A superlinearly convergent subgradient method for sharp semismooth problems

Vasilis Charisopoulos

Joint work with Damek Davis

International Conference on Continuous Optimization, 2022

Setting

Goal: fast first-order algorithms for

$$\operatorname*{argmin}_{x \in \mathbb{R}^d} f(x) \quad \text{where} \quad \min f = 0,$$

[1] Hoffman '52.
[2] J.S. Pang '93; Ioffe '80s.

Setting

Goal: fast first-order algorithms for

$$\operatorname*{argmin}_{x \in \mathbb{R}^d} f(x) \quad \text{where} \quad \min f = 0,$$

under the following assumptions:

- f nonsmooth, locally Lipschitz.
- f satisfies classical **sharp growth** condition:

$$f(x) \geq \mu \operatorname{dist}(x, \mathcal{X}_*), \qquad \mathcal{X}_* = \operatorname{argmin} f.$$

---

[1] Hoffman '52.

[2] J.S. Pang '93; Ioffe '80s.

Setting

Goal: fast first-order algorithms for

$$\operatorname*{argmin}_{x \in \mathbb{R}^d} f(x) \quad \text{where} \quad \min f = 0,$$

under the following assumptions:

- f nonsmooth, locally Lipschitz.
- f satisfies classical **sharp growth** condition:

$$f(x) \geq \mu \operatorname{dist}(x, \mathcal{X}_*), \qquad \mathcal{X}_* = \operatorname{argmin} f.$$

Classical example: Hoffman bound for LPs / linear inequalities.[1]

$$\operatorname{dist}(x, \{x \mid Ax \leq b\}) \leq H_A \|(Ax - b)_+\|$$

[1] Hoffman '52.
[2] J.S. Pang '93; Ioffe '80s.

Setting

Goal: fast first-order algorithms for

$$\operatorname*{argmin}_{x \in \mathbb{R}^d} f(x) \quad \text{where} \quad \min f = 0,$$

under the following assumptions:

- f nonsmooth, locally Lipschitz.
- f satisfies classical **sharp growth** condition:

$$f(x) \geq \mu \operatorname{dist}(x, \mathcal{X}_*), \qquad \mathcal{X}_* = \operatorname{argmin} f.$$

Root-finding problems:

$$\text{find } \ x \ \text{ s.t. } \ F(x) = 0 \ \Leftrightarrow \ \operatorname*{argmin}_x \|F(x)\|.$$

[1] Hoffman '52.
[2] J.S. Pang '93; Ioffe '80s.

Setting

Goal: fast first-order algorithms for

$$\operatorname*{argmin}_{x \in \mathbb{R}^d} f(x) \quad \text{where} \quad \min f = 0,$$

under the following assumptions:

- f nonsmooth, locally Lipschitz.
- f satisfies classical **sharp growth** condition:

$$f(x) \geq \mu \operatorname{dist}(x, \mathcal{X}_*), \qquad \mathcal{X}_* = \operatorname{argmin} f.$$

Root-finding problems:

$$\text{find } \; x \; \text{ s.t. } \; F(x) = 0 \; \Leftrightarrow \; \operatorname*{argmin}_{x} \|F(x)\|.$$

Growth condition known as *metric subregularity*:[2]

$$\|F(x)\| \geq \mu \operatorname{dist}(x, \mathcal{X}_*).$$

---

[1] Hoffman '52.

[2] J.S. Pang '93; Ioffe '80s.

Setting

Goal: fast first-order algorithms for

$$\operatorname*{argmin}_{x \in \mathbb{R}^d} f(x) \quad \text{where} \quad \min f = 0,$$

under the following assumptions:

- f nonsmooth, locally Lipschitz.
- f satisfies classical **sharp growth** condition:

$$f(x) \geq \mu \operatorname{dist}(x, \mathcal{X}_*), \qquad \mathcal{X}_* = \operatorname{argmin} f.$$

Set intersection problems: given $\mathcal{X}_1, \mathcal{X}_2$ closed,

$$\text{find } \bar{x} \in \mathcal{X}_1 \cap \mathcal{X}_2 \;\Leftrightarrow\; \operatorname*{argmin}_{x} \left\{ \operatorname{dist}(x, \mathcal{X}_1) + \operatorname{dist}(x, \mathcal{X}_2) \right\}.$$

---

[1] Hoffman '52.

[2] J.S. Pang '93; Ioffe '80s.

Setting

Goal: fast first-order algorithms for

$$\operatorname*{argmin}_{x \in \mathbb{R}^d} f(x) \quad \text{where} \quad \min f = 0,$$

under the following assumptions:

- f nonsmooth, locally Lipschitz.
- f satisfies classical **sharp growth** condition:

$$f(x) \geq \mu \operatorname{dist}(x, \mathcal{X}_*), \qquad \mathcal{X}_* = \operatorname{argmin} f.$$

Set intersection problems: given $\mathcal{X}_1, \mathcal{X}_2$ closed,

$$\text{find } \bar{x} \in \mathcal{X}_1 \cap \mathcal{X}_2 \iff \operatorname*{argmin}_x \left\{ \operatorname{dist}(x, \mathcal{X}_1) + \operatorname{dist}(x, \mathcal{X}_2) \right\}.$$

Growth condition known as *linear regularity*:

$$\operatorname{dist}(x, \mathcal{X}_1) + \operatorname{dist}(x, \mathcal{X}_2) \geq \mu \operatorname{dist}(x, \mathcal{X}_1 \cap \mathcal{X}_2).$$

---

[1] Hoffman '52.
[2] J.S. Pang '93; Ioffe '80s.

Setting

Goal: fast first-order algorithms for

$$\operatorname*{argmin}_{x \in \mathbb{R}^d} f(x) \quad \text{where} \quad \min f = 0,$$

under the following assumptions:

- f nonsmooth, locally Lipschitz.
- f satisfies classical **sharp growth** condition:

$$f(x) \geq \mu \operatorname{dist}(x, \mathcal{X}_*), \qquad \mathcal{X}_* = \operatorname{argmin} f.$$

Fast algorithms?

---

[1] Hoffman '52.
[2] J.S. Pang '93; Ioffe '80s.

Sharp growth and rapid convergence

Subgradient method:

$$x_{k+1} := x_k - \alpha_k \frac{v_k}{\|v_k\|}, \quad v_k \in \underbrace{\partial f(x_k)}_{\text{Clarke subdifferential}}.$$

[1] Polyak '69.

[2] Davis et al. '18.

