\min_{0 \le t \le T} \|\nabla f(\boldsymbol{x}_t)\| \le C \sqrt{\frac{\beta \Delta}{T+1}}.$$

Gradient Method for Nonconvex Functions

Gradient method for a β-smooth, nonconvex, lower bounded f:

$$\boldsymbol{x}_{t+1} := \boldsymbol{x}_t - \beta^{-1} \cdot \nabla f(\boldsymbol{x}_t).$$

Let $\Delta := f(\boldsymbol{x}_0) - \inf f < \infty$. By descent lemma, we have

$$\min_{0 \le t \le T} \|\nabla f(\boldsymbol{x}_t)\| \le C \sqrt{\frac{\beta \Delta}{T+1}}.$$

Two Observations.

- there exists an ε-stationary point $\boldsymbol{x}_t$, i.e., $\|\nabla f(\boldsymbol{x}_t)\| \le \varepsilon$, in $\{\boldsymbol{x}_t\}_{t=1}^T$, where T can be set as $O(\beta\Delta/\varepsilon^2)$ a priori.

Gradient Method for Nonconvex Functions

Gradient method for a β-smooth, nonconvex, lower bounded f:

$$\boldsymbol{x}_{t+1} := \boldsymbol{x}_t - \beta^{-1} \cdot \nabla f(\boldsymbol{x}_t).$$

Let $\Delta := f(\boldsymbol{x}_0) - \inf f < \infty$. By descent lemma, we have

$$\min_{0 \le t \le T} \|\nabla f(\boldsymbol{x}_t)\| \le C \sqrt{\frac{\beta \Delta}{T+1}}.$$

Two Observations.

- there exists an ε-stationary point $\boldsymbol{x}_t$, i.e., $\|\nabla f(\boldsymbol{x}_t)\| \le \varepsilon$, in $\{\boldsymbol{x}_t\}_{t=1}^T$, where T can be set as $O(\beta\Delta/\varepsilon^2)$ a priori.
- such a point $\boldsymbol{x}_t \in \{\boldsymbol{x}_t\}_{t=1}^T$ can be identified efficiently by evaluating and comparing $\{\|\nabla f(\boldsymbol{x}_t)\|\}_{t=1}^T$.

Nonconvex Nonsmooth Functions

Pervasive in Modern Data Science.

- modern neural networks:
  - (leaky) ReLU, max pooling, hinge loss, GANs, etc.
- max-affine regression, robust SVM, etc.

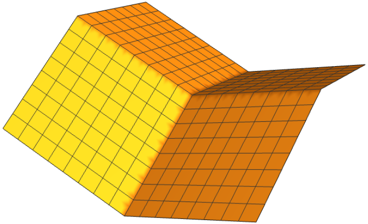


Figure: A nonconvex, nonsmooth, PA function.

Nonconvex Nonsmooth Functions

Pervasive in Modern Data Science.

- modern neural networks:
  - (leaky) ReLU, max pooling, hinge loss, GANs, etc.
- max-affine regression, robust SVM, etc.

An Immediate Question.

- what is the notion of "approximate stationarity" mimicking

$$\|\nabla f(\boldsymbol{x}_t)\| \leq \varepsilon,$$

and how to compute it?

Nonconvex Nonsmooth Functions (cont'd)

Two Generalized Notions.

- <u>differentiation</u>
  - replace ∇f with generalized (Clarke) subdifferential ∂f:
  $$\partial f(\boldsymbol{x}) := \mathrm{conv}\left\{\boldsymbol{g} : \exists \boldsymbol{x}_n \to \boldsymbol{x}, \nabla f(\boldsymbol{x}_n) \text{ exists}, \nabla f(\boldsymbol{x}_n) \to \boldsymbol{g}\right\}.$$
  - $\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$ if f is C^1; convex subdiff. if f is convex.

Nonconvex Nonsmooth Functions (cont'd)

Two Generalized Notions.

- <u>differentiation</u>
  - replace ∇f with generalized (Clarke) subdifferential ∂f:
    $$\partial f(\boldsymbol{x}) := \operatorname{conv}\left\{\boldsymbol{g} : \exists \boldsymbol{x}_n \to \boldsymbol{x}, \nabla f(\boldsymbol{x}_n) \text{ exists}, \nabla f(\boldsymbol{x}_n) \to \boldsymbol{g}\right\}.$$
  - $\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$ if f is C^1; convex subdiff. if f is convex.
- <u>approximation</u>
  - replace $\|\cdot\| \leq \varepsilon$ with something computable.

Nonconvex Nonsmooth Functions (cont'd)

Two Generalized Notions.

- <u>differentiation</u>
  - replace ∇f with generalized (Clarke) subdifferential ∂f:
    $$\partial f(\boldsymbol{x}) := \operatorname{conv}\left\{\boldsymbol{g} : \exists \boldsymbol{x}_n \to \boldsymbol{x}, \nabla f(\boldsymbol{x}_n) \text{ exists}, \nabla f(\boldsymbol{x}_n) \to \boldsymbol{g}\right\}.$$
  - $\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$ if f is C^1; convex subdiff. if f is convex.
- <u>approximation</u>
  - replace $\|\cdot\| \leq \varepsilon$ with something computable.
  - **a trivial idea:** we say $\boldsymbol{x}$ is ε-stationary if $\mathbf{0} \in \partial f(\boldsymbol{x}) + \varepsilon\mathbb{B}$,
    $$\text{or equivalently,} \qquad \operatorname{dist}(\mathbf{0}, \partial f(\boldsymbol{x})) \leq \varepsilon.$$

Nonconvex Nonsmooth Functions (cont'd)

Two Generalized Notions.

