First-order optimization without (much) geometry

Adrian Lewis

Third Sevilla MINLP Workshop

March 2023

Outline: basic tools for optimization

Outline: basic tools for optimization

I: First-order methods in metric space (with Tonghua Tian)

Outline: basic tools for optimization

I: First-order methods in metric space (with Tonghua Tian)

- Slope and sufficient decrease

Outline: basic tools for optimization

I: First-order methods in metric space (with Tonghua Tian)

- Slope and sufficient decrease
- The Kurdyka-Łojasiewicz inequality

Outline: basic tools for optimization

I: First-order methods in metric space (with Tonghua Tian)

- Slope and sufficient decrease
- The Kurdyka-Łojasiewicz inequality

II: Alternating projections using geodesics
(with G. López-Acedo and A. Nicolae)

Outline: basic tools for optimization

I: First-order methods in metric space (with Tonghua Tian)

- Slope and sufficient decrease
- The Kurdyka-Łojasiewicz inequality

II: Alternating projections using geodesics (with G. López-Acedo and A. Nicolae)

- Curvature and transversality

Outline: basic tools for optimization

I: First-order methods in metric space (with Tonghua Tian)

- Slope and sufficient decrease
- The Kurdyka-Łojasiewicz inequality

II: Alternating projections using geodesics (with G. López-Acedo and A. Nicolae)

- Curvature and transversality
- Linear convergence

Outline: basic tools for optimization

I: First-order methods in metric space (with Tonghua Tian)

- Slope and sufficient decrease
- The Kurdyka-Łojasiewicz inequality

II: Alternating projections using geodesics (with G. López-Acedo and A. Nicolae)

- Curvature and transversality
- Linear convergence

III: Subgradient oracles (with Siyu Kong)

Outline: basic tools for optimization

I: First-order methods in metric space (with Tonghua Tian)

- Slope and sufficient decrease
- The Kurdyka-Łojasiewicz inequality

II: Alternating projections using geodesics (with G. López-Acedo and A. Nicolae)

- Curvature and transversality
- Linear convergence

III: Subgradient oracles (with Siyu Kong)

- How hard is Lipschitz optimization?

Outline: basic tools for optimization

I: First-order methods in metric space (with Tonghua Tian)

- Slope and sufficient decrease
- The Kurdyka-Łojasiewicz inequality

II: Alternating projections using geodesics (with G. López-Acedo and A. Nicolae)

- Curvature and transversality
- Linear convergence

III: Subgradient oracles (with Siyu Kong)

- How hard is Lipschitz optimization?
- How much does nonconvexity cost?

PART I

First-order methods in metric space

(with Tonghua Tian)

Slope and the Ekeland Principle

Slope and the Ekeland Principle

Setting

- Complete metric space X

Slope and the Ekeland Principle

Setting

- Complete metric space X
- Lower semicontinuous $f: X \to \overline{\mathbf{R}}$

Slope and the Ekeland Principle

Setting

- Complete metric space X
- Lower semicontinuous $f : X \to \overline{\mathbf{R}}$ . . .
- . . . with finite infimum.

Slope and the Ekeland Principle

Setting

- Complete metric space X
- Lower semicontinuous $f : X \to \overline{\mathbf{R}}$ . . .
- . . . with finite infimum. Define **gap** $g = f - \inf f$.

Slope and the Ekeland Principle

Setting

- Complete metric space X
- Lower semicontinuous $f\colon X \to \overline{\mathbf{R}}$ . . .
- . . . with finite infimum. Define **gap** $g = f - \inf f$.

At $x \in X$, **slope** $|\nabla f|(x)$ is
the infimum of $\epsilon > 0$ such that
x locally minimizes

$$f(\cdot) + \epsilon \operatorname{dist}(x, \cdot).$$

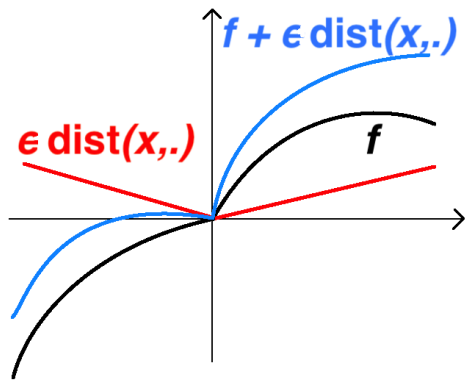
Slope and the Ekeland Principle

Setting

- Complete metric space X
- Lower semicontinuous $f : X \to \overline{\mathbf{R}}$ ...
- ... with finite infimum. Define **gap** $g = f - \inf f$.

At $x \in X$, **slope** $|\nabla f|(x)$ is the infimum of $\epsilon > 0$ such that x locally minimizes

$$f(\cdot) + \epsilon \operatorname{dist}(x, \cdot).$$



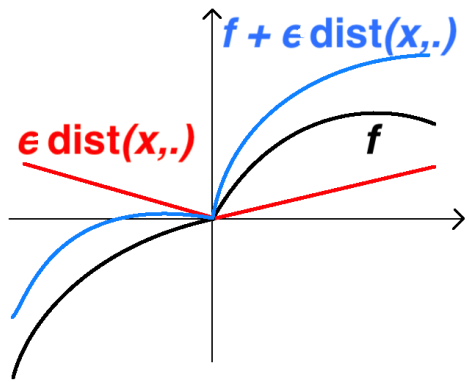
Slope and the Ekeland Principle

Setting

- Complete metric space X
- Lower semicontinuous $f : X \to \overline{\mathbf{R}}$ ...
- ... with finite infimum. Define **gap** $g = f - \inf f$.

At $x \in X$, **slope** $|\nabla f|(x)$ is the infimum of $\epsilon > 0$ such that x locally minimizes

$$f(\cdot) + \epsilon \operatorname{dist}(x, \cdot).$$



Examples

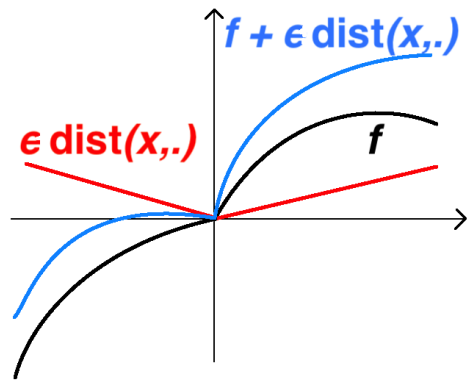
Slope and the Ekeland Principle

Setting

- Complete metric space X
- Lower semicontinuous $f : X \to \overline{\mathbf{R}}$ ...
- ... with finite infimum. Define **gap** $g = f - \inf f$.

At $x \in X$, **slope** $|\nabla f|(x)$ is
the infimum of $\epsilon > 0$ such that
x locally minimizes

$$f(\cdot) + \epsilon \operatorname{dist}(x, \cdot).$$



Examples If X Euclidean, $|\nabla f|(x)$ is

$$\|\nabla f(x)\| \qquad \text{if } f \text{ is smooth}$$

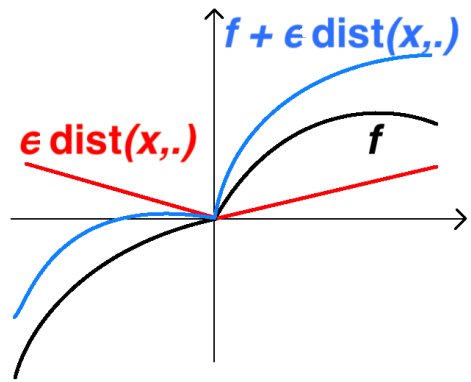
Slope and the Ekeland Principle

Setting

- Complete metric space X
- Lower semicontinuous $f\colon X \to \overline{\mathbf{R}}$ ...
- ... with finite infimum. Define **gap** $g = f - \inf f$.

At $x \in X$, **slope** $|\nabla f|(x)$ is
the infimum of $\epsilon > 0$ such that
x locally minimizes

$$f(\cdot) + \epsilon \operatorname{dist}(x, \cdot).$$



Examples

If X Euclidean, $|\nabla f|(x)$ is

$$\begin{aligned} \|\nabla f(x)\| &\quad \text{if } f \text{ is smooth} \\ \operatorname{dist}\big(0, \partial f(x)\big) &\quad \text{if } f \text{ is convex} \end{aligned}$$

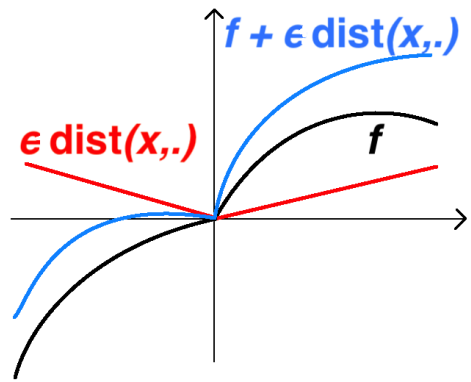
Slope and the Ekeland Principle

Setting

- Complete metric space X
- Lower semicontinuous $f\colon X \to \overline{\mathbf{R}}$ ...
- ... with finite infimum. Define **gap** $g = f - \inf f$.

At $x \in X$, **slope** $|\nabla f|(x)$ is
the infimum of $\epsilon > 0$ such that
x locally minimizes

$$f(\cdot) + \epsilon\,\mathrm{dist}(x, \cdot).$$



Examples If X Euclidean, $|\nabla f|(x)$ is

$$\begin{aligned} \|\nabla f(x)\| &\quad \text{if } f \text{ is smooth} \\ \mathrm{dist}\big(0, \partial f(x)\big) &\quad \text{if } f \text{ is convex} \end{aligned}$$

Theorem (Ekeland '74 **"existence without compactness"**)

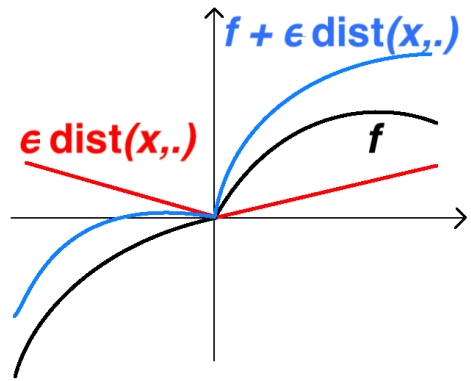
Slope and the Ekeland Principle

Setting

- Complete metric space X
- Lower semicontinuous $f\colon X \to \overline{\mathbf{R}}$ ...
- ... with finite infimum. Define **gap** $g = f - \inf f$.

At $x \in X$, **slope** $|\nabla f|(x)$ is
the infimum of $\epsilon > 0$ such that
x locally minimizes

$$f(\cdot) + \epsilon \operatorname{dist}(x, \cdot).$$



Examples

If X Euclidean, $|\nabla f|(x)$ is

$$\begin{aligned} \|\nabla f(x)\| &\quad \text{if } f \text{ is smooth} \\ \operatorname{dist}\big(0, \partial f(x)\big) &\quad \text{if } f \text{ is convex} \end{aligned}$$

Theorem (Ekeland '74 **"existence without compactness"**)

There exists points where both the gap and slope are small.

Sufficient decrease

Sufficient decrease

Many algorithms generate iterates in X

$$\ldots, \; x \; , \; x_+ \; , \; \ldots$$

satisfying **slope-sufficient decrease**:

Sufficient decrease

Many algorithms generate iterates in X

$$\ldots, \; x \; , \; x_+ \; , \; \ldots$$

satisfying **slope-sufficient decrease**: for some $\delta > 0$

$$f(x) - f(x_+) \; \geq \; \delta\big(|\nabla f|(x_+)\big)^2 \qquad \text{always.}$$

Sufficient decrease

Many algorithms generate iterates in X

$$\ldots, \; x \; , \; x_+ \; , \; \ldots$$

satisfying **slope-sufficient decrease**: for some $\delta > 0$

$$f(x) - f(x_+) \;\geq\; \delta\big(|\nabla f|(x_+)\big)^2 \qquad \text{always.}$$

Euclidean examples: proximal point, gradient descent, proximal gradient, variable metric, trust region, alternating projections...