Sharp growth and rapid convergence

Subgradient method:

$$x_{k+1} := x_k - \alpha_k \frac{v_k}{\|v_k\|}, \quad v_k \in \underbrace{\partial f(x_k)}_{\text{Clarke subdifferential}}.$$

Theorem: $\{x_k\}$ converge *linearly* with *Polyak step size*:

$$\alpha_k = \frac{f(x_k)}{\|v_k\|}.$$

[1] Polyak '69.
[2] Davis et al. '18.

Sharp growth and rapid convergence

Subgradient method:

$$x_{k+1} := x_k - \alpha_k \frac{v_k}{\|v_k\|}, \quad v_k \in \underbrace{\partial f(x_k)}_{\text{Clarke subdifferential}}.$$

Theorem: $\{x_k\}$ converge *linearly* with *Polyak step size*:

$$\alpha_k = \frac{f(x_k)}{\|v_k\|}.$$

- Classically known for convex functions[1].
- Recently generalized for *weakly convex* functions[2].

Key example : $f = (\text{convex}) \circ (\text{smooth})$.

[1] Polyak '69.
[2] Davis et al. '18.

Sharp growth and rapid convergence

Subgradient method:

$$x_{k+1} := x_k - \alpha_k \frac{v_k}{\|v_k\|}, \quad v_k \in \underbrace{\partial f(x_k)}_{\text{Clarke subdifferential}}.$$

Theorem: $\{x_k\}$ converge *linearly* with *Polyak step size*:

$$\alpha_k = \frac{f(x_k)}{\|v_k\|}.$$

- Classically known for convex functions[1].
- Recently generalized for *weakly convex* functions[2].

Key example : $f = (\text{convex}) \circ (\text{smooth})$.

Question: *Faster than linear convergence using only subgradients?*

[1] Polyak '69.
[2] Davis et al. '18.

An answer in pictures

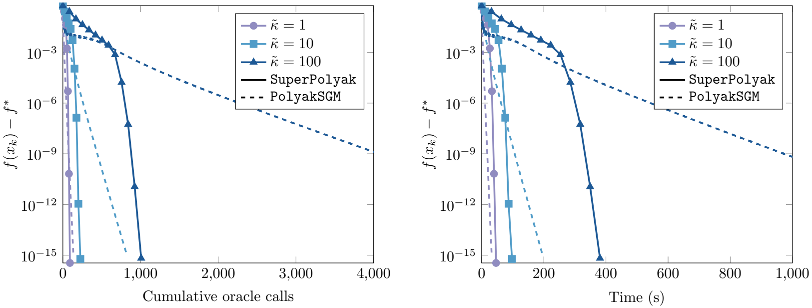


Figure: Our algorithm (`SuperPolyak`) applied to a matrix sensing problem with dimensions $(d, r) = (2^{15}, 2)$ and $m = 2^{19}$ measurements. Here, $\tilde{\kappa}$ is the condition number of the unknown matrix.

Algorithm converges superlinearly, with fewer subgradient oracle calls. **How?**

Motivation: Polyak bundle

Simplifying assumption: problem has *unique solution* $\bar{x}$.

[3] Polyak 87'.

Motivation: Polyak bundle

Simplifying assumption: problem has *unique solution* $\bar{x}$.

Polyak step equivalent to:

$$x_{k+1} := \operatorname*{argmin}_{x} \left\{ \|x - x_k\|^2 \mid f(x_k) + \langle v_k, x - x_k \rangle \leq 0 \right\}, \; v_k \in \partial f(x_k).$$

[3] Polyak 87'.

Motivation: Polyak bundle

Simplifying assumption: problem has *unique solution* $\bar{x}$.

Polyak step equivalent to:

$$x_{k+1} := \underset{x}{\operatorname{argmin}} \left\{ \|x - x_k\|^2 \mid f(x_k) + \langle v_k, x - x_k \rangle \leq 0 \right\}, \; v_k \in \partial f(x_k).$$

To improve convergence, can try to use a **bundle** $(y_i, v_i \in \partial f(y_i))$:

$$x_{k+1} := \underset{x}{\operatorname{argmin}} \left\{ \|x - x_k\|^2 \mid f(y_i) + \langle v_i, x - y_i \rangle \leq 0, \text{ for all } i \right\}.$$

- **Note**: $\bar{x}$ *always feasible* when f is convex.
- Bundle points y_i chosen among the past k iterates.
- Each step requires solving a QP.

[3] Polyak 87'.

Motivation: Polyak bundle

Simplifying assumption: problem has *unique solution* $\bar{x}$.

Polyak step equivalent to:

$$x_{k+1} := \operatorname*{argmin}_x \left\{ \|x - x_k\|^2 \mid f(x_k) + \langle v_k, x - x_k \rangle \leq 0 \right\}, \; v_k \in \partial f(x_k).$$

To improve convergence, can try to use a **bundle** $(y_i, v_i \in \partial f(y_i))$:

$$x_{k+1} := \operatorname*{argmin}_x \left\{ \|x - x_k\|^2 \mid f(y_i) + \langle v_i, x - y_i \rangle \leq 0, \text{ for all } i \right\}.$$

- **Note**: $\bar{x}$ *always feasible* when f is convex.
- Bundle points y_i chosen among the past k iterates.
- Each step requires solving a QP.

Guarantees?

- Good practical performance, but rate similar to subgradient method.[3]

[3] Polyak 87'.

Our approach

Main idea.

- Replace QP by a linear system:

$$x_{k+1} = \underset{x}{\operatorname{argmin}} \left\{ \|x - x_k\|^2 \mid f(y_i) + \langle v_i, x - y_i \rangle = 0, \text{ for all } i \right\},$$

[1] Bolte, Daniilidis & Lewis '09.

Our approach

Main idea.

- Replace QP by a linear system:

$$x_{k+1} = \operatorname*{argmin}_x \left\{ \|x - x_k\|^2 \mid f(y_i) + \langle v_i, x - y_i \rangle = 0, \text{ for all } i \right\},$$

- Choose bundle points y_i "carefully".

[1] Bolte, Daniilidis & Lewis '09.

Our approach

Main idea.

- Replace QP by a linear system:

$$x_{k+1} = \operatorname*{argmin}_x \left\{ \|x - x_k\|^2 \mid f(y_i) + \langle v_i, x - y_i \rangle = 0, \ \text{for all } i \right\},$$

- Choose bundle points y_i "carefully".

Problem admits **closed-form solution**:

$$x_{k+1} = x_k - A^\dagger \begin{bmatrix} f(y_0) + \langle v_0, x_k - y_0 \rangle \\ \vdots \\ f(y_i) + \langle v_i, x_k - y_i \rangle \end{bmatrix}, \quad \text{where } A = \begin{bmatrix} v_0^\top \\ \vdots \\ v_i^\top \end{bmatrix}.$$

[1] Bolte, Daniilidis & Lewis '09.

