Expected decrease for derivative-free algorithms using random subspaces

Joint work with Clément Royer (Paris-Dauphine PSL), Warren Hare (UBC)

Lindon Roberts, University of Sydney (lindon.roberts@sydney.edu.au)

2nd Derivative-Free Optimization Symposium, University of Padova
28 June 2024

Further Reading

This talk is based on:

- L. Roberts & C. W. Royer, Direct search based on probabilistic descent in reduced spaces, *SIAM J. Optim*, 33:4 (2023).
- W. Hare, L. Roberts & C. W. Royer, Expected decrease for derivative-free algorithms using random subspaces, *arXiv:2308.04734*, 2023.

Outline

1. **Large-Scale DFO**
2. Random Subspace Methods
3. Expected Decrease Analysis

Large-Scale DFO

Interested in unconstrained nonlinear optimization

$$\min_{x \in \mathbb{R}^n} f(x),$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ is smooth but derivatives not available.

Specifically looking at the large-scale case where the ambient dimension n is large.

Large-Scale DFO

Interested in unconstrained nonlinear optimization

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x}),$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ is smooth but derivatives not available. Specifically looking at the large-scale case where the ambient dimension n is large.

Standard DFO methods are not well-suited to large-scale problems:

- Direct search: cosine measure property for poll step has explicit n dependency
- Model-based: fully linear/quadratic model accuracy properties have explicit n dependency
- Model-based: per-iteration linear algebra costs scale badly with n (e.g. $\mathcal{O}(n^3)$ for linear interpolation)

Application 1: Adversarial Example Generation [Alzantot et al., 2019]

- Find perturbations of neural network inputs which are misclassified (min. probability of correct label/max. probability of desired incorrect label)
- Neural network structure assumed to be unknown = black-box
- Want to test very few examples $\approx$ expensive
- Useful for copyright protection of artists' work against generative AI [Shan et al., 2023]

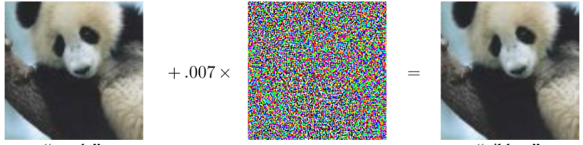


Image from [Goodfellow et al., 2015]

Application 2: Fine-Tuning Large Language Models [Malladi et al., 2023]

- Take pre-trained LLM, tweak parameters to be better at a specific task
  - e.g. Sentiment analysis: "[input text]. It was..." (good or bad?)
- Very large models = backpropagation expensive & distributed
- DFO method (MeZO) uses 12x less memory than gradient-based methods (FT) but with comparable performance

Image from [Malladi et al., 2023]

Prototypical Direct Search Method

Prototypical Direct Search Method

- Given $\boldsymbol{x}_k \in \mathbb{R}^n$ and $\Delta_k > 0$, choose a set $\mathcal{D}_k \subset \mathbb{R}^n$ of m vectors
- If there exists $\boldsymbol{d}_k \in \mathcal{D}_k$ with $f(\boldsymbol{x}_k + \Delta_k \boldsymbol{d}_k) < f(\boldsymbol{x}_k) - \frac{1}{2}\Delta_k^2 \|\boldsymbol{d}_k\|_2^2$
  - Set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta_k \boldsymbol{d}_k$ and $\Delta_{k+1} = \min(\gamma_{\text{inc}}\Delta_k, \Delta_{\max})$
  - Otherwise, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$ and $\Delta_k = \gamma_{\text{dec}}\Delta_k$

Prototypical Direct Search Method

- Given $\boldsymbol{x}_k \in \mathbb{R}^n$ and $\Delta_k > 0$, choose a set $\mathcal{D}_k \subset \mathbb{R}^n$ of m vectors
- If there exists $\boldsymbol{d}_k \in \mathcal{D}_k$ with $f(\boldsymbol{x}_k + \Delta_k \boldsymbol{d}_k) < f(\boldsymbol{x}_k) - \frac{1}{2}\Delta_k^2 \|\boldsymbol{d}_k\|_2^2$
  - Set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta_k \boldsymbol{d}_k$ and $\Delta_{k+1} = \min(\gamma_{\text{inc}}\Delta_k, \Delta_{\max})$
  - Otherwise, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$ and $\Delta_k = \gamma_{\text{dec}}\Delta_k$