Sufficient decrease

Many algorithms generate iterates in X

$$\ldots, \; x \; , \; x_+ \; , \; \ldots$$

satisfying **slope-sufficient decrease**: for some $\delta > 0$

$$f(x) - f(x_+) \; \geq \; \delta\big(|\nabla f|(x_+)\big)^2 \qquad \text{always.}$$

Euclidean examples: proximal point, gradient descent, proximal gradient, variable metric, trust region, alternating projections...

Ekeland Principle guarantees...

Corollary

For any x, some x_+ satisfies slope-sufficient decrease.

Convergence in value

Convergence in value

Assumption (Kurdyka-Łojasiewicz inequality)

Convergence in value

Assumption (Kurdyka-Łojasiewicz inequality)

Some continuous concave $\phi\colon \mathbf{R}_+ \to \mathbf{R}_+$

Convergence in value

Assumption (Kurdyka-Łojasiewicz inequality)

Some continuous concave $\phi\colon \mathbf{R}_+ \to \mathbf{R}_+$ *with* $\phi(0) = 0$ *and with continuous derivative* $\phi' > 0$ *on* $\mathbf{R}_{++}$ *satisfying* $\phi'(t) = o(\frac{1}{t})$ *as* $t \downarrow 0$

Convergence in value

Assumption (Kurdyka-Łojasiewicz inequality)

Some continuous concave $\phi\colon \mathbf{R}_+ \to \mathbf{R}_+$ *with* $\phi(0) = 0$ *and with continuous derivative* $\phi' > 0$ *on* $\mathbf{R}_{++}$ *satisfying* $\phi'(t) = o(\frac{1}{t})$ *as* $t \downarrow 0$ **desingularizes** the gap
to induce **sharpness**:

Convergence in value

Assumption (Kurdyka-Łojasiewicz inequality)

Some continuous concave $\phi\colon \mathbf{R}_+ \to \mathbf{R}_+$ *with* $\phi(0) = 0$ *and with continuous derivative* $\phi' > 0$ *on* $\mathbf{R}_{++}$ *satisfying* $\phi'(t) = o(\frac{1}{t})$ *as* $t \downarrow 0$ **desingularizes** the gap
to induce **sharpness**:

$$|\nabla(\phi \circ g)|(x) \;\geq\; 1$$

whenever $g(x)$ is small and nonzero.

Convergence in value

Assumption (Kurdyka-Łojasiewicz inequality)

Some continuous concave $\phi\colon \mathbf{R}_+ \to \mathbf{R}_+$ *with* $\phi(0) = 0$ *and with continuous derivative* $\phi' > 0$ *on* $\mathbf{R}_{++}$ *satisfying* $\phi'(t) = o(\frac{1}{t})$ *as* $t \downarrow 0$ **desingularizes** the gap
to induce **sharpness**:

$$|\nabla(\phi \circ g)|(x) \;\geq\; 1$$

whenever $g(x)$ is small and nonzero.

Example: On $\mathbf{R}$, for
$f(x) = g(x) = x^2$, choose $\phi(t) = \sqrt{t}$.

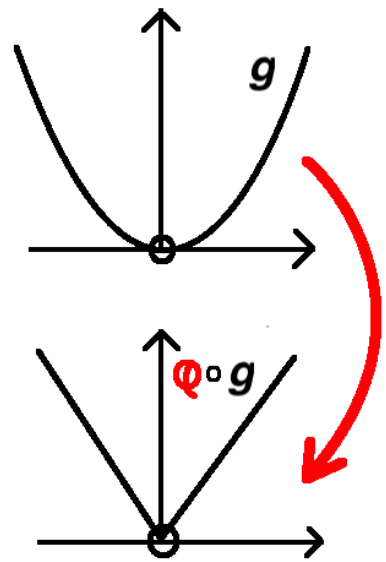
Convergence in value

Assumption (Kurdyka-Łojasiewicz inequality)

Some continuous concave $\phi\colon \mathbf{R}_+ \to \mathbf{R}_+$ *with* $\phi(0) = 0$ *and with continuous derivative* $\phi' > 0$ *on* $\mathbf{R}_{++}$ *satisfying* $\phi'(t) = o(\frac{1}{t})$ *as* $t \downarrow 0$ **desingularizes** the gap
to induce **sharpness**:

$$|\nabla(\phi \circ g)|(x) \;\geq\; 1$$

whenever $g(x)$ is small and nonzero.



Example: On $\mathbf{R}$, for
$f(x) = g(x) = x^2$, choose $\phi(t) = \sqrt{t}$.

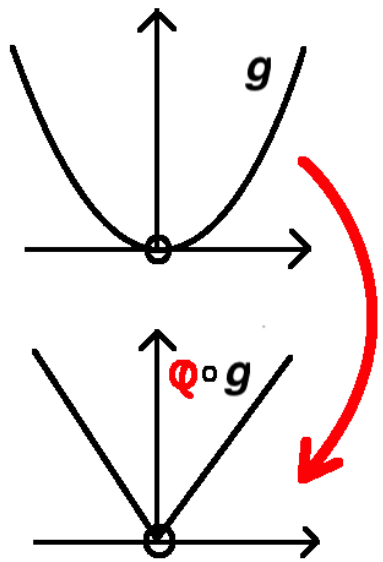
Convergence in value

Assumption (Kurdyka-Łojasiewicz inequality)

Some continuous concave $\phi\colon \mathbf{R}_+ \to \mathbf{R}_+$ *with* $\phi(0) = 0$ *and with continuous derivative* $\phi' > 0$ *on* $\mathbf{R}_{++}$ *satisfying* $\phi'(t) = o(\frac{1}{t})$ *as* $t \downarrow 0$ **desingularizes** the gap
to induce **sharpness**:

$$|\nabla(\phi \circ g)|(x) \;\geq\; 1$$

whenever $g(x)$ is small and nonzero.



Example: On $\mathbf{R}$, for
$f(x) = g(x) = x^2$, choose $\phi(t) = \sqrt{t}$.

Proposition

If **KL** *holds and initial gap small, then gap* $< \epsilon$ *after* $o(\frac{1}{\epsilon})$ *steps.*

The Kurdyka-Łojasiewicz inequality on $\mathbf{R}^n$

The Kurdyka-Łojasiewicz inequality on $\mathbf{R}^n$

- Real-analytic f,

The Kurdyka-Łojasiewicz inequality on $\mathbf{R}^n$

- Real-analytic f, with desingularizer

$$\phi(t) = \beta t^{\theta} \qquad \text{with} \qquad \theta \in (0, 1] \text{ and } \beta > 0.$$

The Kurdyka-Łojasiewicz inequality on $\mathbf{R}^n$

- Real-analytic f, with desingularizer

$$\phi(t) = \beta t^{\theta} \qquad \text{with} \qquad \theta \in (0, 1] \text{ and } \beta > 0.$$

(Łojasiewicz 1965)

The Kurdyka-Łojasiewicz inequality on $\mathbf{R}^n$

- Real-analytic f, with desingularizer

$$\phi(t) = \beta t^{\theta} \qquad \text{with} \qquad \theta \in (0, 1] \text{ and } \beta > 0.$$

(Łojasiewicz 1965)

- **Semi-algebraic** f with bounded minimizing set.

The Kurdyka-Łojasiewicz inequality on $\mathbf{R}^n$

- Real-analytic f, with desingularizer

$$\phi(t) = \beta t^{\theta} \qquad \text{with} \qquad \theta \in (0, 1] \text{ and } \beta > 0.$$

(Łojasiewicz 1965)

- **Semi-algebraic** f with bounded minimizing set.
(Graphs partition into sets defined by polynomial inequalities.)

The Kurdyka-Łojasiewicz inequality on $\mathbf{R}^n$

- Real-analytic f, with desingularizer

$$\phi(t) = \beta t^{\theta} \qquad \text{with} \qquad \theta \in (0, 1] \text{ and } \beta > 0.$$

(Łojasiewicz 1965)

- **Semi-algebraic** f with bounded minimizing set.
  (Graphs partition into sets defined by polynomial inequalities.)
  Same desingularizer works, for semi-algebraic or subanalytic f

The Kurdyka-Łojasiewicz inequality on $\mathbf{R}^n$

- Real-analytic f, with desingularizer

$$\phi(t) = \beta t^{\theta} \qquad \text{with} \qquad \theta \in (0, 1] \text{ and } \beta > 0.$$

(Łojasiewicz 1965)

- **Semi-algebraic** f with bounded minimizing set.
  (Graphs partition into sets defined by polynomial inequalities.)
  Same desingularizer works, for semi-algebraic or subanalytic f
  (Bolte-Daniilidis-L-Shiota 2007).

The Kurdyka-Łojasiewicz inequality on $\mathbf{R}^n$

- Real-analytic f, with desingularizer

$$\phi(t) = \beta t^{\theta} \qquad \text{with} \qquad \theta \in (0, 1] \text{ and } \beta > 0.$$

(Łojasiewicz 1965)

- **Semi-algebraic** f with bounded minimizing set.
  (Graphs partition into sets defined by polynomial inequalities.)
  Same desingularizer works, for semi-algebraic or subanalytic f
  (Bolte-Daniilidis-L-Shiota 2007).

- "Tame" f,

The Kurdyka-Łojasiewicz inequality on $\mathbf{R}^n$

- Real-analytic f, with desingularizer

$$\phi(t) = \beta t^{\theta} \qquad \text{with} \qquad \theta \in (0, 1] \text{ and } \beta > 0.$$

(Łojasiewicz 1965)

- **Semi-algebraic** f with bounded minimizing set.
  (Graphs partition into sets defined by polynomial inequalities.)
  Same desingularizer works, for semi-algebraic or subanalytic f
  (Bolte-Daniilidis-L-Shiota 2007).

- "Tame" f, smooth or nonsmooth

The Kurdyka-Łojasiewicz inequality on $\mathbf{R}^n$

- Real-analytic f, with desingularizer

$$\phi(t) = \beta t^{\theta} \qquad \text{with} \qquad \theta \in (0, 1] \text{ and } \beta > 0.$$

(Łojasiewicz 1965)

- **Semi-algebraic** f with bounded minimizing set.
  (Graphs partition into sets defined by polynomial inequalities.)
  Same desingularizer works, for semi-algebraic or subanalytic f
  (Bolte-Daniilidis-L-Shiota 2007).

- "Tame" f, smooth or nonsmooth (Kurdyka 1998,

The Kurdyka-Łojasiewicz inequality on $\mathbf{R}^n$

- Real-analytic f, with desingularizer

$$\phi(t) = \beta t^{\theta} \qquad \text{with} \qquad \theta \in (0, 1] \text{ and } \beta > 0.$$

(Łojasiewicz 1965)

- **Semi-algebraic** f with bounded minimizing set.
  (Graphs partition into sets defined by polynomial inequalities.)
  Same desingularizer works, for semi-algebraic or subanalytic f
  (Bolte-Daniilidis-L-Shiota 2007).

- "Tame" f, smooth or nonsmooth (Kurdyka 1998, BDLS 2007)

The Kurdyka-Łojasiewicz inequality on $\mathbf{R}^n$

- Real-analytic f, with desingularizer

$$\phi(t) = \beta t^{\theta} \qquad \text{with} \qquad \theta \in (0, 1] \text{ and } \beta > 0.$$

(Łojasiewicz 1965)

- **Semi-algebraic** f with bounded minimizing set.
  (Graphs partition into sets defined by polynomial inequalities.)
  Same desingularizer works, for semi-algebraic or subanalytic f
  (Bolte-Daniilidis-L-Shiota 2007).

- "Tame" f, smooth or nonsmooth (Kurdyka 1998, BDLS 2007)

- Hilbert and metric versions

The Kurdyka-Łojasiewicz inequality on $\mathbf{R}^n$

- Real-analytic f, with desingularizer

$$\phi(t) = \beta t^{\theta} \qquad \text{with} \qquad \theta \in (0, 1] \text{ and } \beta > 0.$$

(Łojasiewicz 1965)

- **Semi-algebraic** f with bounded minimizing set.
  (Graphs partition into sets defined by polynomial inequalities.)
  Same desingularizer works, for semi-algebraic or subanalytic f
  (Bolte-Daniilidis-L-Shiota 2007).