Our approach

Main idea.

- Replace QP by a linear system:

$$x_{k+1} = \operatorname*{argmin}_x \left\{ \|x - x_k\|^2 \mid f(y_i) + \langle v_i, x - y_i \rangle = 0, \text{ for all } i \right\},$$

- Choose bundle points y_i "carefully".

Key additional assumption: **Semismoothness.**

$$f(x) + \langle v, \bar{x} - x \rangle = o(\|x - \bar{x}\|), \quad \text{for } v \in \partial f(x) \text{ and as } x \to \bar{x}.$$

$\Rightarrow$ Implies $\bar{x}$ is *nearly feasible* for system of equations!

[1] Bolte, Daniilidis & Lewis '09.

Our approach

Main idea.

- Replace QP by a linear system:

$$x_{k+1} = \operatorname*{argmin}_x \left\{ \|x - x_k\|^2 \mid f(y_i) + \langle v_i, x - y_i \rangle = 0, \text{ for all } i \right\},$$

- Choose bundle points y_i "carefully".

Key additional assumption: **Semismoothness.**

$$f(x) + \langle v, \bar{x} - x \rangle = o(\|x - \bar{x}\|), \quad \text{for } v \in \partial f(x) \text{ and as } x \to \bar{x}.$$

$\Rightarrow$ Implies $\bar{x}$ is *nearly feasible* for system of equations!

Semismoothness is common. Satisfied by:

- Convex and smooth, and (convex) $\circ$ (smooth) functions.
- Any *semialgebraic* function.[1]

[1] Bolte, Daniilidis & Lewis '09.

Our approach

Main idea.

- Replace QP by a linear system:

$$x_{k+1} = \underset{x}{\operatorname{argmin}} \left\{ \|x - x_k\|^2 \mid f(y_i) + \langle v_i, x - y_i \rangle = 0, \text{ for all } i \right\},$$

- Choose bundle points y_i "carefully".

Key additional assumption: **Semismoothness.**

$$f(x) + \langle v, \bar{x} - x \rangle = o(\|x - \bar{x}\|), \quad \text{for } v \in \partial f(x) \text{ and as } x \to \bar{x}.$$

$\Rightarrow$ Implies $\bar{x}$ is *nearly feasible* for system of equations!

Question: How to choose the bundle points $\{y_i\}$?

[1] Bolte, Daniilidis & Lewis '09.

Algorithm `PolyakBundle`(x, τ)

$y_0 := x$; $v_0 \in \partial f(y_0)$; $A_1 = [v_0^\mathsf{T}]$.

for $i = 1, \ldots, d$ **do**

$$y_i := y_0 - A_i^\dagger \left[f(y_j) + \langle v_j, y_0 - y_j\rangle\right]_{j=0}^{i-1}; \quad A_{i+1} := \begin{bmatrix} A_i \\ v_i^\mathsf{T} \end{bmatrix} \text{ for } v_i \in \partial f(y_i)$$

Algorithm `PolyakBundle`(x, τ)

$y_0 := x$; $v_0 \in \partial f(y_0)$; $A_1 = [v_0^\mathsf{T}]$.

for $i = 1, \ldots, d$ **do**

$$y_i := y_0 - A_i^\dagger \left[f(y_j) + \langle v_j, y_0 - y_j \rangle\right]_{j=0}^{i-1}; \quad A_{i+1} := \begin{bmatrix} A_i \\ v_i^\mathsf{T} \end{bmatrix} \text{ for } v_i \in \partial f(y_i)$$

return y_s, where $s = \operatorname{argmin}_{i : \|y_i - y_0\| \le \tau f(y_0)} f(y_i)$

Algorithm `PolyakBundle`(x, τ)

$y_0 := x$; $v_0 \in \partial f(y_0)$; $A_1 = [v_0^\top]$.

for $i = 1, \ldots, d$ **do**

$$y_i := y_0 - A_i^\dagger \left[f(y_j) + \langle v_j, y_0 - y_j \rangle\right]_{j=0}^{i-1}; \quad A_{i+1} := \begin{bmatrix} A_i \\ v_i^\top \end{bmatrix} \text{ for } v_i \in \partial f(y_i)$$

return y_s, where $s = \operatorname{argmin}_{i : \|y_i - y_0\| \le \tau f(y_0)} f(y_i)$

Note: first bundle point recovers Polyak subgradient step:

$$y_1 = y_0 - (v_0^\top)^\dagger (f(y_0) + \langle v_0, y_0 - y_0 \rangle) = y_0 - \frac{f(y_0)}{\|v_0\|^2} v_0.$$

Algorithm `PolyakBundle`(x, τ)

$y_0 := x$; $v_0 \in \partial f(y_0)$; $A_1 = [v_0^\top]$.

for $i = 1, \ldots, d$ **do**

$$y_i := y_0 - A_i^\dagger \left[f(y_j) + \langle v_j, y_0 - y_j \rangle\right]_{j=0}^{i-1}; \quad A_{i+1} := \begin{bmatrix} A_i \\ v_i^\top \end{bmatrix} \text{ for } v_i \in \partial f(y_i)$$

return y_s, where $s = \operatorname{argmin}_{i : \|y_i - y_0\| \leq \tau f(y_0)} f(y_i)$

Strategy. Sharpness and semismoothness lead to "lemma of alternatives":

1. Suppose that $y_0, \ldots, y_{j-1}$ have not improved superlinearly.

Algorithm `PolyakBundle`(x, τ)

$y_0 := x$; $v_0 \in \partial f(y_0)$; $A_1 = [v_0^\top]$.

for $i = 1, \ldots, d$ **do**

$$y_i := y_0 - A_i^\dagger \left[f(y_j) + \langle v_j, y_0 - y_j \rangle\right]_{j=0}^{i-1}; \quad A_{i+1} := \begin{bmatrix} A_i \\ v_i^\top \end{bmatrix} \text{ for } v_i \in \partial f(y_i)$$

return y_s, where $s = \operatorname{argmin}_{i : \|y_i - y_0\| \leq \tau f(y_0)} f(y_i)$

Strategy. Sharpness and semismoothness lead to "lemma of alternatives":

1. Suppose that $y_0, \ldots, y_{j-1}$ have not improved superlinearly.
2. Then, *either one* of the following must hold:
   - y_j improves superlinearly upon x;
   - $\operatorname{rank}(A_j) = j + 1$.