- <u>differentiation</u>
  - replace ∇f with generalized (Clarke) subdifferential ∂f:

$$\partial f(\boldsymbol{x}) := \operatorname{conv}\left\{\boldsymbol{g} : \exists \boldsymbol{x}_n \to \boldsymbol{x}, \nabla f(\boldsymbol{x}_n) \text{ exists}, \nabla f(\boldsymbol{x}_n) \to \boldsymbol{g}\right\}.$$

  - $\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$ if f is C^1; convex subdiff. if f is convex.
- <u>approximation</u>
  - replace $\|\cdot\| \leq \varepsilon$ with something computable.
  - **a trivial idea:** we say $\boldsymbol{x}$ is ε-stationary if $\mathbf{0} \in \partial f(\boldsymbol{x}) + \varepsilon\mathbb{B}$,

$$\text{or equivalently,} \qquad \operatorname{dist}(\mathbf{0}, \partial f(\boldsymbol{x})) \leq \varepsilon.$$

An ε-Stationary Point is Uncomputable.

- if $f(x) = |x - \pi|$, then $\operatorname{dist}(0, \partial f(\mathbb{Q})) \geq 1$.

Nonconvex Nonsmooth Functions (cont'd)

Two Generalized Notions.

- <u>differentiation</u>
  - replace ∇f with generalized (Clarke) subdifferential ∂f:
  $$\partial f(\boldsymbol{x}) := \mathrm{conv}\left\{\boldsymbol{g} : \exists \boldsymbol{x}_n \to \boldsymbol{x}, \nabla f(\boldsymbol{x}_n) \text{ exists}, \nabla f(\boldsymbol{x}_n) \to \boldsymbol{g}\right\}.$$
  - $\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$ if f is C^1; convex subdiff. if f is convex.
- <u>approximation</u>
  - replace $\|\cdot\| \leq \varepsilon$ with something computable.
  - **a trivial idea:** we say $\boldsymbol{x}$ is ε-stationary if $\mathbf{0} \in \partial f(\boldsymbol{x}) + \varepsilon\mathbb{B}$,
  $$\text{or equivalently,} \qquad \mathrm{dist}(\mathbf{0}, \partial f(\boldsymbol{x})) \leq \varepsilon.$$

An ε-Stationary Point is Uncomputable.

- if $f(x) = |x - \pi|$, then $\mathrm{dist}(0, \partial f(\mathbb{Q})) \geq 1$.
- we need a computable notion of approximation.

Near-Approximate Stationarity (NAS)

Definition (near-approximate stationarity; NAS)

We say $\boldsymbol{x}$ is an (ε, δ)-NAS point of f, if

$$\mathbf{0} \in \partial f(\boldsymbol{x} + \delta \mathbb{B}) + \varepsilon \mathbb{B}.$$

- recall $\partial f(\boldsymbol{x}) = \bigcap_{\delta > 0} \partial f(\boldsymbol{x} + \delta \mathbb{B})$.
- also, $(\varepsilon, 0)$-NAS is ε-stationarity.

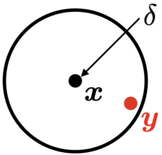
Near-Approximate Stationarity (NAS)

Definition (near-approximate stationarity; NAS)

We say $\boldsymbol{x}$ is an (ε, δ)-NAS point of f, if

$$\mathbf{0} \in \partial f(\boldsymbol{x} + \delta\mathbb{B}) + \varepsilon\mathbb{B}.$$

- recall $\partial f(\boldsymbol{x}) = \bigcap_{\delta>0} \partial f(\boldsymbol{x} + \delta\mathbb{B})$.
- also, $(\varepsilon, 0)$-NAS is ε-stationarity.



$$\mathbf{0} \in \partial f(\boldsymbol{y}) + \varepsilon\mathbb{B}$$

Subgradient Method

- let a lower-bounded semialgebraic (e.g., PA) f be given.

- Stochastic subgradient method converges on tame functions, FoCM '20.
- No dimension-free deterministic algorithm computes approximate stationarities of Lipschitzians, MP '24.

Subgradient Method

- let a lower-bounded semialgebraic (e.g., PA) f be given.
- consider a subgradient-type method:

$$\boldsymbol{x}_{n+1} := \boldsymbol{x}_n - \alpha_n \cdot \boldsymbol{g}_n \quad \text{with} \quad \boldsymbol{g}_n \in \partial f(\boldsymbol{x}_n).$$

- Stochastic subgradient method converges on tame functions, FoCM '20.
- No dimension-free deterministic algorithm computes approximate stationarities of Lipschitzians, MP '24.