- "Tame" f, smooth or nonsmooth (Kurdyka 1998, BDLS 2007)

- Hilbert and metric versions (Simon 1983,

The Kurdyka-Łojasiewicz inequality on $\mathbf{R}^n$

- Real-analytic f, with desingularizer

$$\phi(t) = \beta t^{\theta} \qquad \text{with} \qquad \theta \in (0, 1] \text{ and } \beta > 0.$$

(Łojasiewicz 1965)

- **Semi-algebraic** f with bounded minimizing set.
  (Graphs partition into sets defined by polynomial inequalities.)
  Same desingularizer works, for semi-algebraic or subanalytic f
  (Bolte-Daniilidis-L-Shiota 2007).

- "Tame" f, smooth or nonsmooth (Kurdyka 1998, BDLS 2007)

- Hilbert and metric versions (Simon 1983, Hauer-Mazón 2019)

Slope-descent sequences

Slope-descent sequences

Typical descent is stronger:

Slope-descent sequences

Typical descent is stronger: for $\alpha, \beta > 0$, successive iterates satisfy

$$f(x) - f(x_+) \quad \geq \quad \alpha \left(\mathsf{dist}(x, x_+)\right)^2$$

Slope-descent sequences

Typical descent is stronger: for $\alpha, \beta > 0$, successive iterates satisfy

$$\begin{aligned} f(x) - f(x_+) &\geq \alpha \left(\mathsf{dist}(x, x_+)\right)^2 \qquad \text{and} \\ \mathsf{dist}(x, x_+) &\geq \beta \, |\nabla f|(x_+) \end{aligned}$$

Slope-descent sequences

Typical descent is stronger: for $\alpha, \beta > 0$, successive iterates satisfy

$$\begin{aligned} f(x) - f(x_+) &\geq \alpha \left(\operatorname{dist}(x, x_+)\right)^2 \qquad \text{and} \\ \operatorname{dist}(x, x_+) &\geq \beta \, |\nabla f|(x_+) \end{aligned}$$

Example: Proximal point method in $\mathbf{R}^n$

$$x_+ \quad \text{minimizes} \quad f(\cdot) + \frac{1}{2}\| \cdot - x\|^2$$

Slope-descent sequences

Typical descent is stronger: for $\alpha, \beta > 0$, successive iterates satisfy

$$\begin{aligned} f(x) - f(x_+) &\geq \alpha \left(\operatorname{dist}(x, x_+)\right)^2 \qquad \text{and} \\ \operatorname{dist}(x, x_+) &\geq \beta \, |\nabla f|(x_+) \end{aligned}$$

Example: Proximal point method in $\mathbf{R}^n$

$$x_+ \quad \text{minimizes} \quad f(\cdot) + \frac{1}{2}\| \cdot - x\|^2$$

so satisfies $0 \in \partial f(x_+) + x_+ - x$,

Slope-descent sequences

Typical descent is stronger: for $\alpha, \beta > 0$, successive iterates satisfy

$$\begin{aligned} f(x) - f(x_+) &\geq \alpha \left(\operatorname{dist}(x, x_+)\right)^2 \qquad \text{and} \\ \operatorname{dist}(x, x_+) &\geq \beta \, |\nabla f|(x_+) \end{aligned}$$

Example: Proximal point method in $\mathbf{R}^n$

$$x_+ \quad \text{minimizes} \quad f(\cdot) + \frac{1}{2}\| \cdot - x\|^2$$

so satisfies $0 \in \partial f(x_+) + x_+ - x$, and hence

$$\|x - x_+\| \quad \geq \quad |\nabla f|(x_+)$$

Slope-descent sequences

Typical descent is stronger: for $\alpha, \beta > 0$, successive iterates satisfy

$$\begin{aligned} f(x) - f(x_+) &\geq \alpha \left(\operatorname{dist}(x, x_+)\right)^2 \qquad \text{and} \\ \operatorname{dist}(x, x_+) &\geq \beta \, |\nabla f|(x_+) \end{aligned}$$

Example: Proximal point method in $\mathbf{R}^n$

$$x_+ \quad \text{minimizes} \quad f(\cdot) + \frac{1}{2}\| \cdot - x\|^2$$

so satisfies $0 \in \partial f(x_+) + x_+ - x$, and hence

$$\begin{aligned} \|x - x_+\| &\geq |\nabla f|(x_+) \\ f(x) - f(x_+) &\geq \frac{1}{2}\|x - x_+\|^2 \end{aligned}$$

Slope-descent sequences

Typical descent is stronger: for $\alpha, \beta > 0$, successive iterates satisfy

$$\begin{aligned} f(x) - f(x_+) &\geq \alpha \big(\mathrm{dist}(x, x_+)\big)^2 \qquad \text{and} \\ \mathrm{dist}(x, x_+) &\geq \beta\, |\nabla f|(x_+) \end{aligned}$$

Example: Proximal point method in $\mathbf{R}^n$

$$x_+ \quad \text{minimizes} \quad f(\cdot) + \frac{1}{2}\|\cdot - x\|^2$$

so satisfies $0 \in \partial f(x_+) + x_+ - x$, and hence

$$\begin{aligned} \|x - x_+\| &\geq |\nabla f|(x_+) \\ f(x) - f(x_+) &\geq \frac{1}{2}\|x - x_+\|^2 \end{aligned}$$

Corollary (from Ekeland)

If $\alpha\beta < \frac{1}{2}$, then for any x, some x_+ satisfies slope descent.

Convergence of iterates

Convergence of iterates

Following (Attouch-Bolte-Svaiter 2013)...

Convergence of iterates

Following (Attouch-Bolte-Svaiter 2013)...

Theorem

Assuming **KL** *and small initial gap, any slope-descent sequence* (x_k) *converges to a minimizer* x^*.

Convergence of iterates

Following (Attouch-Bolte-Svaiter 2013)...

Theorem

Assuming **KL** *and small initial gap, any slope-descent sequence* (x_k) *converges to a minimizer* x^*.

If the desingularizer is $\phi(t) = \beta t^\theta$, for $\theta \in [0, \frac{1}{2})$, then

$$\text{dist}(x_k, x^*) \;<\; \epsilon$$

after

$$O\left(\left(\frac{1}{\epsilon}\right)^{\frac{1}{\theta}-2}\right)$$

steps.

PART II

Alternating projections using geodesics

PART II

Alternating projections using geodesics

(. . . recapping Adriana Nicolae's talk with G. López-Acedo)

The setting

Neighborhood of $\bar{z} \in Z$,
complete **geodesic** metric space

[1] commons.wikimedia.org/w/index.php?curid=647033

The setting

Neighborhood of $\bar{z} \in Z$,
complete **geodesic** metric space

(those where points connect via
arcs isometric to compact intervals).

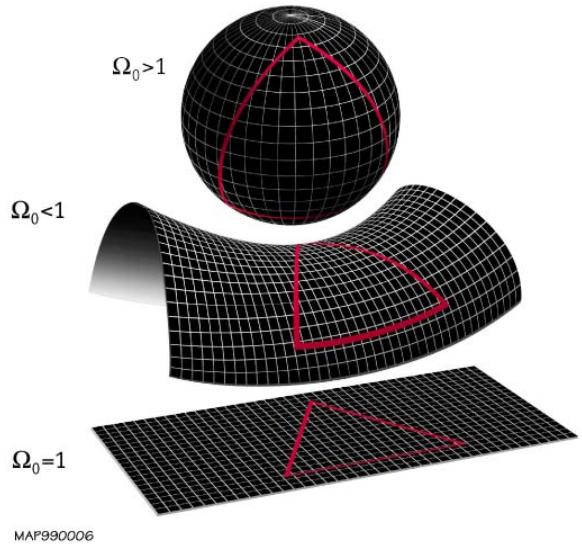
[1] commons.wikimedia.org/w/index.php?curid=647033

The setting

Neighborhood of $\bar{z} \in Z$,
complete **geodesic** metric space

(those where points connect via
arcs isometric to compact intervals).

Eg: **manifolds** $Z = F^{-1}(0)$
for smooth $F\colon \mathbf{R}^n \to \mathbf{R}^m$ with
$F(\bar{z}) = 0$ and $\nabla F(\bar{z})$ onto.



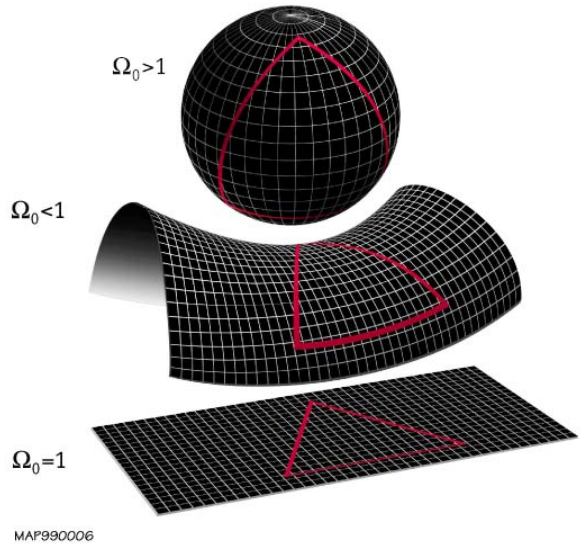
[1] commons.wikimedia.org/w/index.php?curid=647033

The setting

Neighborhood of $\bar{z} \in Z$,
complete **geodesic** metric space

(those where points connect via
arcs isometric to compact intervals).

Eg: **manifolds** $Z = F^{-1}(0)$
for smooth $F: \mathbf{R}^n \to \mathbf{R}^m$ with
$F(\bar{z}) = 0$ and $\nabla F(\bar{z})$ onto.



Consider von Neumann's
alternating projection iteration
for closed subsets X, Y

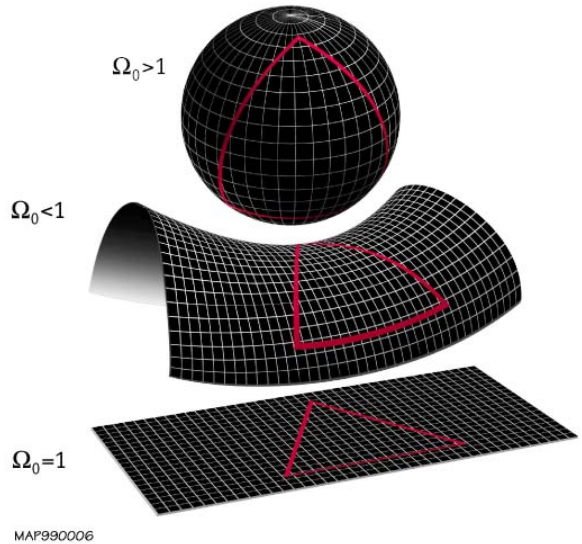
[1] commons.wikimedia.org/w/index.php?curid=647033

The setting

Neighborhood of $\bar{z} \in Z$,
complete **geodesic** metric space

(those where points connect via
arcs isometric to compact intervals).

Eg: **manifolds** $Z = F^{-1}(0)$
for smooth $F: \mathbf{R}^n \to \mathbf{R}^m$ with
$F(\bar{z}) = 0$ and $\nabla F(\bar{z})$ onto.



Consider von Neumann's
alternating projection iteration
for closed subsets X, Y
near a common point $\bar{z}$.

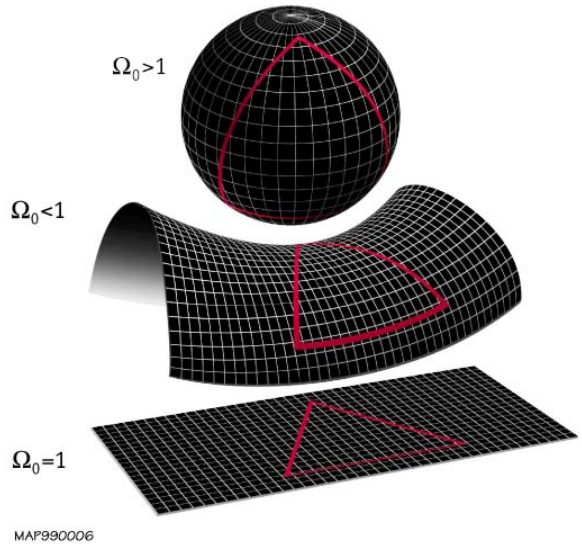
[1] commons.wikimedia.org/w/index.php?curid=647033

The setting

Neighborhood of $\bar{z} \in Z$,
complete **geodesic** metric space

(those where points connect via
arcs isometric to compact intervals).