Algorithm `PolyakBundle`(x, τ)

$y_0 := x$; $v_0 \in \partial f(y_0)$; $A_1 = [v_0^\top]$.

for $i = 1, \ldots, d$ **do**

$$y_i := y_0 - A_i^\dagger \left[f(y_j) + \langle v_j, y_0 - y_j \rangle\right]_{j=0}^{i-1}; \quad A_{i+1} := \begin{bmatrix} A_i \\ v_i^\top \end{bmatrix} \text{ for } v_i \in \partial f(y_i)$$

return y_s, where $s = \operatorname{argmin}_{i : \|y_i - y_0\| \le \tau f(y_0)} f(y_i)$

Strategy. Sharpness and semismoothness lead to "lemma of alternatives":

1. Suppose that $y_0, \ldots, y_{j-1}$ have not improved superlinearly.
2. Then, *either one* of the following must hold:
   - y_j improves superlinearly upon x;
   - $\operatorname{rank}(A_j) = j + 1$.
3. Since $A_j \in \mathbb{R}^{(j+1) \times d}$, superlinear improvement achieved within d steps.

Algorithm `PolyakBundle`(x, τ)

$y_0 := x$; $v_0 \in \partial f(y_0)$; $A_1 = [v_0^\top]$.

for $i = 1, \ldots, d$ **do**

$$y_i := y_0 - A_i^\dagger \left[f(y_j) + \langle v_j, y_0 - y_j\rangle\right]_{j=0}^{i-1}; \quad A_{i+1} := \begin{bmatrix} A_i \\ v_i^\top \end{bmatrix} \text{ for } v_i \in \partial f(y_i)$$

return y_s, where $s = \operatorname{argmin}_{i : \|y_i - y_0\| \leq \tau f(y_0)} f(y_i)$

Theorem: C. & Davis, 2022 (informal)

Assume x near $\bar{x}$ and τ is sufficiently large. Then

$$f(\texttt{PolyakBundle}(x, \tau)) = o(f(x)).$$

- ✓ Provable early termination strategies available.
- ✓ Sequence of linear systems solved incrementally (QR-based algorithm).

Algorithm `PolyakBundle`(x, τ)

$y_0 := x$; $v_0 \in \partial f(y_0)$; $A_1 = [v_0^\top]$.

for $i = 1, \ldots, d$ **do**

$$y_i := y_0 - A_i^\dagger \left[f(y_j) + \langle v_j, y_0 - y_j \rangle\right]_{j=0}^{i-1}; \quad A_{i+1} := \begin{bmatrix} A_i \\ v_i^\top \end{bmatrix} \text{ for } v_i \in \partial f(y_i)$$

return y_s, where $s = \operatorname{argmin}_{i : \|y_i - y_0\| \leq \tau f(y_0)} f(y_i)$

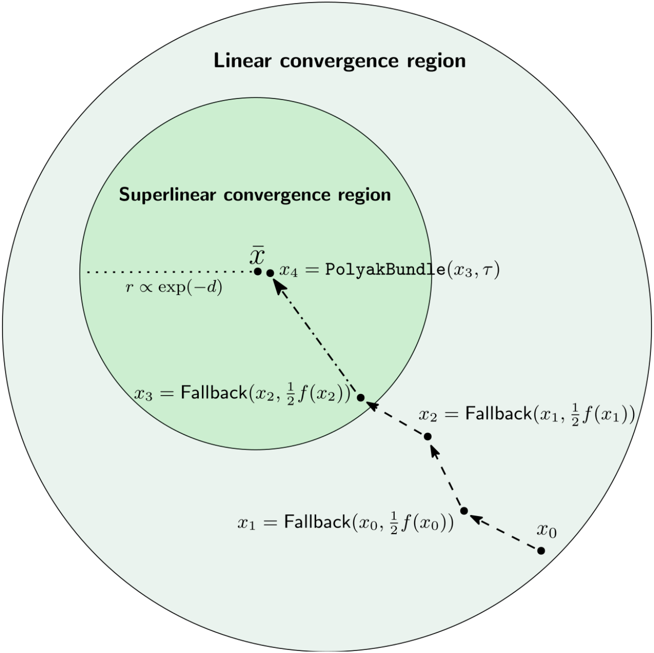
Theorem: **C.** & Davis, 2022 (informal)

Assume x near $\bar{x}$ and τ is sufficiently large. Then

$$f(\texttt{PolyakBundle}(x, \tau)) = o(f(x)).$$

- ✓ Provable early termination strategies available.
- ✓ Sequence of linear systems solved incrementally (QR-based algorithm).
- ✗ **Issue**: region of local convergence $\propto \exp(-d)$.
  - **Fix:** couple with linearly convergent method (e.g., subgradient).

High-level overview



Algorithm: `SuperPolyak`

Input: $x_0 \in \mathbb{R}^d$.

Repeat for $k = 0, 1, \ldots$

Algorithm: `SuperPolyak`

Input: $x_0 \in \mathbb{R}^d$.

Repeat for $k = 0, 1, \ldots$

Attempt a superlinear step:

$$\tilde{x} = \texttt{PolyakBundle}(x_k, (3/2)^k)$$

Algorithm: `SuperPolyak`

Input: $x_0 \in \mathbb{R}^d$.

Repeat for $k = 0, 1, \ldots$

Attempt a superlinear step:

$$\tilde{x} = \texttt{PolyakBundle}(x_k, (3/2)^k)$$

If $\tilde{x}$ available **and** sufficient decrease:

$$f(\tilde{x}) \leq \frac{1}{2} f(x_k) \implies x_{k+1} := \tilde{x}.$$

Algorithm: `SuperPolyak`

Input: $x_0 \in \mathbb{R}^d$.

Repeat for $k = 0, 1, \ldots$

Attempt a superlinear step:

$$\tilde{x} = \texttt{PolyakBundle}(x_k, (3/2)^k)$$

If $\tilde{x}$ available **and** sufficient decrease:

$$f(\tilde{x}) \leq \frac{1}{2} f(x_k) \implies x_{k+1} := \tilde{x}.$$

Else: run `Fallback` method from x_k until objective halved.

$$x_{k+1} := \texttt{Fallback}\Big(x_k, \frac{1}{2} f(x_k)\Big).$$

Algorithm: `SuperPolyak`

Input: $x_0 \in \mathbb{R}^d$.

Repeat for $k = 0, 1, \ldots$

Attempt a superlinear step:

$$\tilde{x} = \texttt{PolyakBundle}(x_k, (3/2)^k)$$

If $\tilde{x}$ available **and** sufficient decrease:

$$f(\tilde{x}) \leq \frac{1}{2} f(x_k) \implies x_{k+1} := \tilde{x}.$$

Else: run `Fallback` method from x_k until objective halved.

$$x_{k+1} := \texttt{Fallback}\Big(x_k, \frac{1}{2} f(x_k)\Big).$$

Question: which algorithm can we use as fallback method?