Subgradient Method

- let a lower-bounded semialgebraic (e.g., PA) f be given.
- consider a subgradient-type method:

$$\boldsymbol{x}_{n+1} := \boldsymbol{x}_n - \alpha_n \cdot \boldsymbol{g}_n \quad \text{with} \quad \boldsymbol{g}_n \in \partial f(\boldsymbol{x}_n).$$

- (Davis et al. '20) shows that if $\boldsymbol{x}_n \to \boldsymbol{x}^*$, then $\mathbf{0} \in \partial f(\boldsymbol{x}^*)$.

- Stochastic subgradient method converges on tame functions, FoCM '20.
- No dimension-free deterministic algorithm computes approximate stationarities of Lipschitzians, MP '24.

Subgradient Method

- let a lower-bounded semialgebraic (e.g., PA) f be given.
- consider a subgradient-type method:

$$\boldsymbol{x}_{n+1} := \boldsymbol{x}_n - \alpha_n \cdot \boldsymbol{g}_n \quad \text{with} \quad \boldsymbol{g}_n \in \partial f(\boldsymbol{x}_n).$$

- (Davis et al. '20) shows that if $\boldsymbol{x}_n \to \boldsymbol{x}^*$, then $\mathbf{0} \in \partial f(\boldsymbol{x}^*)$.
- in other words, for any $\varepsilon \geq 0$ and $\delta > 0$, there exists a finite N such that $\boldsymbol{x}_N$ is (ε, δ)-NAS, or equivalently,

$$\mathbf{0} \in \partial f(\boldsymbol{x}_N + \delta\mathbb{B}) + \varepsilon\mathbb{B}.$$

- Stochastic subgradient method converges on tame functions, FoCM '20.
- No dimension-free deterministic algorithm computes approximate stationarities of Lipschitzians, MP '24.

Subgradient Method

- let a lower-bounded semialgebraic (e.g., PA) f be given.
- consider a subgradient-type method:

$$\boldsymbol{x}_{n+1} := \boldsymbol{x}_n - \alpha_n \cdot \boldsymbol{g}_n \quad \text{with} \quad \boldsymbol{g}_n \in \partial f(\boldsymbol{x}_n).$$

- (Davis et al. '20) shows that if $\boldsymbol{x}_n \to \boldsymbol{x}^*$, then $\mathbf{0} \in \partial f(\boldsymbol{x}^*)$.
- in other words, for any $\varepsilon \geq 0$ and $\delta > 0$, there exists a finite N such that $\boldsymbol{x}_N$ is (ε, δ)-NAS, or equivalently,

$$\mathbf{0} \in \partial f(\boldsymbol{x}_N + \delta\mathbb{B}) + \varepsilon\mathbb{B}.$$

- unlike the smooth case, (T.-So, MP '24) shows that a priori estimation of N is impossible, even when f is PA.

- Stochastic subgradient method converges on tame functions, FoCM '20.
- No dimension-free deterministic algorithm computes approximate stationarities of Lipschitzians, MP '24.

Subgradient Method

- let a lower-bounded semialgebraic (e.g., PA) f be given.
- consider a subgradient-type method:

$$\boldsymbol{x}_{n+1} := \boldsymbol{x}_n - \alpha_n \cdot \boldsymbol{g}_n \quad \text{with} \quad \boldsymbol{g}_n \in \partial f(\boldsymbol{x}_n).$$

- (Davis et al. '20) shows that if $\boldsymbol{x}_n \to \boldsymbol{x}^*$, then $\mathbf{0} \in \partial f(\boldsymbol{x}^*)$.
- in other words, for any $\varepsilon \geq 0$ and $\delta > 0$, there exists a finite N such that $\boldsymbol{x}_N$ is (ε, δ)-NAS, or equivalently,

$$\mathbf{0} \in \partial f(\boldsymbol{x}_N + \delta\mathbb{B}) + \varepsilon\mathbb{B}.$$

- unlike the smooth case, (T.-So, MP '24) shows that a priori estimation of N is impossible, even when f is PA.
- **Question.** how can we identify such an $\boldsymbol{x}_N$ from $\{\boldsymbol{x}_n\}_n$?

- Stochastic subgradient method converges on tame functions, FoCM '20.
- No dimension-free deterministic algorithm computes approximate stationarities of Lipschitzians, MP '24.

Testing NAS: What and Why?

Given $f : \mathbb{R}^d \to \mathbb{R}$, $\boldsymbol{x} \in \mathbb{R}^d$, and $\varepsilon, \delta \geq 0$, decide whether

$$\mathbf{0} \in \partial f(\boldsymbol{x} + \delta \mathbb{B}) + \varepsilon \mathbb{B}.$$

Testing NAS: What and Why?

Given $f : \mathbb{R}^d \to \mathbb{R}$, $\boldsymbol{x} \in \mathbb{R}^d$, and $\varepsilon, \delta \geq 0$, decide whether

$$\mathbf{0} \in \partial f(\boldsymbol{x} + \delta\mathbb{B}) + \varepsilon\mathbb{B}.$$

Why is Testing Interesting?

- why not? a “dual” task to finding stationary points.

Testing NAS: What and Why?