Eg: **manifolds** $Z = F^{-1}(0)$
for smooth $F: \mathbf{R}^n \to \mathbf{R}^m$ with
$F(\bar{z}) = 0$ and $\nabla F(\bar{z})$ onto.



Consider von Neumann's
alternating projection iteration
for closed subsets X, Y
near a common point $\bar{z}$.

Does it converge?

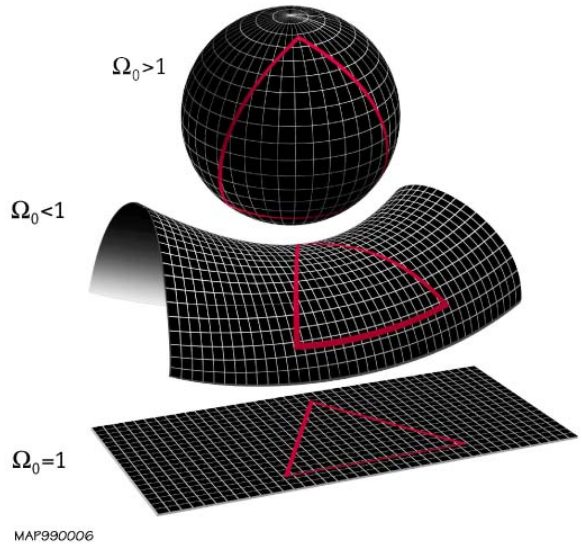
[1] commons.wikimedia.org/w/index.php?curid=647033

The setting

Neighborhood of $\bar{z} \in Z$,
complete **geodesic** metric space

(those where points connect via
arcs isometric to compact intervals).

Eg: **manifolds** $Z = F^{-1}(0)$
for smooth $F: \mathbf{R}^n \to \mathbf{R}^m$ with
$F(\bar{z}) = 0$ and $\nabla F(\bar{z})$ onto.



Consider von Neumann's
alternating projection iteration
for closed subsets X, Y
near a common point $\bar{z}$.

Does it converge? How fast?

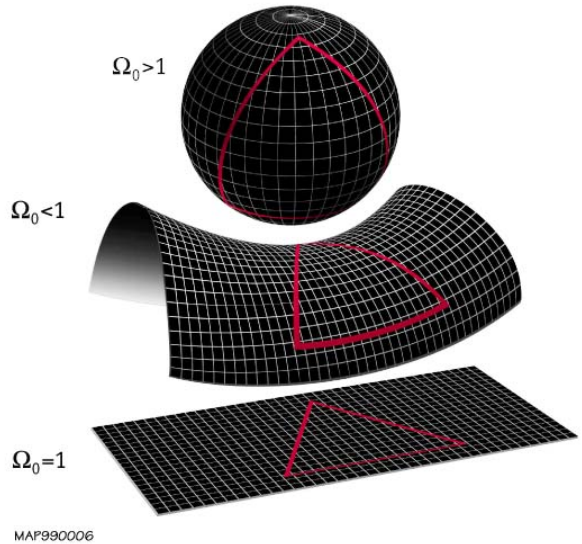
[1] commons.wikimedia.org/w/index.php?curid=647033

The setting

Neighborhood of $\bar{z} \in Z$,
complete **geodesic** metric space

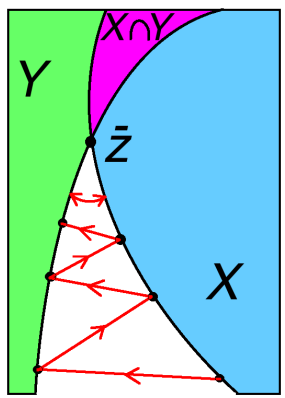
(those where points connect via
arcs isometric to compact intervals).

Eg: **manifolds** $Z = F^{-1}(0)$
for smooth $F: \mathbf{R}^n \to \mathbf{R}^m$ with
$F(\bar{z}) = 0$ and $\nabla F(\bar{z})$ onto.



Consider von Neumann's
alternating projection iteration
for closed subsets X, Y
near a common point $\bar{z}$.

Does it converge? How fast?



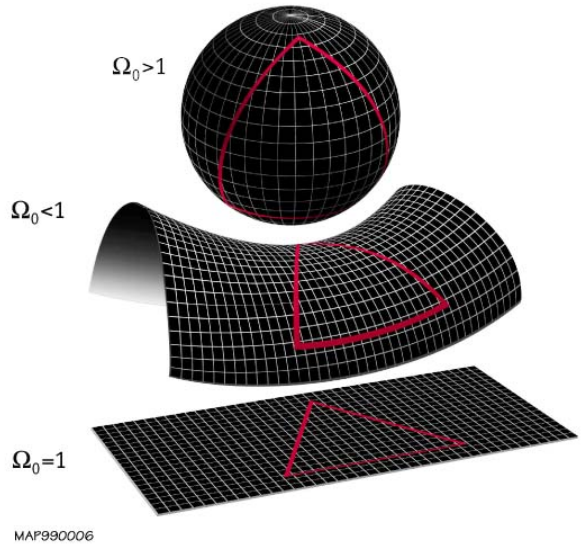
[1] commons.wikimedia.org/w/index.php?curid=647033

The setting

Neighborhood of $\bar{z} \in Z$,
complete **geodesic** metric space

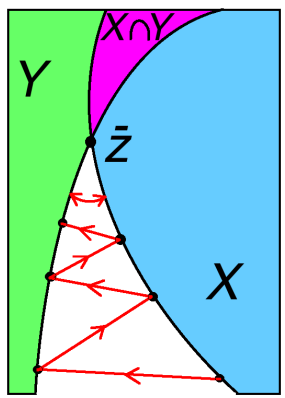
(those where points connect via
arcs isometric to compact intervals).

Eg: **manifolds** $Z = F^{-1}(0)$
for smooth $F: \mathbf{R}^n \to \mathbf{R}^m$ with
$F(\bar{z}) = 0$ and $\nabla F(\bar{z})$ onto.



Consider von Neumann's
alternating projection iteration
for closed subsets X, Y
near a common point $\bar{z}$.

Does it converge? How fast?



1

[1] commons.wikimedia.org/w/index.php?curid=647033

Local linear convergence of alternating projections

Local linear convergence of alternating projections

Near $\bar{z}$, iterates converge to a point in $X \cap Y$,

Local linear convergence of alternating projections

Near $\bar{z}$, iterates converge to a point in $X \cap Y$, with error $< \epsilon$ after

$$O\left(\log\left(\frac{1}{\epsilon}\right)\right) \quad \text{steps.}$$

Local linear convergence of alternating projections

Near $\bar{z}$, iterates converge to a point in $X \cap Y$, with error $< \epsilon$ after

$$O\left(\log\left(\frac{1}{\epsilon}\right)\right) \quad \text{steps.}$$

The ingredients: **curvature** and **transversality**

Local linear convergence of alternating projections

Near $\bar{z}$, iterates converge to a point in $X \cap Y$, with error $< \epsilon$ after

$$O\left(\log\left(\frac{1}{\epsilon}\right)\right) \quad \text{steps.}$$

The ingredients: **curvature** and **transversality**

- **Intrinsic** curvature of Z

Local linear convergence of alternating projections

Near $\bar{z}$, iterates converge to a point in $X \cap Y$, with error $< \epsilon$ after

$$O\left(\log\left(\frac{1}{\epsilon}\right)\right) \quad \text{steps.}$$

The ingredients: **curvature** and **transversality**

- **Intrinsic** curvature of Z (the "CAT(k) inequality" via the sides and angles of geodesic triangles).

Local linear convergence of alternating projections

Near $\bar{z}$, iterates converge to a point in $X \cap Y$, with error $< \epsilon$ after

$$O\left(\log\left(\frac{1}{\epsilon}\right)\right) \quad \text{steps.}$$

The ingredients: **curvature** and **transversality**

- **Intrinsic** curvature of Z (the "CAT(k) inequality" via the sides and angles of geodesic triangles). Example: $\mathbf{R}^n$

Local linear convergence of alternating projections

Near $\bar{z}$, iterates converge to a point in $X \cap Y$, with error $< \epsilon$ after

$$O\left(\log\left(\frac{1}{\epsilon}\right)\right) \quad \text{steps.}$$

The ingredients: **curvature** and **transversality**

- **Intrinsic** curvature of Z (the "CAT(k) inequality" via the sides and angles of geodesic triangles). Example: $\mathbf{R}^n$
- **Extrinsic** curvature of X (or Y) in Z:

Local linear convergence of alternating projections

Near $\bar{z}$, iterates converge to a point in $X \cap Y$, with error $< \epsilon$ after

$$O\left(\log\left(\frac{1}{\epsilon}\right)\right) \quad \text{steps.}$$

The ingredients: **curvature** and **transversality**

- **Intrinsic** curvature of Z (the "CAT(k) inequality" via the sides and angles of geodesic triangles). Example: $\mathbf{R}^n$
- **Extrinsic** curvature of X (or Y) in Z: locally, $|\text{dist}_X - \text{dist}_Z| = O(\text{dist}^3)$

Local linear convergence of alternating projections

Near $\bar{z}$, iterates converge to a point in $X \cap Y$, with error $< \epsilon$ after

$$O\left(\log\left(\frac{1}{\epsilon}\right)\right) \quad \text{steps.}$$

The ingredients: **curvature** and **transversality**

- **Intrinsic** curvature of Z (the "CAT(k) inequality" via the sides and angles of geodesic triangles). Example: $\mathbf{R}^n$
- **Extrinsic** curvature of X (or Y) in Z: locally, $|\text{dist}_X - \text{dist}_Z| = O(\text{dist}^3)$

(superfluous if Z Euclidean Drusvyatskiy-L-Ioffe 2013)

Local linear convergence of alternating projections

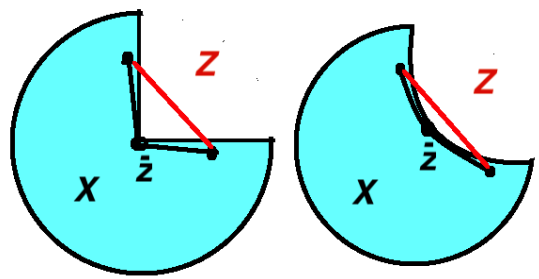
Near $\bar{z}$, iterates converge to a point in $X \cap Y$, with error $< \epsilon$ after

$$O\left(\log\left(\frac{1}{\epsilon}\right)\right) \quad \text{steps.}$$

The ingredients: **curvature** and **transversality**

- **Intrinsic** curvature of Z (the "CAT(k) inequality" via the sides and angles of geodesic triangles). Example: $\mathbf{R}^n$
- **Extrinsic** curvature of X (or Y) in Z: locally, $|\text{dist}_X - \text{dist}_Z| = O(\text{dist}^3)$

(superfluous if Z Euclidean Drusvyatskiy-L-Ioffe 2013)



Local linear convergence of alternating projections

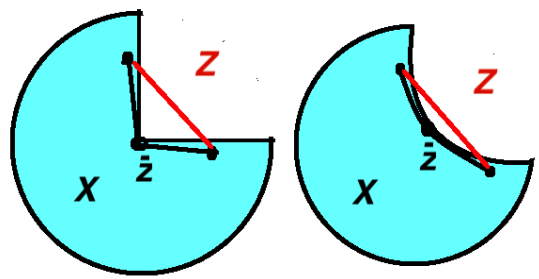
Near $\bar{z}$, iterates converge to a point in $X \cap Y$, with error $< \epsilon$ after

$$O\left(\log\left(\frac{1}{\epsilon}\right)\right) \quad \text{steps.}$$

The ingredients: **curvature** and **transversality**

- **Intrinsic** curvature of Z (the "CAT(k) inequality" via the sides and angles of geodesic triangles). Example: $\mathbf{R}^n$
- **Extrinsic** curvature of X (or Y) in Z: locally, $|\text{dist}_X - \text{dist}_Z| = O(\text{dist}^3)$