Algorithm: `SuperPolyak`

Input: $x_0 \in \mathbb{R}^d$.

Repeat for $k = 0, 1, \ldots$

Attempt a superlinear step:

$$\tilde{x} = \texttt{PolyakBundle}(x_k, (3/2)^k)$$

If $\tilde{x}$ available **and** sufficient decrease:

$$f(\tilde{x}) \leq \frac{1}{2} f(x_k) \implies x_{k+1} := \tilde{x}.$$

Else: run `Fallback` method from x_k until objective halved.

$$x_{k+1} := \texttt{Fallback}\left(x_k, \frac{1}{2} f(x_k)\right).$$

Theorem: **C.** and Davis, 2022 (informal)

`SuperPolyak` with the Polyak subgradient method as fallback enters the region of superlinear convergence in $O(d)$ iterations, as long as x_0 is in a dimension-independent region around $\bar{x}$.

Consequences: root-finding problems

Problem: find x s.t. $F(x) = 0$.

[1] Qi & Sun, '93.

Consequences: root-finding problems

Problem: find x s.t. $F(x) = 0$.

Classical guarantees. Suppose that F satisfies:

1. **Semismoothness**: for all x near $\bar{x}$ and $A \in \partial F(x)$,

$$\|F(x) + A(\bar{x} - x)\| = o(\|x - \bar{x}\|)$$

[1] Qi & Sun, '93.

Consequences: root-finding problems

Problem: find x s.t. $F(x) = 0$.

Classical guarantees. Suppose that F satisfies:

1. **Semismoothness**: for all x near $\bar{x}$ and $A \in \partial F(x)$,

$$\|F(x) + A(\bar{x} - x)\| = o(\|x - \bar{x}\|)$$

2. **Invertibility**: Clarke Jacobian $\partial F(x)$ invertible for x near $\bar{x}$.

[1] Qi & Sun, '93.

Consequences: root-finding problems

Problem: find x s.t. $F(x) = 0$.

Classical guarantees. Suppose that F satisfies:

1. **Semismoothness**: for all x near $\bar{x}$ and $A \in \partial F(x)$,

$$\|F(x) + A(\bar{x} - x)\| = o(\|x - \bar{x}\|)$$

2. **Invertibility**: Clarke Jacobian $\partial F(x)$ invertible for x near $\bar{x}$.

Then the *semismooth Newton* method converges locally superlinearly:[1]

$$x_{k+1} := x_k - A_k^{-1} F(x_k), \quad A_k \in \partial F(x_k).$$

[1] Qi & Sun, '93.

Consequences: root-finding problems

Problem: find x s.t. $F(x) = 0$.

Classical guarantees. Suppose that F satisfies:

1. **Semismoothness**: for all x near $\bar{x}$ and $A \in \partial F(x)$,

$$\|F(x) + A(\bar{x} - x)\| = o(\|x - \bar{x}\|)$$

2. **Invertibility**: Clarke Jacobian $\partial F(x)$ invertible for x near $\bar{x}$.

Then the *semismooth Newton* method converges locally superlinearly:[1]

$$x_{k+1} := x_k - A_k^{-1} F(x_k), \quad A_k \in \partial F(x_k).$$

Question: is superlinear convergence possible *without* invertibility condition?

[1] Qi & Sun, '93.

Consequences: root-finding problems

Problem: find x s.t. $F(x) = 0$.

Assumptions. Suppose that F satisfies:

1. **Semismoothness**: for all x near $\bar{x}$ and $A(x) \in \partial F(x)$,

$$\|F(x) + A(\bar{x} - x)\| = o(\|x - \bar{x}\|)$$

2. **Metric subregularity**: for x near $\bar{x}$,

$$\|F(x)\| \geq \mu \, \|x - \bar{x}\|.$$

[1] Qi & Sun, '93.

Consequences: root-finding problems

Problem: find x s.t. $F(x) = 0$.

Assumptions. Suppose that F satisfies:

1. **Semismoothness**: for all x near $\bar{x}$ and $A(x) \in \partial F(x)$,

$$\|F(x) + A(\bar{x} - x)\| = o(\|x - \bar{x}\|)$$

2. **Metric subregularity**: for x near $\bar{x}$,

$$\|F(x)\| \geq \mu \|x - \bar{x}\|.$$

Corollary: **C.** and Davis, 2022 (informal)

Under above assumptions, `SuperPolyak` converges locally superlinearly.

Natural fallback method: fixed-point iteration

$$z_{i+1} := T(z_i), \quad \text{where} \quad T := I - F.$$

[1] Qi & Sun, '93.

Example: LASSO

Goal: recover s-sparse $x_\sharp \in \mathbb{R}^d$ from $y = Ax_\sharp \in \mathbb{R}^m$ $(m \ll d)$.

$$\underset{x \in \mathbb{R}^d}{\operatorname{argmin}} \left\{ \frac{1}{2} \|Ax - b\|^2 + \lambda \|x\|_1 \right\}$$

Example: LASSO

Goal: recover s-sparse $x_\sharp \in \mathbb{R}^d$ from $y = Ax_\sharp \in \mathbb{R}^m$ ($m \ll d$).

$$\underset{x \in \mathbb{R}^d}{\operatorname{argmin}} \left\{ \frac{1}{2} \left\| Ax - b \right\|^2 + \lambda \left\| x \right\|_1 \right\}$$

Fallback algorithm: proximal gradient (ISTA).

$$\begin{aligned} x_{k+1} &= T(x_k) := \operatorname{prox}_{\lambda \|\cdot\|_1} \left(x_k - \tau A^\mathsf{T}(Ax_k - y) \right) \\ f(x) &:= \left\| (I - T)(x) \right\|. \end{aligned}$$

Example: LASSO

Goal: recover s-sparse $x_\sharp \in \mathbb{R}^d$ from $y = Ax_\sharp \in \mathbb{R}^m$ $(m \ll d)$.

$$\operatorname*{argmin}_{x \in \mathbb{R}^d} \left\{ \frac{1}{2} \left\| Ax - b \right\|^2 + \lambda \left\| x \right\|_1 \right\}$$

Fallback algorithm: proximal gradient (ISTA).

$$x_{k+1} = T(x_k) := \operatorname{prox}_{\lambda \|\cdot\|_1} \big(x_k - \tau A^\mathsf{T}(Ax_k - y) \big)$$
$$f(x) := \left\| (I - T)(x) \right\|.$$

Note: $I - T$ metrically subregular but need not satisfy invertibility condition!