Given $f : \mathbb{R}^d \to \mathbb{R}$, $\boldsymbol{x} \in \mathbb{R}^d$, and $\varepsilon, \delta \geq 0$, decide whether

$$\mathbf{0} \in \partial f(\boldsymbol{x} + \delta\mathbb{B}) + \varepsilon\mathbb{B}.$$

Why is Testing Interesting?

- why not? a "dual" task to finding stationary points.
- a universal stopping rule pertains to finite termination.

Testing NAS: What and Why?

Given $f : \mathbb{R}^d \to \mathbb{R}$, $\boldsymbol{x} \in \mathbb{R}^d$, and $\varepsilon, \delta \geq 0$, decide whether

$$\mathbf{0} \in \partial f(\boldsymbol{x} + \delta\mathbb{B}) + \varepsilon\mathbb{B}.$$

Why is Testing Interesting?

- why not? a "dual" task to finding stationary points.
- a universal stopping rule pertains to finite termination.
  - quoting L. Vandenberghe for subgradient method in ECE236C:

(even for convex nonsmooth f) "no good stopping criterion."

Testing NAS: What and Why?

Given $f : \mathbb{R}^d \to \mathbb{R}$, $\boldsymbol{x} \in \mathbb{R}^d$, and $\varepsilon, \delta \geq 0$, decide whether

$$\mathbf{0} \in \partial f(\boldsymbol{x} + \delta\mathbb{B}) + \varepsilon\mathbb{B}.$$

Why is Testing Interesting?

- why not? a "dual" task to finding stationary points.
- a universal stopping rule pertains to finite termination.
  - quoting L. Vandenberghe for subgradient method in ECE236C:

    (even for convex nonsmooth f) "no good stopping criterion."
- if f is nonconvex nonsmooth (e.g., PA), when to stop?

Piecewise Affine (PA) Functions

- On the expressibility of piecewise-linear continuous functions as the difference of two piecewise-linear convex functions, MP '86

Piecewise Affine (PA) Functions

Theorem (Melzer '86)

Any PA function $f : \mathbb{R}^d \to \mathbb{R}$ can be written as the difference of two convex PA functions $h, g : \mathbb{R}^d \to \mathbb{R}$, i.e., $f = h - g$.

Remarks.

- analytic approximation of piecewise smooth functions.

• On the expressibility of piecewise-linear continuous functions as the difference of two piecewise-linear convex functions, MP '86

Piecewise Affine (PA) Functions

Theorem (Melzer '86)

Any PA function $f : \mathbb{R}^d \to \mathbb{R}$ can be written as the difference of two convex PA functions $h, g : \mathbb{R}^d \to \mathbb{R}$, i.e., $f = h - g$.

Remarks.

- analytic approximation of piecewise smooth functions.
- we will consider h, g (locally) as support functions of projection of $\mathcal{H}$-polytopes.

• On the expressibility of piecewise-linear continuous functions as the difference of two piecewise-linear convex functions, MP '86

Outline

Complexity of Testing Solution Concepts

- local minimality:
  - testing local optimality for constrained QPs and degree-4 polynomials are both co-NP-hard (Murty-Kabadi '87).

- Some NP-complete problems in quadratic and nonlinear programming, MP '87.
- Gradient methods for minimizing composite functions, MP '13.

Complexity of Testing Solution Concepts

- local minimality:
  - testing local optimality for constrained QPs and degree-4 polynomials are both co-NP-hard (Murty-Kabadi '87).
  - testing local optimality for a PA functions is weakly co-NP-hard (Nesterov '13).

- Some NP-complete problems in quadratic and nonlinear programming, MP '87.
- Gradient methods for minimizing composite functions, MP '13.

Complexity of Testing Solution Concepts

- local minimality:
  - testing local optimality for constrained QPs and degree-4 polynomials are both co-NP-hard (Murty-Kabadi '87).
  - testing local optimality for a PA functions is weakly co-NP-hard (Nesterov '13).
- stationarity:
  - testing ε-stationarity ($\|\nabla f(\cdot)\| \leq \varepsilon$) for polynomials is in P.

- Some NP-complete problems in quadratic and nonlinear programming, MP '87.
- Gradient methods for minimizing composite functions, MP '13.

Complexity of Testing Solution Concepts

- local minimality:
  - testing local optimality for constrained QPs and degree-4 polynomials are both co-NP-hard (Murty-Kabadi '87).
  - testing local optimality for a PA functions is weakly co-NP-hard (Nesterov '13).
- stationarity:
  - testing ε-stationarity ($\|\nabla f(\cdot)\| \leq \varepsilon$) for polynomials is in P.
- no work on testing stationarity for general PA functions.

- Some NP-complete problems in quadratic and nonlinear programming, MP '87.
- Gradient methods for minimizing composite functions, MP '13.

Main Result I: Computational Complexity

Theorem (T.-So '25)

Fix any $\varepsilon \in [0, 1/2)$. Let two convex PA *functions $h, g : \mathbb{R}^d \to \mathbb{R}$ and a point $\boldsymbol{x} \in \mathbb{R}^d$ be given. For $h - g$, the following hold:*

- *Testing the local minimality of $\mathbf{0}$ is strongly co-*NP*-hard.*
- *Testing $\mathbf{0} \in \partial(h - g)(\boldsymbol{x}) + \varepsilon\mathbb{B}$ is strongly* NP*-hard.*

Remarks.