  (superfluous if Z Euclidean Drusvyatskiy-L-Ioffe 2013)



- Transversality:

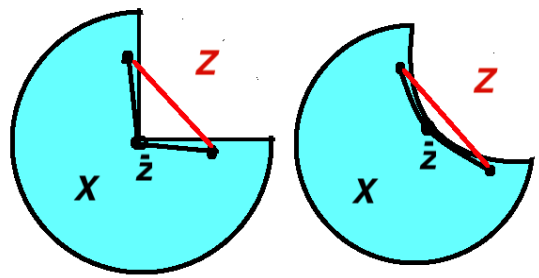
Local linear convergence of alternating projections

Near $\bar{z}$, iterates converge to a point in $X \cap Y$, with error $< \epsilon$ after

$$O\left(\log\left(\frac{1}{\epsilon}\right)\right) \quad \text{steps.}$$

The ingredients: **curvature** and **transversality**

- **Intrinsic** curvature of Z (the "CAT(k) inequality" via the sides and angles of geodesic triangles). Example: $\mathbf{R}^n$
- **Extrinsic** curvature of X (or Y) in Z: locally, $|\text{dist}_X - \text{dist}_Z| = O(\text{dist}^3)$



(superfluous if Z Euclidean Drusvyatskiy-L-Ioffe 2013)

- Transversality: no geodesic from $x \in X$ to $y \in Y$ through $\bar{z}$ has $\bar{z}$ nearest to y in X and nearest to x in Y.

Local linear convergence of alternating projections

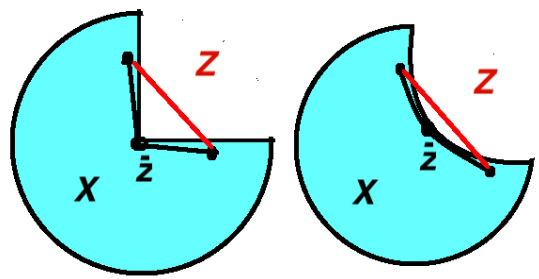
Near $\bar{z}$, iterates converge to a point in $X \cap Y$, with error $< \epsilon$ after

$$O\left(\log\left(\frac{1}{\epsilon}\right)\right) \quad \text{steps.}$$

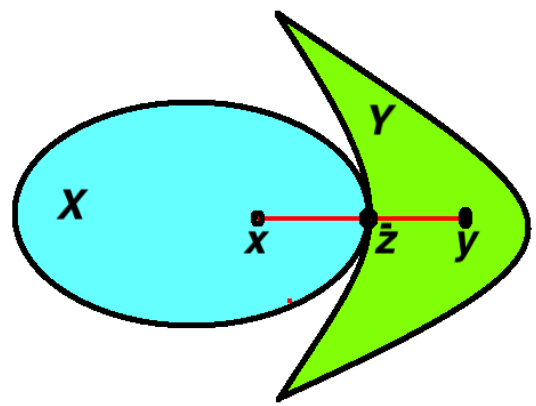
The ingredients: **curvature** and **transversality**

- **Intrinsic** curvature of Z (the "CAT(k) inequality" via the sides and angles of geodesic triangles). Example: $\mathbf{R}^n$
- **Extrinsic** curvature of X (or Y) in Z: locally, $|\text{dist}_X - \text{dist}_Z| = O(\text{dist}^3)$

(superfluous if Z Euclidean Drusvyatskiy-L-Ioffe 2013)



- Transversality: no geodesic from $x \in X$ to $y \in Y$ through $\bar{z}$ has $\bar{z}$ nearest to y in X and nearest to x in Y.



PART III

Subgradient oracles

(with Siyu Kong)

Lipschitz optimization in Euclidean space

Lipschitz optimization in Euclidean space

In Parts I and II:

- space X **general**:

Lipschitz optimization in Euclidean space

In Parts I and II:

- space X **general**: metric, geodesic ...

Lipschitz optimization in Euclidean space

In Parts I and II:

- space X **general**: metric, geodesic ...
- objective f **special**:

Lipschitz optimization in Euclidean space

In Parts I and II:

- space X **general**: metric, geodesic ...
- objective f **special**: satisfies KL

Lipschitz optimization in Euclidean space

In Parts I and II:

- space X **general**: metric, geodesic ...
- objective f **special**: satisfies KL
- a powerful **proximal**-like updating tool.

Lipschitz optimization in Euclidean space

In Parts I and II:

- space X **general**: metric, geodesic ...
- objective f **special**: satisfies KL
- a powerful **proximal**-like updating tool.

By contrast, Part III concerns

- **general** Lipschitz objectives f

Lipschitz optimization in Euclidean space

In Parts I and II:

- space X **general**: metric, geodesic ...
- objective f **special**: satisfies KL
- a powerful **proximal**-like updating tool.

By contrast, Part III concerns

- **general** Lipschitz objectives f
- but only on **Euclidean** space $\mathbf{R}^d$

Lipschitz optimization in Euclidean space

In Parts I and II:

- space X **general**: metric, geodesic ...
- objective f **special**: satisfies KL
- a powerful **proximal**-like updating tool.

By contrast, Part III concerns

- **general** Lipschitz objectives f
- but only on **Euclidean** space $\mathbf{R}^d$
- using only a **directional subgradient oracle:**

Lipschitz optimization in Euclidean space

In Parts I and II:

- space X **general**: metric, geodesic ...
- objective f **special**: satisfies KL
- a powerful **proximal**-like updating tool.

By contrast, Part III concerns

- **general** Lipschitz objectives f
- but only on **Euclidean** space $\mathbf{R}^d$
- using only a **directional subgradient oracle:**

  input: point x and direction e

Lipschitz optimization in Euclidean space

In Parts I and II:

- space X **general**: metric, geodesic ...
- objective f **special**: satisfies KL
- a powerful **proximal**-like updating tool.

By contrast, Part III concerns

- **general** Lipschitz objectives f
- but only on **Euclidean** space $\mathbf{R}^d$
- using only a **directional subgradient oracle:**

  input: point x and direction e

  output: value $f(x)$ and, correspondingly, a subgradient $g = G(x, e) \in \partial f(x)$

Lipschitz optimization in Euclidean space

In Parts I and II:

- space X **general**: metric, geodesic ...
- objective f **special**: satisfies KL
- a powerful **proximal**-like updating tool.

By contrast, Part III concerns

- **general** Lipschitz objectives f
- but only on **Euclidean** space $\mathbf{R}^d$
- using only a **directional subgradient oracle:**

  input: point x and direction e

  output: value $f(x)$ and, correspondingly,
  a subgradient $g = G(x, e) \in \partial f(x)$ such that
  $\langle g, e \rangle$ is the directional derivative $f'(x; e)$.

Subgradient oracle example

Subgradient oracle example

For some smooth functions f_i, define

$$f(x) = \max_{i=1,\ldots,k} f_i(x)$$

Subgradient oracle example

For some smooth functions f_i, define

$$f(x) = \max_{i=1,\ldots,k} f_i(x)$$

input: point x and direction e

Subgradient oracle example

For some smooth functions f_i, define

$$f(x) = \max_{i=1,\ldots,k} f_i(x)$$

input: point x and direction e

output: value $f(x)$

Subgradient oracle example

For some smooth functions f_i, define

$$f(x) = \max_{i=1,\ldots,k} f_i(x)$$

input: point x and direction e

output: value $f(x)$ and subgradient

$$G(x,e) \in \operatorname*{argmax}_{g \in G} \langle g, e \rangle$$

where $G = \{\nabla f_i(x) : i \text{ maximizes } f_i(x)\}$.

Optimizing general Lipschitz functions is hard

For L-Lipschitz f, to approximate

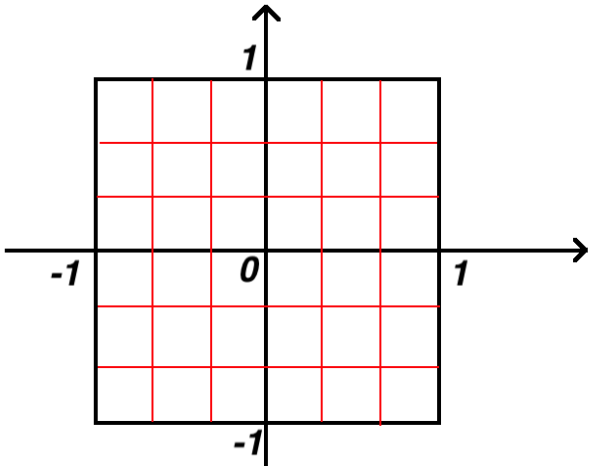
$$\min_{x \in \mathbf{R}^d} \{ f(x) : \|x\|_\infty \leq 1 \}$$

Optimizing general Lipschitz functions is hard

For L-Lipschitz f, to approximate

$$\min_{x \in \mathbf{R}^d} \{ f(x) : \|x\|_\infty \leq 1 \}$$

we must **grid-search**
j^d boxes of side $\frac{2}{j}$
to avoid error $> \frac{L}{j}$.

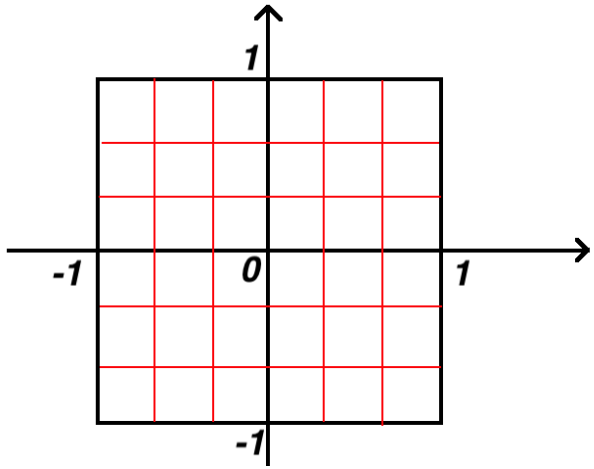


Optimizing general Lipschitz functions is hard

For L-Lipschitz f, to approximate

$$\min_{x \in \mathbf{R}^d} \{ f(x) : \|x\|_\infty \leq 1 \}$$

we must **grid-search**
j^d boxes of side $\frac{2}{j}$
to avoid error $> \frac{L}{j}$.



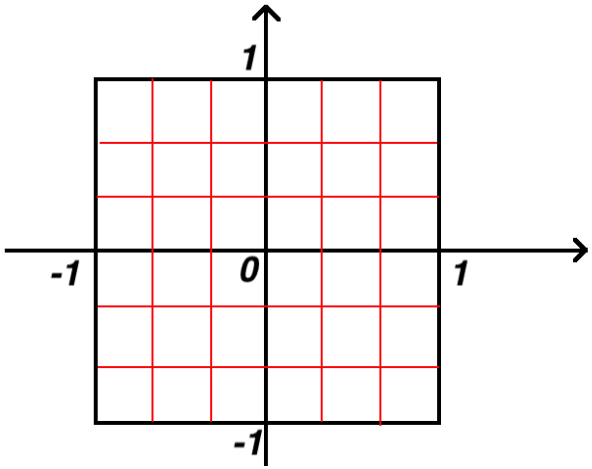
So to guarantee error $< \epsilon$ requires $(\frac{L}{\epsilon})^{\mathbf{d}}$ calls to **any** local oracle.

Optimizing general Lipschitz functions is hard

For L-Lipschitz f, to approximate

$$\min_{x \in \mathbf{R}^d} \{ f(x) : \|x\|_\infty \leq 1 \}$$

we must **grid-search**
j^d boxes of side $\frac{2}{j}$
to avoid error $> \frac{L}{j}$.