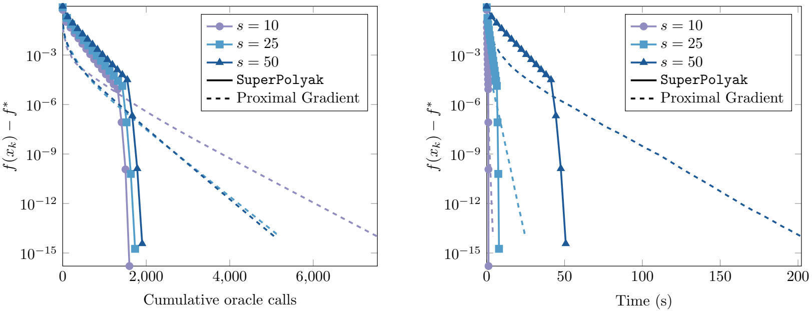
Example: LASSO

Goal: recover s-sparse $x_\sharp \in \mathbb{R}^d$ from $y = Ax_\sharp \in \mathbb{R}^m$ $(m \ll d)$.

$$\operatorname*{argmin}_{x \in \mathbb{R}^d} \left\{ \frac{1}{2} \|Ax - b\|^2 + \lambda \|x\|_1 \right\}$$

Fallback algorithm: proximal gradient (ISTA).

$$x_{k+1} = T(x_k) := \operatorname{prox}_{\lambda\|\cdot\|_1}\left(x_k - \tau A^\mathsf{T}(Ax_k - y)\right)$$
$$f(x) := \|(I - T)(x)\|.$$



Consequences: feasibility problems

Problem: find $\bar{x} \in \mathcal{X}_* = \mathcal{X}_1 \cap \mathcal{X}_2$, $\mathcal{X}_1$ and $\mathcal{X}_2$ closed.

---

[1] Drusvyatskiy '13.
[2] Lewis, Luke & Malick '09.
[3] C.H.J. Pang '15.

Consequences: feasibility problems

Problem: find $\bar{x} \in \mathcal{X}_* = \mathcal{X}_1 \cap \mathcal{X}_2$, $\mathcal{X}_1$ and $\mathcal{X}_2$ closed.

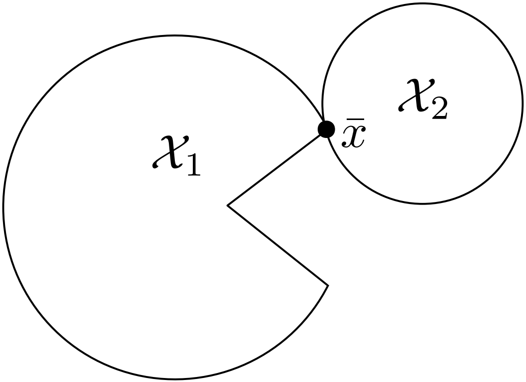


Figure: Semialgebraic sets intersecting at a single point $\bar{x}$.

---

[1] Drusvyatskiy '13.
[2] Lewis, Luke & Malick '09.
[3] C.H.J. Pang '15.

Consequences: feasibility problems

Problem: find $\bar{x} \in \mathcal{X}_* = \mathcal{X}_1 \cap \mathcal{X}_2$, $\mathcal{X}_1$ and $\mathcal{X}_2$ closed.

Setting I: intersections of semialgebraic sets.

1. The family $\{\mathcal{X}_1, \mathcal{X}_2\}$ is μ-linearly regular:

$$\mathrm{dist}(x, \mathcal{X}_1) + \mathrm{dist}(x, \mathcal{X}_2) \geq \mu \, \mathrm{dist}(x, \mathcal{X}_*) \text{ for all } x \text{ near } \bar{x}.$$

[1] Drusvyatskiy '13.
[2] Lewis, Luke & Malick '09.
[3] C.H.J. Pang '15.

Consequences: feasibility problems

Problem: find $\bar{x} \in \mathcal{X}_* = \mathcal{X}_1 \cap \mathcal{X}_2$, $\mathcal{X}_1$ and $\mathcal{X}_2$ closed.

Setting I: intersections of semialgebraic sets.

1. The family $\{\mathcal{X}_1, \mathcal{X}_2\}$ is μ-linearly regular:

$$\mathrm{dist}(x, \mathcal{X}_1) + \mathrm{dist}(x, \mathcal{X}_2) \geq \mu\,\mathrm{dist}(x, \mathcal{X}_*) \text{ for all } x \text{ near } \bar{x}.$$

2. Every $\mathcal{X}_i$ is semialgebraic and $\mathcal{X}_* = \{\bar{x}\}$.

---

[1] Drusvyatskiy '13.

[2] Lewis, Luke & Malick '09.

[3] C.H.J. Pang '15.

Consequences: feasibility problems

Problem: find $\bar{x} \in \mathcal{X}_* = \mathcal{X}_1 \cap \mathcal{X}_2$, $\mathcal{X}_1$ and $\mathcal{X}_2$ closed.

Setting I: intersections of semialgebraic sets.

1. The family $\{\mathcal{X}_1, \mathcal{X}_2\}$ is μ-linearly regular:

$$\mathrm{dist}(x, \mathcal{X}_1) + \mathrm{dist}(x, \mathcal{X}_2) \geq \mu\, \mathrm{dist}(x, \mathcal{X}_*) \text{ for all } x \text{ near } \bar{x}.$$

2. Every $\mathcal{X}_i$ is semialgebraic and $\mathcal{X}_* = \{\bar{x}\}$.

Corollary: **C.** and Davis, 2022 (informal)

Under above assumptions, `SuperPolyak` converges locally superlinearly.

Natural fallback method: alternating projections algorithm.[1]

$$z_{i+1} := \mathcal{P}_{\mathcal{X}_2}\left(\mathcal{P}_{\mathcal{X}_1}(z_i)\right).$$

[1] Drusvyatskiy '13.
[2] Lewis, Luke & Malick '09.
[3] C.H.J. Pang '15.

Consequences: feasibility problems

Problem: find $\bar{x} \in \mathcal{X}_* = \mathcal{X}_1 \cap \mathcal{X}_2$, $\mathcal{X}_1$ and $\mathcal{X}_2$ closed.

Setting II: intersections of manifolds.

1. The family $\{\mathcal{X}_1, \mathcal{X}_2\}$ is μ-linearly regular:

$$\mathrm{dist}(x, \mathcal{X}_1) + \mathrm{dist}(x, \mathcal{X}_2) \geq \mu \, \mathrm{dist}(x, \mathcal{X}_*) \text{ for all } x \text{ near } \bar{x}.$$

[1] Drusvyatskiy '13.
[2] Lewis, Luke & Malick '09.
[3] C.H.J. Pang '15.