- cp. (Nesterov '13): weak co-NP-hardness; reduction from 2-PARTITION (pseudo-polynomial time solvable).

• Computational complexity of norm-maximization, Combinatorica '90.

Main Result I: Computational Complexity

Theorem (T.-So '25)

Fix any $\varepsilon \in [0, 1/2)$. Let two convex PA *functions $h, g : \mathbb{R}^d \to \mathbb{R}$ and a point $\boldsymbol{x} \in \mathbb{R}^d$ be given. For $h - g$, the following hold:*

- *Testing the local minimality of $\mathbf{0}$ is strongly co-*NP*-hard.*
- *Testing $\mathbf{0} \in \partial(h - g)(\boldsymbol{x}) + \varepsilon\mathbb{B}$ is strongly* NP*-hard.*

Remarks.

- cp. (Nesterov '13): weak co-NP-hardness; reduction from 2-PARTITION (pseudo-polynomial time solvable).
- to our knowledge, first hardness result for testing (approximate) stationarity.

• Computational complexity of norm-maximization, Combinatorica '90.

Main Result I: Computational Complexity

Theorem (T.-So '25)

Fix any $\varepsilon \in [0, 1/2)$. Let two convex PA *functions $h, g : \mathbb{R}^d \to \mathbb{R}$ and a point $\boldsymbol{x} \in \mathbb{R}^d$ be given. For $h - g$, the following hold:*

- *Testing the local minimality of $\mathbf{0}$ is strongly co-*NP*-hard.*
- *Testing $\mathbf{0} \in \partial(h - g)(\boldsymbol{x}) + \varepsilon\mathbb{B}$ is strongly* NP*-hard.*

Remarks.

- cp. (Nesterov '13): weak co-NP-hardness; reduction from 2-PARTITION (pseudo-polynomial time solvable).
- to our knowledge, first hardness result for testing (approximate) stationarity.
- reduction from the problem of maximizing ℓ_1-norm over a centered parallelotope (Bodlaender et al. '90).

• Computational complexity of norm-maximization, Combinatorica '90.

Outline

Sum Rule Relaxation

Sum Rule Relaxation (SRR). Given convex PA functions h, g and $\boldsymbol{x}$, we check the "ε-stationarity" of $h - g$ by running:

- find the shortest vector $\boldsymbol{g}$ in $\partial h(\boldsymbol{x}) - \partial g(\boldsymbol{x})$.
- if $\|\boldsymbol{g}\| \leq \varepsilon$: return `True`; else return `False`.

Sum Rule Relaxation

Sum Rule Relaxation (SRR). Given convex PA functions h, g and $\boldsymbol{x}$, we check the "ε-stationarity" of $h - g$ by running:

- find the shortest vector $\boldsymbol{g}$ in $\partial h(\boldsymbol{x}) - \partial g(\boldsymbol{x})$.
- if $\|\boldsymbol{g}\| \le \varepsilon$: return `True`; else return `False`.

Remarks.

- $\partial h(\boldsymbol{x})$ and $\partial g(\boldsymbol{x})$ are polytopes.

Sum Rule Relaxation

Sum Rule Relaxation (SRR). Given convex PA functions h, g and $\boldsymbol{x}$, we check the "ε-stationarity" of $h - g$ by running:

- find the shortest vector $\boldsymbol{g}$ in $\partial h(\boldsymbol{x}) - \partial g(\boldsymbol{x})$.
- if $\|\boldsymbol{g}\| \leq \varepsilon$: return `True`; else return `False`.

Remarks.

- $\partial h(\boldsymbol{x})$ and $\partial g(\boldsymbol{x})$ are polytopes.
- abusing convex subdifferential sum rule $\partial(h+g) = \partial h + \partial g$.

Sum Rule Relaxation

Sum Rule Relaxation (SRR). Given convex PA functions h, g and $\boldsymbol{x}$, we check the "ε-stationarity" of $h - g$ by running:

- find the shortest vector $\boldsymbol{g}$ in $\partial h(\boldsymbol{x}) - \partial g(\boldsymbol{x})$.
- if $\|\boldsymbol{g}\| \leq \varepsilon$: return `True`; else return `False`.

Remarks.

- $\partial h(\boldsymbol{x})$ and $\partial g(\boldsymbol{x})$ are polytopes.
- abusing convex subdifferential sum rule $\partial(h + g) = \partial h + \partial g$.
- efficiently computable (a convex QP).

Sum Rule Relaxation

Sum Rule Relaxation (SRR). Given convex PA functions h, g and $\boldsymbol{x}$, we check the "ε-stationarity" of $h - g$ by running:

- find the shortest vector $\boldsymbol{g}$ in $\partial h(\boldsymbol{x}) - \partial g(\boldsymbol{x})$.
- if $\|\boldsymbol{g}\| \leq \varepsilon$: return `True`; else return `False`.

Remarks.