For convergence, need $\mathcal{D}_k$ to be κ-descent:

$$\max_{\boldsymbol{d} \in \mathcal{D}_k} \frac{-\boldsymbol{d}^T \nabla f(\boldsymbol{x}_k)}{\|\boldsymbol{d}\|_2 \cdot \|\nabla f(\boldsymbol{x}_k)\|_2} \geq \kappa \in (0, 1]$$

i.e. there is a vector $\boldsymbol{d}$ making an acute angle with $-\nabla f(\boldsymbol{x}_k)$.

Examples: $\{\pm\boldsymbol{e}_1, \ldots, \pm\boldsymbol{e}_n\}$ with $\kappa = 1/\sqrt{n}$ or $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n, -\boldsymbol{e}\}$ with $\kappa \sim 1/n$.

[Kolda, Lewis & Torczon, 2003; Conn, Scheinberg & Vicente, 2009]

Complexity Theory

Analyze methods using **worst-case complexity**: how long before $\|\nabla f(\boldsymbol{x}_k)\|_2 \leq \epsilon$?

Complexity Theory

Analyze methods using **worst-case complexity**: how long before $\|\nabla f(\boldsymbol{x}_k)\|_2 \leq \epsilon$?

Theorem (Vicente, 2013)

If f sufficiently smooth and bounded below, then we find $\boldsymbol{x}_k$ with $\|\nabla f(\boldsymbol{x}_k)\|_2 \leq \epsilon$ after at most $\mathcal{O}(m\kappa^{-2}\epsilon^{-2})$ evaluations of f.

If $\mathcal{D}_k = \{\pm\boldsymbol{e}_1, \ldots, \pm\boldsymbol{e}_n\}$, this becomes $\mathcal{O}(n^2\epsilon^{-2})$.

Complexity Theory

Analyze methods using **worst-case complexity**: how long before $\|\nabla f(\boldsymbol{x}_k)\|_2 \leq \epsilon$?

Theorem (Vicente, 2013)

If f sufficiently smooth and bounded below, then we find $\boldsymbol{x}_k$ with $\|\nabla f(\boldsymbol{x}_k)\|_2 \leq \epsilon$ after at most $\mathcal{O}(m\kappa^{-2}\epsilon^{-2})$ evaluations of f.

If $\mathcal{D}_k = \{\pm \boldsymbol{e}_1, \ldots, \pm \boldsymbol{e}_n\}$, this becomes $\mathcal{O}(n^2\epsilon^{-2})$.

The dependency on n can (only) be reduced via randomization.

Theorem (Gratton et al., 2015)

If $\mathcal{D}_k$ is formed by taking $m \geq 2$ uniformly random unit vectors, then $\mathcal{O}(n\epsilon^{-2})$ function evaluations are required with probability at least $1 - \mathcal{O}(e^{-c\epsilon^{-2}})$.

Analyze methods using **worst-case complexity**: how long before $\|\nabla f(\boldsymbol{x}_k)\|_2 \leq \epsilon$?

Theorem (Vicente, 2013)

If f sufficiently smooth and bounded below, then we find $\boldsymbol{x}_k$ with $\|\nabla f(\boldsymbol{x}_k)\|_2 \leq \epsilon$ after at most $\mathcal{O}(m\kappa^{-2}\epsilon^{-2})$ evaluations of f.

If $\mathcal{D}_k = \{\pm \boldsymbol{e}_1, \ldots, \pm \boldsymbol{e}_n\}$, this becomes $\mathcal{O}(n^2\epsilon^{-2})$.

The dependency on n can (only) be reduced via randomization.