Consequences: feasibility problems

Problem: find $\bar{x} \in \mathcal{X}_* = \mathcal{X}_1 \cap \mathcal{X}_2$, $\mathcal{X}_1$ and $\mathcal{X}_2$ closed.

Setting II: intersections of manifolds.

1. The family $\{\mathcal{X}_1, \mathcal{X}_2\}$ is μ-linearly regular:

$$\operatorname{dist}(x, \mathcal{X}_1) + \operatorname{dist}(x, \mathcal{X}_2) \geq \mu \operatorname{dist}(x, \mathcal{X}_*) \text{ for all } x \text{ near } \bar{x}.$$

2. Every $\mathcal{X}_i$ is a C^2 manifold near $\bar{x}$.

[1] Drusvyatskiy '13.
[2] Lewis, Luke & Malick '09.
[3] C.H.J. Pang '15.

Consequences: feasibility problems

Problem: find $\bar{x} \in \mathcal{X}_* = \mathcal{X}_1 \cap \mathcal{X}_2$, $\mathcal{X}_1$ and $\mathcal{X}_2$ closed.

Setting II: intersections of manifolds.

1. The family $\{\mathcal{X}_1, \mathcal{X}_2\}$ is μ-linearly regular:

$$\operatorname{dist}(x, \mathcal{X}_1) + \operatorname{dist}(x, \mathcal{X}_2) \geq \mu \operatorname{dist}(x, \mathcal{X}_*) \text{ for all } x \text{ near } \bar{x}.$$

2. Every $\mathcal{X}_i$ is a C^2 manifold near $\bar{x}$.

Corollary: **C.** and Davis, 2022 (informal)

Under above assumptions, `SuperPolyak` converges locally *quadratically*.

Natural fallback method: alternating projections algorithm.[2]

$$z_{i+1} := \mathcal{P}_{\mathcal{X}_2}\left(\mathcal{P}_{\mathcal{X}_1}(z_i)\right).$$

[1] Drusvyatskiy '13.
[2] Lewis, Luke & Malick '09.
[3] C.H.J. Pang '15.

Consequences: feasibility problems

Problem: find $\bar{x} \in \mathcal{X}_* = \mathcal{X}_1 \cap \mathcal{X}_2$, $\mathcal{X}_1$ and $\mathcal{X}_2$ closed.

Setting II: intersections of manifolds.

1. The family $\{\mathcal{X}_1, \mathcal{X}_2\}$ is μ-linearly regular:

$$\operatorname{dist}(x, \mathcal{X}_1) + \operatorname{dist}(x, \mathcal{X}_2) \geq \mu \operatorname{dist}(x, \mathcal{X}_*) \text{ for all } x \text{ near } \bar{x}.$$

2. Every $\mathcal{X}_i$ is a C^2 manifold near $\bar{x}$.

Corollary: **C.** and Davis, 2022 (informal)

Under above assumptions, `SuperPolyak` converges locally *quadratically*.

Natural fallback method: alternating projections algorithm.[2]

$$z_{i+1} := \mathcal{P}_{\mathcal{X}_2}\left(\mathcal{P}_{\mathcal{X}_1}(z_i)\right).$$

Related work: QP-based algorithm that converges under similar conditions.[3]

[1] Drusvyatskiy '13.
[2] Lewis, Luke & Malick '09.
[3] C.H.J. Pang '15.

Example: complex phase retrieval

Complex phase retrieval: given $y_\sharp \in \mathbb{R}^m$ with $(y_\sharp)_i = |\langle a_i, x_\sharp \rangle|$:

$$\text{find} \quad \hat{y} \in \mathcal{Y}_1 \cap \mathcal{Y}_2, \quad \mathcal{Y}_1 := \{u \in \mathbb{C}^m \mid |u| = y\}, \; \mathcal{Y}_2 := \mathrm{Range}(A).$$

[4] Waldspurger '18.

Example: complex phase retrieval

Complex phase retrieval: given $y_\sharp \in \mathbb{R}^m$ with $(y_\sharp)_i = |\langle a_i, x_\sharp \rangle|$:

$$\text{find} \quad \hat{y} \in \mathcal{Y}_1 \cap \mathcal{Y}_2, \quad \mathcal{Y}_1 := \{u \in \mathbb{C}^m \mid |u| = y\}, \; \mathcal{Y}_2 := \mathrm{Range}(A).$$

Fallback algorithm: alternating projections[4].

$$y_{k+1} = T(y_k) := AA^\dagger \left(y_\sharp \odot \mathrm{phase}(y_k)\right)$$

Function used by `PolyakBundle`: $f(y) := \mathrm{dist}(y, \mathcal{Y}_1) + \mathrm{dist}(y, \mathcal{Y}_2)$.

---

[4] Waldspurger '18.

Example: complex phase retrieval

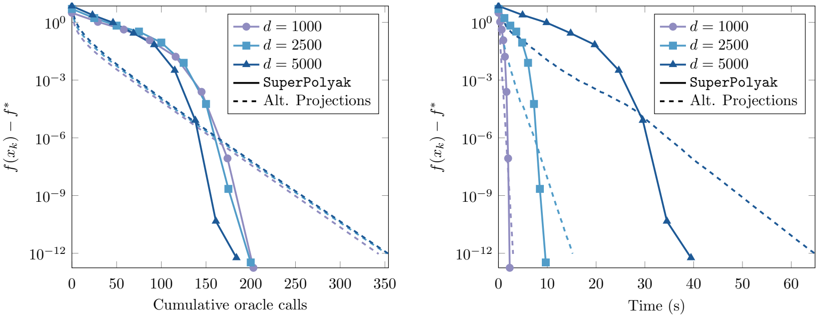
Complex phase retrieval: given $y_\sharp \in \mathbb{R}^m$ with $(y_\sharp)_i = |\langle a_i, x_\sharp \rangle|$:

$$\text{find} \quad \hat{y} \in \mathcal{Y}_1 \cap \mathcal{Y}_2, \quad \mathcal{Y}_1 := \{u \in \mathbb{C}^m \mid |u| = y\}, \; \mathcal{Y}_2 := \text{Range}(A).$$

Fallback algorithm: alternating projections[4].

$$y_{k+1} = T(y_k) := AA^\dagger \left(y_\sharp \odot \text{phase}(y_k)\right)$$

Function used by `PolyakBundle`: $f(y) := \text{dist}(y, \mathcal{Y}_1) + \text{dist}(y, \mathcal{Y}_2)$.