- $\partial h(\boldsymbol{x})$ and $\partial g(\boldsymbol{x})$ are polytopes.
- abusing convex subdifferential sum rule $\partial(h + g) = \partial h + \partial g$.
- efficiently computable (a convex QP).
- sacrifice **correctness** for **efficiency** (why?).

Correctness and Sum Rule

- for smooth functions, we have $\nabla(h - g) = \nabla h - \nabla g$.

Correctness and Sum Rule

- for smooth functions, we have $\nabla(h-g) = \nabla h - \nabla g$.
- for Lipschitz continuous h, g, we only have $\partial(h-g)(\boldsymbol{x}) \subseteq \partial h(\boldsymbol{x}) - \partial g(\boldsymbol{x})$.
  - $\mathbf{0} \in \partial h(\boldsymbol{x}) - \partial g(\boldsymbol{x}) \;\not\Longrightarrow\; \mathbf{0} \in \partial(h-g)(\boldsymbol{x})$

Correctness and Sum Rule

- for smooth functions, we have $\nabla(h-g) = \nabla h - \nabla g$.
- for Lipschitz continuous h, g, we only have $\partial(h-g)(\boldsymbol{x}) \subseteq \partial h(\boldsymbol{x}) - \partial g(\boldsymbol{x})$.
  - $\mathbf{0} \in \partial h(\boldsymbol{x}) - \partial g(\boldsymbol{x}) \not\Longrightarrow \mathbf{0} \in \partial(h-g)(\boldsymbol{x})$
- in general, exact sum rule does not hold.
  - e.g., $\{0\} = \partial(|\cdot| - |\cdot|)(0) \subsetneq \partial|\cdot|(0) - \partial|\cdot|(0) = [-2, 2]$

Correctness and Sum Rule

- for smooth functions, we have $\nabla(h-g) = \nabla h - \nabla g$.
- for Lipschitz continuous h, g, we only have $\partial(h-g)(\boldsymbol{x}) \subseteq \partial h(\boldsymbol{x}) - \partial g(\boldsymbol{x})$.
  - $\mathbf{0} \in \partial h(\boldsymbol{x}) - \partial g(\boldsymbol{x}) \ \not\Longrightarrow\ \mathbf{0} \in \partial(h-g)(\boldsymbol{x})$
- in general, exact sum rule does not hold.
  - e.g., $\{0\} = \partial(|\cdot| - |\cdot|)(0) \subsetneq \partial|\cdot|(0) - \partial|\cdot|(0) = [-2, 2]$
- **goal:** isolate functions that enjoy the best of both worlds.
  - efficiency without sacrificing correctness

A New Geometric Notion

Definition (T.-So '25)

Two polytopes A and B are called compatible if for any vectors $\boldsymbol{a} \in A$ and $\boldsymbol{b} \in B$ such that $\boldsymbol{a} - \boldsymbol{b} \in \text{ext}(A - B)$, it holds

$$\boldsymbol{a} + \boldsymbol{b} \in \text{ext}(A + B).$$

A New Geometric Notion

Definition (T.-So '25)

Two polytopes A and B are called compatible if for any vectors $\boldsymbol{a} \in A$ and $\boldsymbol{b} \in B$ such that $\boldsymbol{a} - \boldsymbol{b} \in \operatorname{ext}(A - B)$, it holds

$$\boldsymbol{a} + \boldsymbol{b} \in \operatorname{ext}(A + B).$$

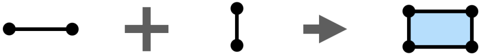


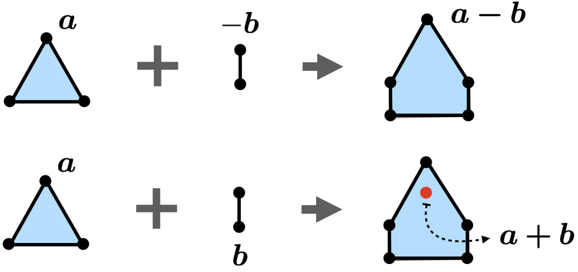
Figure: Two Compatible Polytopes in $\mathbb{R}^2$.

A New Geometric Notion

Definition (T.-So '25)

Two polytopes A and B are called compatible if for any vectors $\boldsymbol{a} \in A$ and $\boldsymbol{b} \in B$ such that $\boldsymbol{a} - \boldsymbol{b} \in \mathrm{ext}(A - B)$, it holds

$$\boldsymbol{a} + \boldsymbol{b} \in \mathrm{ext}(A + B).$$



Main Result II: Sum Rule

Theorem (T.-So '25)

Let any convex PA *functions* $h, g : \mathbb{R}^d \to \mathbb{R}$ *and a point* $\boldsymbol{x} \in \mathbb{R}^d$ *be given. The following are equivalent.*

1. $\partial(h - g)(\boldsymbol{x}) = \partial h(\boldsymbol{x}) - \partial g(\boldsymbol{x})$.
2. $\partial h(\boldsymbol{x})$ *and* $\partial g(\boldsymbol{x})$ *are compatible polytopes.*

Remarks on Compatibility.

- efficiently verifiable if $\partial h(\boldsymbol{x})$ and $\partial g(\boldsymbol{x})$ are $\mathcal{V}$-polytopes.