So to guarantee error $< \epsilon$ requires $(\frac{L}{\epsilon})^{\mathbf{d}}$ calls to **any** local oracle.

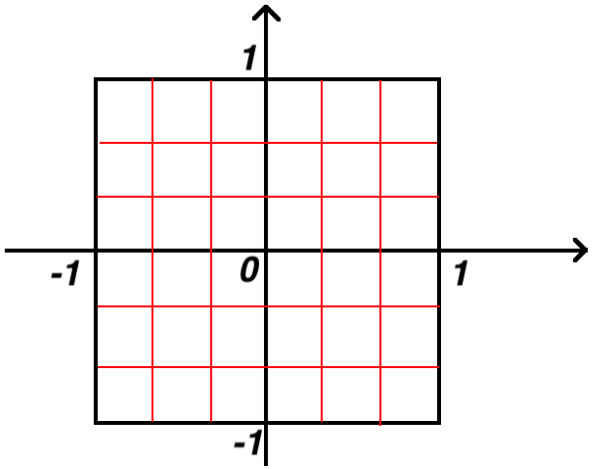
Curse of dimensionality (Yudin-Nemirovsky, 1976).

Optimizing general Lipschitz functions is hard

For L-Lipschitz f, to approximate

$$\min_{x \in \mathbf{R}^d} \{ f(x) : \|x\|_\infty \leq 1 \}$$

we must **grid-search**
j^d boxes of side $\frac{2}{j}$
to avoid error $> \frac{L}{j}$.



So to guarantee error $< \epsilon$ requires $(\frac{L}{\epsilon})^{\mathbf{d}}$ calls to **any** local oracle.

Curse of dimensionality (Yudin-Nemirovsky, 1976).

What if, instead, we aim for **small slope** $|\nabla f|(x)$?

Approximating points with small slope

For *L*-**smooth** $f \geq 0$ on $\mathbf{R}^d$, **Gradient Descent**

Approximating points with small slope

For *L*-**smooth** $f \geq 0$ on $\mathbf{R}^d$, **Gradient Descent**

```
while ‖∇f(x)‖ > ε:
        x = x − (1/L)∇f(x)
```

Approximating points with small slope

For *L*-**smooth** $f \geq 0$ on $\mathbf{R}^d$, **Gradient Descent**

```
while ‖∇f(x)‖ > ε:
        x = x − (1/L)∇f(x)
```

needs only $2Lf(x_0)\frac{1}{\epsilon^2}$ iterations.

Approximating points with small slope

For *L*-**smooth** $f \geq 0$ on $\mathbf{R}^d$, **Gradient Descent**

```
while ‖∇f(x)‖ > ε:
        x = x − (1/L)∇f(x)
```

needs only $2Lf(x_0)\frac{1}{\epsilon^2}$ iterations. **Independent of dimension** d!

Approximating points with small slope

For *L*-**smooth** $f \geq 0$ on $\mathbf{R}^d$, **Gradient Descent**

```
while ‖∇f(x)‖ > ε:
        x = x − (1/L)∇f(x)
```

needs only $2Lf(x_0)\frac{1}{\epsilon^2}$ iterations. **Independent of dimension** d!

And for **convex** (maybe nonsmooth) f, Davis-Drusvyatskiy (2018) need $\tilde{O}(\frac{1}{\epsilon^2})$ randomized subgradients (dimension-independent) to find x satisfying

$$\inf |\nabla f|\big(B_\epsilon(x)\big) \;<\; \epsilon.$$

Approximating points with small slope

For *L*-**smooth** $f \geq 0$ on $\mathbf{R}^d$, **Gradient Descent**

```
while ‖∇f(x)‖ > ε:
        x = x − (1/L)∇f(x)
```

needs only $2Lf(x_0)\frac{1}{\epsilon^2}$ iterations. **Independent of dimension** d!

And for **convex** (maybe nonsmooth) f, Davis-Drusvyatskiy (2018) need $\tilde{O}(\frac{1}{\epsilon^2})$ randomized subgradients (dimension-independent) to find x satisfying

$$\inf |\nabla f|\big(B_\epsilon(x)\big) \;<\; \epsilon.$$

But guaranteeing this accuracy if f is **neither** smooth nor convex needs $\Omega\big((\frac{1}{\epsilon})^{\mathbf{d}}\big)$ local oracle calls. Kornowski-Shamir (2022)

Convex combinations of gradients

Convex combinations of gradients

$$G_\epsilon(x) \quad = \quad \textbf{conv}\Big(\nabla f\big(B_\epsilon(x)\big)\Big) \qquad \textbf{(Goldstein subgradients}, 1977)$$

Convex combinations of gradients

$$G_\epsilon(x) = \textbf{conv}\Big(\nabla f\big(B_\epsilon(x)\big)\Big) \qquad \textbf{(Goldstein subgradients, 1977)}$$

$$\partial f(x) = \bigcap_{\epsilon>0} \mathsf{cl}\big(G_\epsilon(x)\big) \qquad \textbf{(Clarke subgradients, 1973)}$$

Convex combinations of gradients

$$G_\epsilon(x) \quad = \quad \mathbf{conv}\Big(\nabla f\big(B_\epsilon(x)\big)\Big) \qquad \text{(\textbf{Goldstein subgradients}, 1977)}$$

$$\partial f(x) \quad = \quad \bigcap_{\epsilon>0} \mathrm{cl}\big(G_\epsilon(x)\big) \qquad \text{(\textbf{Clarke subgradients}, 1973)}$$

In $\mathbf{R}^d$ we need $\Omega\big((\frac{1}{\epsilon})^{\mathbf{d}}\big)$ local oracle calls to find x satisfying

$$\inf \| \underbrace{\partial f\big(B_\epsilon(x)\big)}_{} \| \quad < \quad \epsilon \qquad \text{(KS 2022)}$$

Convex combinations of gradients

$$G_\epsilon(x) \quad = \quad \mathbf{conv}\Big(\nabla f\big(B_\epsilon(x)\big)\Big) \qquad \text{(\textbf{Goldstein subgradients}, 1977)}$$

$$\partial f(x) \quad = \quad \bigcap_{\epsilon>0} \mathrm{cl}\big(G_\epsilon(x)\big) \qquad \text{(\textbf{Clarke subgradients}, 1973)}$$

In $\mathbf{R}^d$ we need $\Omega\big((\frac{1}{\epsilon})^{\mathbf{d}}\big)$ local oracle calls to find x satisfying

$$\inf \| \underbrace{\partial f\big(B_\epsilon(x)\big)}_{\textbf{not convex!}} \| \quad < \quad \epsilon \qquad \text{(KS 2022)}$$

Convex combinations of gradients

$$G_\epsilon(x) \quad = \quad \textbf{conv}\Big(\nabla f\big(B_\epsilon(x)\big)\Big) \qquad \textbf{(Goldstein subgradients}, 1977)$$

$$\partial f(x) \quad = \quad \bigcap_{\epsilon>0} \mathrm{cl}\big(G_\epsilon(x)\big) \qquad \textbf{(Clarke subgradients}, 1973)$$

In $\mathbf{R}^d$ we need $\Omega\big((\frac{1}{\epsilon})^{\mathbf{d}}\big)$ local oracle calls to find x satisfying

$$\inf \| \underbrace{\partial f\big(B_\epsilon(x)\big)}_{\textbf{not convex!}} \| \quad < \quad \epsilon \qquad \text{(KS 2022)}$$

But Zhang-Lin-Jegelka-Sra-Jadbabaie (2020) find x satisfying

$$\inf \| \overbrace{G_\epsilon(x)}^{\textbf{convex}} \| \quad < \quad \epsilon$$

Convex combinations of gradients

$$G_\epsilon(x) \quad = \quad \textbf{conv}\Big(\nabla f\big(B_\epsilon(x)\big)\Big) \qquad \textbf{(Goldstein subgradients}, 1977)$$

$$\partial f(x) \quad = \quad \bigcap_{\epsilon>0} \mathrm{cl}\big(G_\epsilon(x)\big) \qquad \textbf{(Clarke subgradients}, 1973)$$

In $\mathbf{R}^d$ we need $\Omega\big((\frac{1}{\epsilon})^{\mathbf{d}}\big)$ local oracle calls to find x satisfying

$$\inf \| \underbrace{\partial f\big(B_\epsilon(x)\big)}_{\textbf{not convex!}} \| \quad < \quad \epsilon \qquad \text{(KS 2022)}$$

But Zhang-Lin-Jegelka-Sra-Jadbabaie (2020) find x satisfying

$$\inf \| \overbrace{G_\epsilon(x)}^{\textbf{convex}} \| \quad < \quad \epsilon$$

using just $O(\frac{1}{\epsilon^4})$ directional subgradients

Convex combinations of gradients

$$G_\epsilon(x) \quad = \quad \textbf{conv}\Big(\nabla f\big(B_\epsilon(x)\big)\Big) \qquad \textbf{(Goldstein subgradients}, 1977)$$

$$\partial f(x) \quad = \quad \bigcap_{\epsilon>0} \mathrm{cl}\big(G_\epsilon(x)\big) \qquad \textbf{(Clarke subgradients}, 1973)$$

In $\mathbf{R}^d$ we need $\Omega\big((\frac{1}{\epsilon})^{\mathbf{d}}\big)$ local oracle calls to find x satisfying

$$\inf \| \underbrace{\partial f\big(B_\epsilon(x)\big)}_{\textbf{not convex!}} \| \quad < \quad \epsilon \qquad \text{(KS 2022)}$$

But Zhang-Lin-Jegelka-Sra-Jadbabaie (2020) find x satisfying

$$\inf \| \overbrace{G_\epsilon(x)}^{\textbf{convex}} \| \quad < \quad \epsilon$$

using just $O(\frac{1}{\epsilon^4})$ directional subgradients (**independent of d!**).

Convex combinations of gradients

$$G_\epsilon(x) \quad = \quad \textbf{conv}\Big(\nabla f\big(B_\epsilon(x)\big)\Big) \qquad \textbf{(Goldstein subgradients}, 1977)$$

$$\partial f(x) \quad = \quad \bigcap_{\epsilon>0} \mathrm{cl}\big(G_\epsilon(x)\big) \qquad \textbf{(Clarke subgradients}, 1973)$$

In $\mathbf{R}^d$ we need $\Omega\big((\frac{1}{\epsilon})^{\mathbf{d}}\big)$ local oracle calls to find x satisfying

$$\inf \| \underbrace{\partial f\big(B_\epsilon(x)\big)}_{\textbf{not convex!}} \| \quad < \quad \epsilon \qquad \text{(KS 2022)}$$

But Zhang-Lin-Jegelka-Sra-Jadbabaie (2020) find x satisfying

$$\inf \| \overbrace{G_\epsilon(x)}^{\textbf{convex}} \| \quad < \quad \epsilon$$

using just $O(\frac{1}{\epsilon^4})$ directional subgradients (**independent of d!**).
And any subgradients work: Tian et al (2021), Davis et al (2022).