---

[4] Waldspurger '18.

Concluding remarks

Not covered in this talk:

- Results for non-isolated $\mathcal{X}_*$ via a uniformization of semismoothness.[5]
- Provable early termination strategies for `PolyakBundle` loop.
    - In practice, lead to small (even constant-sized) linear systems.

[5] Davis et al. '21

Concluding remarks

Not covered in this talk:

- Results for non-isolated $\mathcal{X}_*$ via a uniformization of semismoothness.[5]
- Provable early termination strategies for `PolyakBundle` loop.
  - In practice, lead to small (even constant-sized) linear systems.

Open questions:

1. Reduce dim. dependence of local convergence region of `PolyakBundle`.
2. Remove requirement that $f_* = \min f$ is known.
3. Layer on top of existing large-scale solvers (LPs? QPs?)

[5] Davis et al. '21

Concluding remarks

Not covered in this talk:

- Results for non-isolated $\mathcal{X}_*$ via a uniformization of semismoothness.[5]
- Provable early termination strategies for `PolyakBundle` loop.
  - In practice, lead to small (even constant-sized) linear systems.

Open questions:

1. Reduce dim. dependence of local convergence region of `PolyakBundle`.
2. Remove requirement that $f_* = \min f$ is known.
3. Layer on top of existing large-scale solvers (LPs? QPs?)

Thank you!

`arXiv:abs/2201.04611`

[5] Davis et al. '21

Example: low-rank matrix sensing

Bilinear sensing: recover low-rank factors $U_\sharp, V_\sharp$ from bilinear measurements:

$$y_i = \ell_i^\mathsf{T} U_\sharp V_\sharp^\mathsf{T} r_i, \quad \ell_i, r_i \overset{\text{iid}}{\sim} \mathcal{N}(0, I_d).$$

[6] **C.** et al., '21

Example: low-rank matrix sensing

Bilinear sensing: recover low-rank factors $U_\sharp, V_\sharp$ from bilinear measurements:

$$y_i = \ell_i^\top U_\sharp V_\sharp^\top r_i, \quad \ell_i, r_i \overset{\text{iid}}{\sim} \mathcal{N}(0, I_d).$$

Loss function[6]:

$$f(U,V) = \frac{1}{m} \sum_{i=1}^{m} \left| y_i - \ell_i^\top U V^\top r_i \right|.$$

Fallback algorithm: Polyak subgradient method.

[6] **C.** et al., '21

Example: low-rank matrix sensing

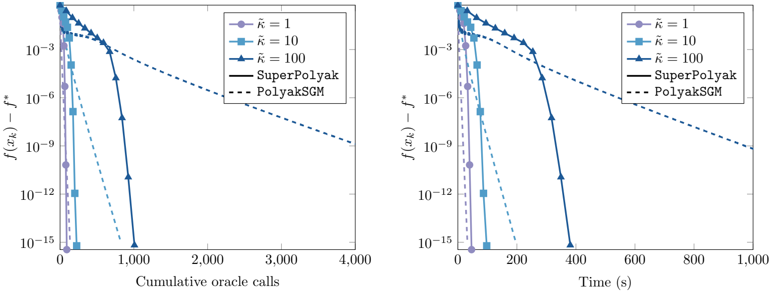
Bilinear sensing: recover low-rank factors $U_\sharp, V_\sharp$ from bilinear measurements:

$$y_i = \ell_i^\top U_\sharp V_\sharp^\top r_i, \quad \ell_i, r_i \overset{\text{iid}}{\sim} \mathcal{N}(0, I_d).$$

Loss function[6]:

$$f(U,V) = \frac{1}{m}\sum_{i=1}^{m} \left| y_i - \ell_i^\top U V^\top r_i \right|.$$

Fallback algorithm: Polyak subgradient method.



---

[6] **C.** et al., '21

Example: max-linear regression

Max-linear regression: recover unknown "slopes" $\beta_1^\sharp, \ldots, \beta_r^\sharp$ from

$$y_i = \max_{j \in [r]} \left\langle a_i, \beta_j^\sharp \right\rangle \quad \text{where} \quad a_i \sim \mathcal{N}(0, I_d).$$

Example: max-linear regression

Max-linear regression: recover unknown "slopes" $\beta_1^\sharp, \ldots, \beta_r^\sharp$ from

$$y_i = \max_{j \in [r]} \left\langle a_i, \beta_j^\sharp \right\rangle \quad \text{where} \quad a_i \sim \mathcal{N}(0, I_d).$$

Loss function:

$$f(\beta_1, \ldots, \beta_r) = \frac{1}{m} \sum_{i=1}^{m} \left| y_i - \max_{j \in [r]} \langle a_i, \beta_j \rangle \right|.$$

Fallback algorithm: Polyak subgradient method.

Example: max-linear regression

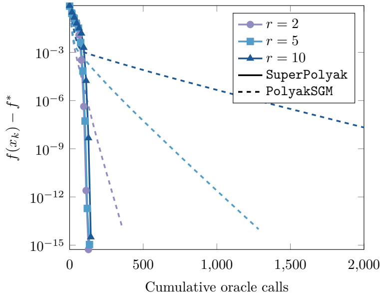
Max-linear regression: recover unknown "slopes" $\beta_1^\sharp, \ldots, \beta_r^\sharp$ from

$$y_i = \max_{j \in [r]} \langle a_i, \beta_j^\sharp \rangle \quad \text{where} \quad a_i \sim \mathcal{N}(0, I_d).$$

Loss function:

$$f(\beta_1, \ldots, \beta_r) = \frac{1}{m} \sum_{i=1}^{m} \left| y_i - \max_{j \in [r]} \langle a_i, \beta_j \rangle \right|.$$

Fallback algorithm: Polyak subgradient method.



From projection problem to linear system:

$$\begin{aligned}
&\operatorname{argmin}\left\{\|x - x_k\|^2 \mid f(y_i) + \langle v_i, x - y_i\rangle = 0\right\} \\
&= \operatorname{argmin}\left\{\|x - x_k\|^2 \mid \langle v_i, x - x_k\rangle = \langle v_i, y_i - x_k\rangle - f(y_i)\right\} \\
&= \operatorname{argmin}\left\{\|z\|^2 \mid \langle v_i, z\rangle = \langle v_i, y_i - x_k\rangle - f(y_i)\right\} \\
&= \operatorname{argmin}\left\{\|z\|^2 \mid Az + \left[f(y_i) + \langle v_i, x_k - y_i\rangle\right]_i = 0\right\}
\end{aligned}$$

Least-norm solution of $Ax + b = 0$: $x = -A^{\dagger} b$.