Main Result II: Sum Rule

Theorem (T.-So '25)

Let any convex PA *functions* $h, g : \mathbb{R}^d \to \mathbb{R}$ *and a point* $\boldsymbol{x} \in \mathbb{R}^d$ *be given. The following are equivalent.*

1. $\partial(h - g)(\boldsymbol{x}) = \partial h(\boldsymbol{x}) - \partial g(\boldsymbol{x})$.
2. $\partial h(\boldsymbol{x})$ *and* $\partial g(\boldsymbol{x})$ *are compatible polytopes.*

Remarks on Compatibility.

- efficiently verifiable if $\partial h(\boldsymbol{x})$ and $\partial g(\boldsymbol{x})$ are $\mathcal{V}$-polytopes.
- in general, verification may require vertex enumeration.

Transversality

To Improve Computability:

Defintion (T.-So '25)

Given two convex PA functions $h, g : \mathbb{R}^d \to \mathbb{R}$, we say functions h and g are transversal at a point $\boldsymbol{x} \in \mathbb{R}^d$ if
$$\mathrm{par}(\partial h(\boldsymbol{x})) \cap \mathrm{par}(\partial g(\boldsymbol{x})) = \{\mathbf{0}\}.$$

Remarks.

- recall $\mathrm{par}(C) = \mathrm{aff}(C - C)$.

Transversality

To Improve Computability:

Defintion (T.-So '25)

Given two convex PA functions $h, g : \mathbb{R}^d \to \mathbb{R}$, we say functions h and g are transversal at a point $\boldsymbol{x} \in \mathbb{R}^d$ if

$$\mathrm{par}(\partial h(\boldsymbol{x})) \cap \mathrm{par}(\partial g(\boldsymbol{x})) = \{\mathbf{0}\}.$$

Remarks.

- recall $\mathrm{par}(C) = \mathrm{aff}(C - C)$.
- polynomial-time checkable for $\mathcal{V}$- ,$\mathcal{H}$-, and affine transformation of $\mathcal{H}$-polytopes.

Compatibility vs Transversality

Compatibility vs Transversality

Proposition (T.-So '25)

For convex polytopes A and B, the following hold:

- *Transversality of A and B implies compatibility.*
- *If A and B are zonotopes, compatibility implies transversality.*

Remarks.

- transversality is an efficiently verifiable sufficient condition.

Compatibility vs Transversality

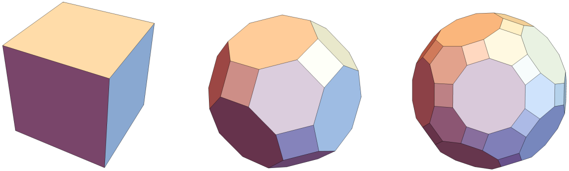
Proposition (T.-So '25)

For convex polytopes A and B, the following hold:

- *Transversality of A and B implies compatibility.*
- *If A and B are zonotopes, compatibility implies transversality.*

Remarks.

- transversality is an efficiently verifiable sufficient condition.
- zonotopes:
  - a subclass of polytopes generated by **sum of segments**.



Compatibility vs Transversality

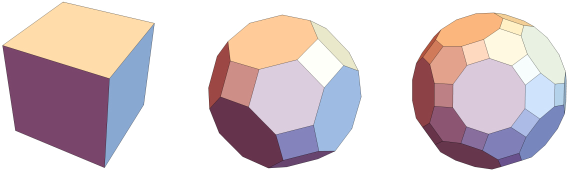
Proposition (T.-So '25)

For convex polytopes A and B, the following hold:

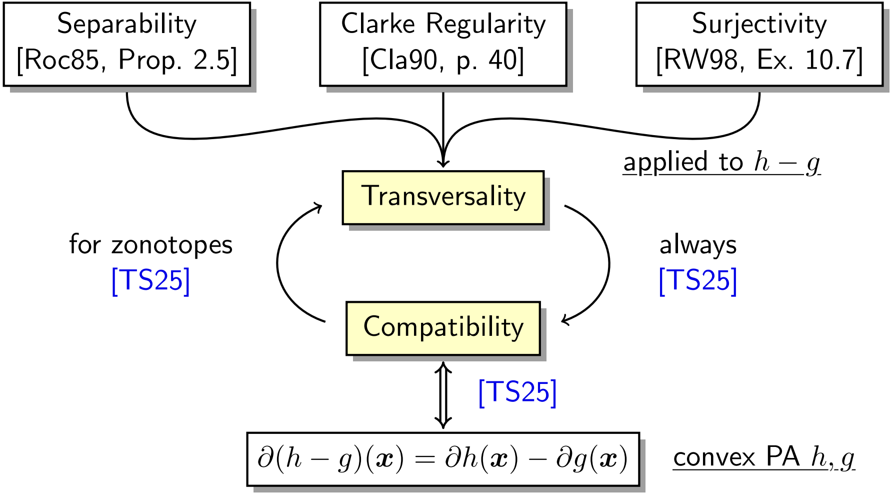
- *Transversality of A and B implies compatibility.*
- *If A and B are zonotopes, compatibility implies transversality.*

Remarks.