Framework algorithm (JLJSJ 2020)

Framework algorithm (JLJSJ 2020)

Idea: from point x, moving distance ϵ in direction $-\operatorname{argmin}\|G_\epsilon(x)\|$ would decrease f by ϵ^2. So, given a Goldstein subgradient $g\ldots$

Framework algorithm (JLJSJ 2020)

Idea: from point x, moving distance ϵ in direction $-\operatorname{argmin}\|G_\epsilon(x)\|$ would decrease f by ϵ^2. So, given a Goldstein subgradient $g\ldots$

while $\|g\| > \epsilon$ **do**

Framework algorithm (JLJSJ 2020)

Idea: from point x, moving distance ϵ in direction $-\operatorname{argmin}\|G_\epsilon(x)\|$ would decrease f by ϵ^2. So, given a Goldstein subgradient g...

while $\|g\| > \epsilon$ **do**

$u = \frac{1}{\|g\|} g \qquad y = x - \epsilon u$ **(trial subgradient step)**

Framework algorithm (JLJSJ 2020)

Idea: from point x, moving distance ϵ in direction $-\operatorname{argmin}\|G_\epsilon(x)\|$ would decrease f by ϵ^2. So, given a Goldstein subgradient g...

while $\|g\| > \epsilon$ **do**

  $u = \frac{1}{\|g\|} g \quad y = x - \epsilon u$ **(trial subgradient step)**

  if $f(x) - f(y) > \frac{1}{3}\epsilon^2$ **then**

    $x = y \quad g = G(x, 0)$ **(good decrease so accept)**

Framework algorithm (JLJSJ 2020)

Idea: from point x, moving distance ϵ in direction $-\operatorname{argmin}\|G_\epsilon(x)\|$ would decrease f by ϵ^2. So, given a Goldstein subgradient g...

while $\|g\| > \epsilon$ **do**
  $u = \frac{1}{\|g\|} g$ $y = x - \epsilon u$ **(trial subgradient step)**
  if $f(x) - f(y) > \frac{1}{3}\epsilon^2$ **then**
    $x = y$ $g = G(x, 0)$ **(good decrease so accept)**
  else
    explore between x and y: shorten subgradient

Framework algorithm (JLJSJ 2020)

Idea: from point x, moving distance ϵ in direction $-\operatorname{argmin}\|G_\epsilon(x)\|$ would decrease f by ϵ^2. So, given a Goldstein subgradient g...

while $\|g\| > \epsilon$ **do**
  $u = \frac{1}{\|g\|} g$ $y = x - \epsilon u$ **(trial subgradient step)**
  if $f(x) - f(y) > \frac{1}{3}\epsilon^2$ **then**
    $x = y$ $g = G(x, 0)$ **(good decrease so accept)**
  else
    explore between x and y: shorten subgradient
    $h(t) = f\big(x(t)\big) - \frac{1}{2}\epsilon t$ (for $x(t) = x + (t - \epsilon)u$)
    satisfies $h(\epsilon) < h(0) - \frac{1}{6}\epsilon^2$

Framework algorithm (JLJSJ 2020)

Idea: from point x, moving distance ϵ in direction $-\operatorname{argmin}\|G_\epsilon(x)\|$ would decrease f by ϵ^2. So, given a Goldstein subgradient g...

while $\|g\| > \epsilon$ **do**
  $u = \frac{1}{\|g\|} g \quad y = x - \epsilon u$ **(trial subgradient step)**
  if $f(x) - f(y) > \frac{1}{3}\epsilon^2$ **then**
    $x = y \quad g = G(x, 0)$ **(good decrease so accept)**
  else
    explore between x and y: shorten subgradient
    $h(t) = f\big(x(t)\big) - \frac{1}{2}\epsilon t$ (for $x(t) = x + (t - \epsilon)u$)
    satisfies $h(\epsilon) < h(0) - \frac{1}{6}\epsilon^2$
    oracle supplies $h'_+(t) = \langle G(x(t), u), u\rangle - \frac{\epsilon}{2}$

Framework algorithm (JLJSJ 2020)

Idea: from point x, moving distance ϵ in direction $-\operatorname{argmin}\|G_\epsilon(x)\|$ would decrease f by ϵ^2. So, given a Goldstein subgradient g...

while $\|g\| > \epsilon$ **do**
  $u = \frac{1}{\|g\|} g \quad y = x - \epsilon u$ **(trial subgradient step)**
  if $f(x) - f(y) > \frac{1}{3}\epsilon^2$ **then**
    $x = y \quad g = G(x, 0)$ **(good decrease so accept)**
  else
    explore between x and y: shorten subgradient
    $h(t) = f\big(x(t)\big) - \frac{1}{2}\epsilon t$ (for $x(t) = x + (t - \epsilon)u$)
    satisfies $h(\epsilon) < h(0) - \frac{1}{6}\epsilon^2$
    oracle supplies $h'_+(t) = \langle G(x(t), u), u\rangle - \frac{\epsilon}{2}$ so
    use a **line search** to solve: $h'_+(t) < 0 \quad 0 \le t \le \epsilon$

Framework algorithm (JLJSJ 2020)

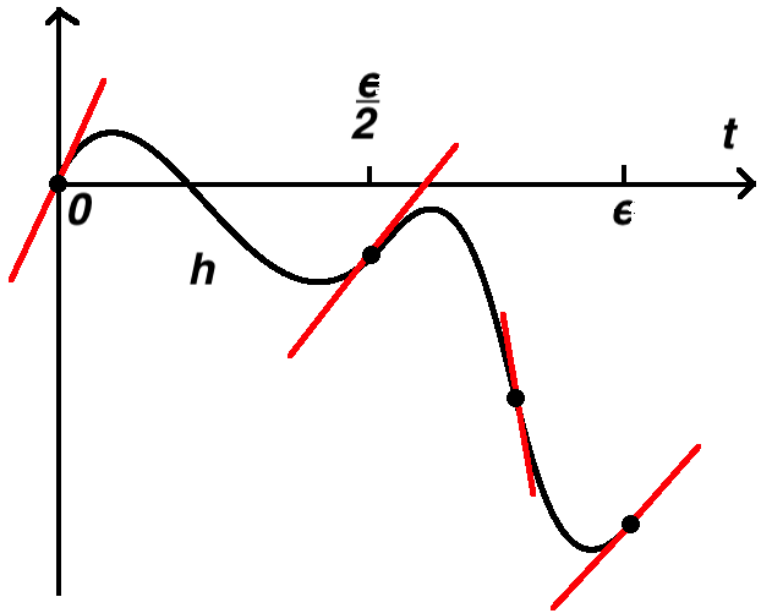
Idea: from point x, moving distance ϵ in direction $-\operatorname{argmin}\|G_\epsilon(x)\|$ would decrease f by ϵ^2. So, given a Goldstein subgradient g...

while $\|g\| > \epsilon$ **do**
  $u = \frac{1}{\|g\|} g \quad y = x - \epsilon u$ **(trial subgradient step)**
  if $f(x) - f(y) > \frac{1}{3}\epsilon^2$ **then**
    $x = y \quad g = G(x, 0)$ **(good decrease so accept)**
  else
    explore between x and y: shorten subgradient
    $h(t) = f\big(x(t)\big) - \frac{1}{2}\epsilon t$ (for $x(t) = x + (t - \epsilon)u$)
    satisfies $h(\epsilon) < h(0) - \frac{1}{6}\epsilon^2$
    oracle supplies $h'_+(t) = \langle G(x(t), u), u\rangle - \frac{\epsilon}{2}$ so
    use a **line search** to solve: $h'_+(t) < 0 \quad 0 \le t \le \epsilon$
    g = shortest vector in $[g, G(x(t), u)]$
  end if
end while

The natural line search

Given $h(\epsilon) < h(0)$, we can bisect repeatedly to solve

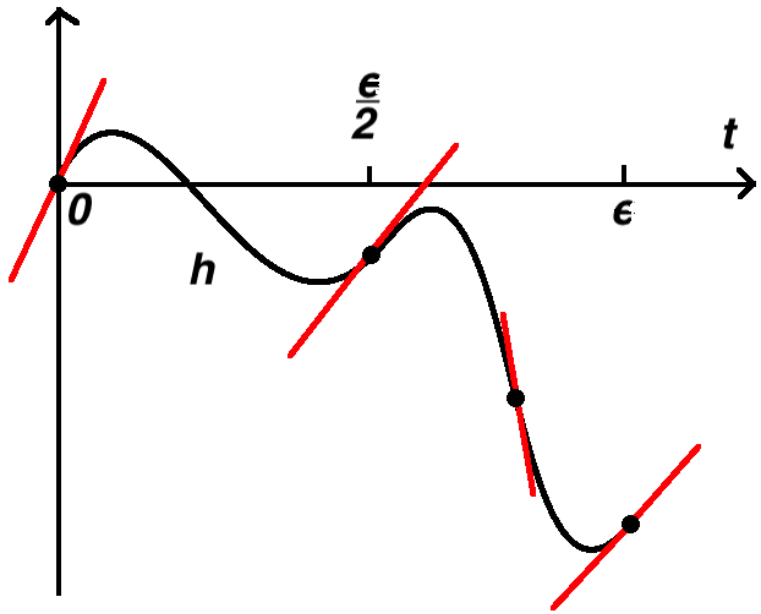
$$h'_+(t) < 0 \qquad 0 \le t \le \epsilon$$



The natural line search

Given $h(\epsilon) < h(0)$, we can bisect repeatedly to solve

$$h'_+(t) < 0 \qquad 0 \le t \le \epsilon$$

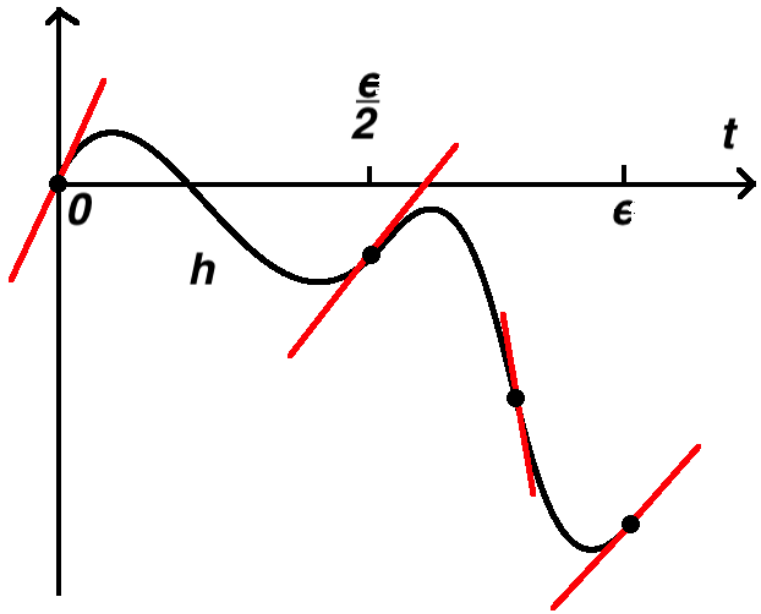


But bisection may fail on Lipschitz h with one-sided derivatives.

The natural line search

Given $h(\epsilon) < h(0)$, we can bisect repeatedly to solve

$$h'_+(t) < 0 \qquad 0 \le t \le \epsilon$$



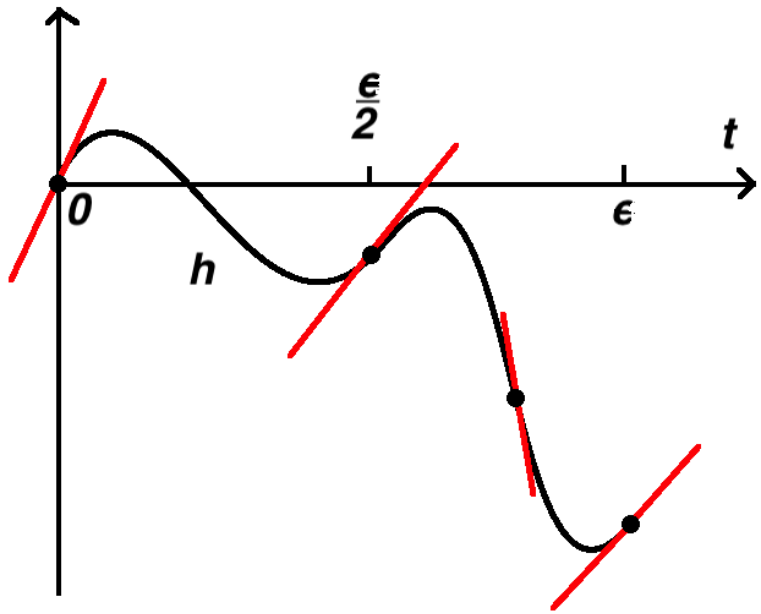
But bisection may fail on Lipschitz h with one-sided derivatives.

The JLJSJ (2020) line search:

The natural line search

Given $h(\epsilon) < h(0)$, we can bisect repeatedly to solve

$$h'_+(t) < 0 \qquad 0 \le t \le \epsilon$$



But bisection may fail on Lipschitz h with one-sided derivatives.

The JLJSJ (2020) line search: just choose **random** $t \in [0, \epsilon]$ ☺

Complexity analysis

Complexity analysis

Since f is bounded below,

Complexity analysis

Since f is bounded below, sufficient decrease

$$f(x) - f(x') \geq \frac{1}{3}\epsilon^2$$

Complexity analysis

Since f is bounded below, sufficient decrease

$$f(x) - f(x') \geq \frac{1}{3}\epsilon^2$$

holds at most $3(f(x_0) - \inf f)\frac{1}{\epsilon^2}$ times.

Complexity analysis

Since f is bounded below, sufficient decrease

$$f(x) - f(x') \;\geq\; \frac{1}{3}\epsilon^2$$

holds at most $3(f(x_0) - \inf f)\frac{1}{\epsilon^2}$ times. As long as it fails, the subgradients g satisfy $\|g\| > \epsilon$ but shrink with each trial:

Complexity analysis

Since f is bounded below, sufficient decrease

$$f(x) - f(x') \;\geq\; \frac{1}{3}\epsilon^2$$

holds at most $3(f(x_0) - \inf f)\frac{1}{\epsilon^2}$ times. As long as it fails, the subgradients g satisfy $\|g\| > \epsilon$ but shrink with each trial:

$$\frac{1}{\|g\|^2} \quad \text{always increases by at least} \quad \frac{1}{16L^2}$$

Complexity analysis

Since f is bounded below, sufficient decrease

$$f(x) - f(x') \;\geq\; \frac{1}{3}\epsilon^2$$

holds at most $3(f(x_0) - \inf f)\frac{1}{\epsilon^2}$ times. As long as it fails, the subgradients g satisfy $\|g\| > \epsilon$ but shrink with each trial:

$$\frac{1}{\|g\|^2} \quad \text{always increases by at least} \quad \frac{1}{16L^2}$$

This can occur at most $16L^2\frac{1}{\epsilon^2}$ times.

Complexity analysis

Since f is bounded below, sufficient decrease

$$f(x) - f(x') \;\geq\; \frac{1}{3}\epsilon^2$$

holds at most $3(f(x_0) - \inf f)\frac{1}{\epsilon^2}$ times. As long as it fails, the subgradients g satisfy $\|g\| > \epsilon$ but shrink with each trial:

$$\frac{1}{\|g\|^2} \quad \text{always increases by at least} \quad \frac{1}{16L^2}$$

This can occur at most $16L^2\frac{1}{\epsilon^2}$ times.

So JLJSJ (2020) need only

$$\frac{48L^2(f(x_0) - \inf f)}{\epsilon^4} \qquad \text{line searches.}$$

Complexity analysis

Since f is bounded below, sufficient decrease

$$f(x) - f(x') \;\geq\; \frac{1}{3}\epsilon^2$$

holds at most $3(f(x_0) - \inf f)\frac{1}{\epsilon^2}$ times. As long as it fails, the subgradients g satisfy $\|g\| > \epsilon$ but shrink with each trial:

$$\frac{1}{\|g\|^2} \quad \text{always increases by at least} \quad \frac{1}{16L^2}$$

This can occur at most $16L^2\frac{1}{\epsilon^2}$ times.

So JLJSJ (2020) need only

$$\frac{48L^2(f(x_0) - \inf f)}{\epsilon^4} \qquad \text{line searches.}$$

Randomizing the line search gives $O(\frac{1}{\epsilon^4})$ with high probability.

A deterministic version?

A deterministic version?

Deterministic first-order algorithms **cannot** find small Goldstein subgradients in general with dimension-independent complexity (Jordan-Lin-Zampetakis 2022).

A deterministic version?

Deterministic first-order algorithms **cannot** find small Goldstein subgradients in general with dimension-independent complexity (Jordan-Lin-Zampetakis 2022).

But for **concrete** objectives
f (and hence h),

A deterministic version?

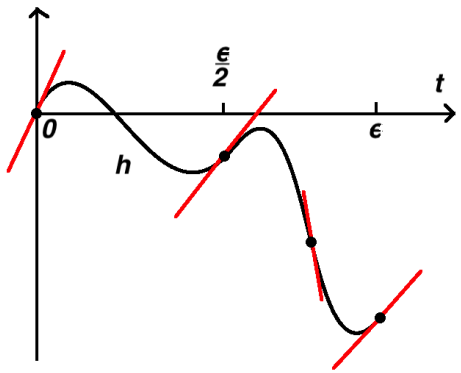
Deterministic first-order algorithms **cannot** find small Goldstein subgradients in general with dimension-independent complexity (Jordan-Lin-Zampetakis 2022).

But for **concrete** objectives
f (and hence h), if $h(\epsilon) < h(0)$
the natural bisection method **does**
solve $h'_+(t) < 0$ with $0 \le t \le \epsilon$.

A deterministic version?

Deterministic first-order algorithms **cannot** find small Goldstein subgradients in general with dimension-independent complexity (Jordan-Lin-Zampetakis 2022).

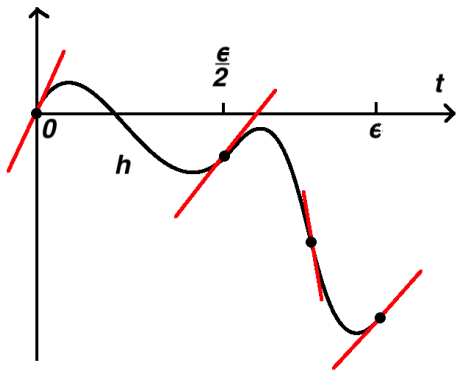
But for **concrete** objectives f (and hence h), if $h(\epsilon) < h(0)$ the natural bisection method **does** solve $h'_+(t) < 0$ with $0 \le t \le \epsilon$.



A deterministic version?

Deterministic first-order algorithms **cannot** find small Goldstein subgradients in general with dimension-independent complexity (Jordan-Lin-Zampetakis 2022).

But for **concrete** objectives f (and hence h), if $h(\epsilon) < h(0)$ the natural bisection method **does** solve $h'_+(t) < 0$ with $0 \le t \le \epsilon$.

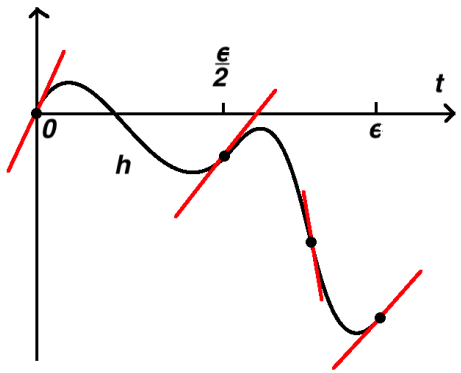


Hence a deterministic algorithm (Kong-L 2022) will terminate on concrete objectives f

A deterministic version?

Deterministic first-order algorithms **cannot** find small Goldstein subgradients in general with dimension-independent complexity (Jordan-Lin-Zampetakis 2022).

But for **concrete** objectives f (and hence h), if $h(\epsilon) < h(0)$ the natural bisection method **does** solve $h'_+(t) < 0$ with $0 \le t \le \epsilon$.

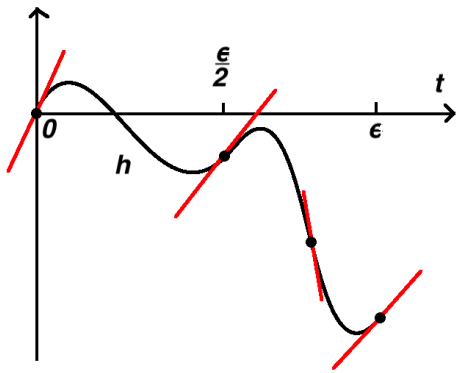


Hence a deterministic algorithm (Kong-L 2022) will terminate on concrete objectives f (in particular semi-algebraic)...

A deterministic version?

Deterministic first-order algorithms **cannot** find small Goldstein subgradients in general with dimension-independent complexity (Jordan-Lin-Zampetakis 2022).

But for **concrete** objectives f (and hence h), if $h(\epsilon) < h(0)$ the natural bisection method **does** solve $h'_+(t) < 0$ with $0 \le t \le \epsilon$.



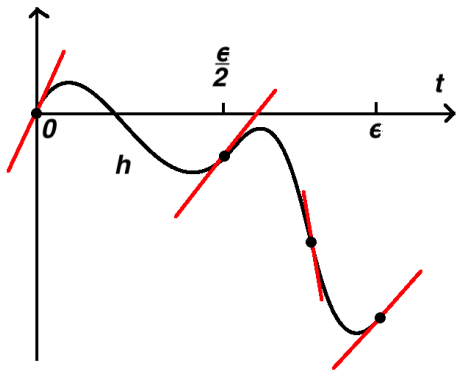
Hence a deterministic algorithm (Kong-L 2022) will terminate on concrete objectives f (in particular semi-algebraic)...

...and for difference-of-convex f,

A deterministic version?

Deterministic first-order algorithms **cannot** find small Goldstein subgradients in general with dimension-independent complexity (Jordan-Lin-Zampetakis 2022).

But for **concrete** objectives f (and hence h), if $h(\epsilon) < h(0)$ the natural bisection method **does** solve $h'_+(t) < 0$ with $0 \le t \le \epsilon$.



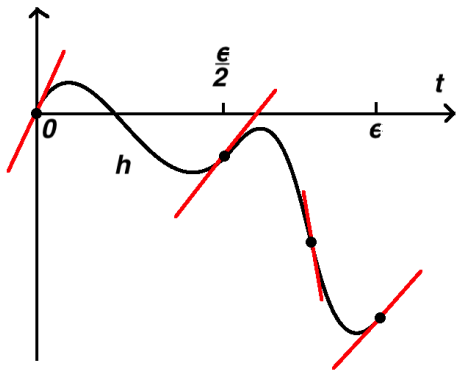
Hence a deterministic algorithm (Kong-L 2022) will terminate on concrete objectives f (in particular semi-algebraic)...

...and for difference-of-convex f, complexity is $O(\frac{1}{\epsilon^5})$,

A deterministic version?

Deterministic first-order algorithms **cannot** find small Goldstein subgradients in general with dimension-independent complexity (Jordan-Lin-Zampetakis 2022).

But for **concrete** objectives f (and hence h), if $h(\epsilon) < h(0)$ the natural bisection method **does** solve $h'_+(t) < 0$ with $0 \le t \le \epsilon$.



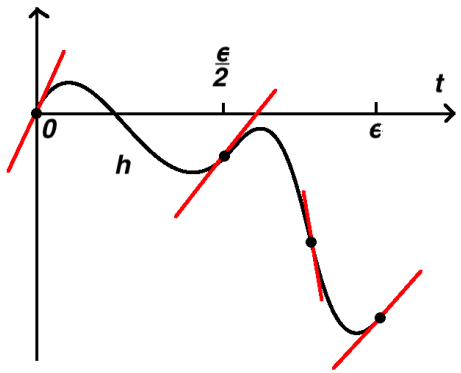
Hence a deterministic algorithm (Kong-L 2022) will terminate on concrete objectives f (in particular semi-algebraic)...

...and for difference-of-convex f, complexity is $O(\frac{1}{\epsilon^5})$, depending on **nonconvexity**,

A deterministic version?

Deterministic first-order algorithms **cannot** find small Goldstein subgradients in general with dimension-independent complexity (Jordan-Lin-Zampetakis 2022).

But for **concrete** objectives f (and hence h), if $h(\epsilon) < h(0)$ the natural bisection method **does** solve $h'_+(t) < 0$ with $0 \le t \le \epsilon$.



Hence a deterministic algorithm (Kong-L 2022) will terminate on concrete objectives f (in particular semi-algebraic)...

...and for difference-of-convex f, complexity is $O(\frac{1}{\epsilon^5})$, depending on **nonconvexity**, measured by negative part of directional second derivatives $D^2 f(x; e)$ (understood **distributionally**).

Summary: basic geometry underlying complexity

- Kurdyka-Łojasiewicz-based analysis of first-order methods is fundamentally **metric** in nature.
- One basic geometric view of alternating projections is fundamentally **geodesic** in nature.
- Dimension-independent complexity for Lipschitz optimization is fundamentally **nonconvexity**-related.