- transversality is an efficiently verifiable sufficient condition.
- zonotopes:
  - a subclass of polytopes generated by **sum of segments**.
  - two-layer ReLU networks, ρ-margin loss SVM, penalized deep ReLU networks, etc.



Interrelations of Various Conditions



Outline

Robust Testing: Motivation

- For now, we only discuss <u>exact</u> testing $\mathbf{0} \in \partial f(\boldsymbol{x}) + \varepsilon\mathbb{B}$.

Robust Testing: Motivation

- For now, we only discuss exact testing $\mathbf{0} \in \partial f(\boldsymbol{x}) + \varepsilon\mathbb{B}$.
- In practice, iterations may never hit a nonsmooth point.
  - Randomness/finite-precision in algorithm.
  - Close to, but never hit (consider $x \mapsto |x|$).

Robust Testing: Motivation

- For now, we only discuss exact testing $\mathbf{0} \in \partial f(\boldsymbol{x}) + \varepsilon\mathbb{B}$.
- In practice, iterations may never hit a nonsmooth point.
  - Randomness/finite-precision in algorithm.
  - Close to, but never hit (consider $x \mapsto |x|$).
- To be practical, we need a robust testing approach.
  - If $\boldsymbol{w}$ is sufficiently (δ-)close to an ε-stationary $\boldsymbol{w}^*$, certify

$$\mathbf{0} \in \partial f(\boldsymbol{w} + \delta\mathbb{B}) + \varepsilon\mathbb{B}.$$

- To **separate the difficulties** in exact/robust testing, we use an **oracle**:
  - Given f, $\boldsymbol{x}$, and $\varepsilon \geq 0$, the oracle decides whether $\mathbf{0} \in \partial f(\boldsymbol{x}) + \varepsilon\mathbb{B}$.

Main Result III: Robust Testing

Corollary (T.-So '25)

Consider $\{\boldsymbol{x}_n\}_n$ produced by the subgradient method on $h-g$. For any $\varepsilon \geq 0$ and $\delta > 0$, the stopping criterion

$$\mathbf{0} \in \partial(h-g)(\boldsymbol{x}_T + \delta\mathbb{B}) + \varepsilon\mathbb{B}$$

can be certified for a finite $T \in \mathbb{N}_+$ in (oracle) polynomial time.

Remarks.

- inspired by the termination of LPs.

• Efficiently testing local optimality and escaping saddles for ReLU networks, ICLR '19.

Main Result III: Robust Testing

Corollary (T.-So '25)

Consider $\{\boldsymbol{x}_n\}_n$ produced by the subgradient method on $h - g$. For any $\varepsilon \geq 0$ and $\delta > 0$, the stopping criterion

$$\mathbf{0} \in \partial(h-g)(\boldsymbol{x}_T + \delta\mathbb{B}) + \varepsilon\mathbb{B}$$

can be certified for a finite $T \in \mathbb{N}_+$ in (oracle) polynomial time.

Remarks.

- inspired by the termination of LPs.
- applicable to **any** algorithm with asymptotic convergence.

• Efficiently testing local optimality and escaping saddles for ReLU networks, ICLR '19.

Main Result III: Robust Testing

Corollary (T.-So '25)

Consider $\{\boldsymbol{x}_n\}_n$ produced by the subgradient method on $h - g$. For any $\varepsilon \geq 0$ and $\delta > 0$, the stopping criterion

$$\mathbf{0} \in \partial(h - g)(\boldsymbol{x}_T + \delta\mathbb{B}) + \varepsilon\mathbb{B}$$

can be certified for a finite $T \in \mathbb{N}_+$ in (oracle) polynomial time.

Remarks.

- inspired by the termination of LPs.
- applicable to **any** algorithm with asymptotic convergence.
- when specialized to two-layer ReLU networks, this corollary resolves the open problem mentioned in (Yun et al. '19).

• Efficiently testing local optimality and escaping saddles for ReLU networks, ICLR '19.

Outline

Summary

I. Testing stationarity and local optimality of a point for PA functions are computationally intractable, unless $\mathsf{P} = \mathsf{NP}$.

Summary

I. Testing stationarity and local optimality of a point for PA functions are computationally intractable, unless P = NP.

II. Compatibility is a necessary and sufficient (geometric) condition validating exact sum rule, which facilitates efficient ε-stationarity testing.

Summary

I. Testing stationarity and local optimality of a point for PA functions are computationally intractable, unless $\mathsf{P} = \mathsf{NP}$.

II. Compatibility is a necessary and sufficient (geometric) condition validating exact sum rule, which facilitates efficient ε-stationarity testing.

III. Using an ε-stationarity testing oracle, we can check (ε, δ)-NAS points in polynomial time, which is a universal stopping rule.

Summary

I. Testing stationarity and local optimality of a point for PA functions are computationally intractable, unless $\mathrm{P} = \mathrm{NP}$.

II. Compatibility is a necessary and sufficient (geometric) condition validating exact sum rule, which facilitates efficient ε-stationarity testing.

III. Using an ε-stationarity testing oracle, we can check (ε, δ)-NAS points in polynomial time, which is a universal stopping rule.

Thank You! Questions?