Theorem (Gratton et al., 2015)

If $\mathcal{D}_k$ is formed by taking $m \geq 2$ uniformly random unit vectors, then $\mathcal{O}(n\epsilon^{-2})$ function evaluations are required with probability at least $1 - \mathcal{O}(e^{-c\epsilon^{-2}})$.

Question: Can we find a systematic way to improve scalability?

Challenge

How can DFO methods be made scalable in a systematic way?

The machine learning community typically uses gradient sampling (randomized finite differencing): take a first-order method with the approximation

$$\nabla f(\boldsymbol{x}) \approx \left[\frac{f(\boldsymbol{x}+h\boldsymbol{v})-f(\boldsymbol{x})}{h}\right]\boldsymbol{v},$$

for random $\boldsymbol{v}$ (e.g. standard Gaussian). [Ghadimi & Lan, 2013; Nesterov & Spokoiny, 2017]

Randomised methods

Challenge

How can DFO methods be made scalable in a systematic way?

The machine learning community typically uses gradient sampling (randomized finite differencing): take a first-order method with the approximation

$$\nabla f(\boldsymbol{x}) \approx \left[\frac{f(\boldsymbol{x}+h\boldsymbol{v})-f(\boldsymbol{x})}{h}\right]\boldsymbol{v},$$

for random $\boldsymbol{v}$ (e.g. standard Gaussian). [Ghadimi & Lan, 2013; Nesterov & Spokoiny, 2017]

- Get improved complexity, but still requires hyperparameter tuning
- More structure in sampling gives better gradient estimates [Berahas et al., 2022]

Outline

1. Large-Scale DFO
2. **Random Subspace Methods**
3. Expected Decrease Analysis

Lemma (Johnson-Lindenstrauss, 1984)

Suppose $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d$ *and* $\epsilon \in (0,1)$. *Let* $A \in \mathbb{R}^{p \times d}$ *be a matrix with i.i.d.* $\mathcal{N}(0, p^{-2})$ *entries and* $p = \Omega(\log(N)/\epsilon)$. *Then with high probability,*

$$(1-\epsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \leq \|A\boldsymbol{x}_i - A\boldsymbol{x}_j\|_2 \leq (1+\epsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2, \qquad \forall i, j = 1, \ldots, N.$$

Lemma (Johnson-Lindenstrauss, 1984)

Suppose $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d$ *and* $\epsilon \in (0,1)$. *Let* $A \in \mathbb{R}^{p \times d}$ *be a matrix with i.i.d.* $\mathcal{N}(0, p^{-2})$ *entries and* $p = \Omega(\log(N)/\epsilon)$. *Then with high probability,*

$$(1-\epsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \leq \|A\boldsymbol{x}_i - A\boldsymbol{x}_j\|_2 \leq (1+\epsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2, \qquad \forall i,j = 1,\ldots,N.$$

- Random projections approximately preserve distances (& inner products, norms, ...)
- Reduced dimension p depends only on # of points N, not the ambient dimension d!
- Other random constructions satisfy J-L Lemma (Haar subsampling, hashing, ...)

Subspace methods

We use a subspace method: only search in low-dimensional subspaces of $\mathbb{R}^n$

Subspace methods

We use a subspace method: only search in low-dimensional subspaces of $\mathbb{R}^n$

Subspace framework:

- Generate subspace of dimension $p \ll n$ given by $\text{col}(P_k)$ for random $P_k \in \mathbb{R}^{n \times p}$
- Choose $\mathcal{D}_k \subset \mathbb{R}^p$ which is κ-descent for $P_k^T \nabla f(\boldsymbol{x}_k) \in \mathbb{R}^p$

Subspace methods

We use a subspace method: only search in low-dimensional subspaces of $\mathbb{R}^n$

Subspace framework:

- Generate subspace of dimension $p \ll n$ given by $\text{col}(P_k)$ for random $P_k \in \mathbb{R}^{n \times p}$
- Choose $\mathcal{D}_k \subset \mathbb{R}^p$ which is κ-descent for $P_k^T \nabla f(\boldsymbol{x}_k) \in \mathbb{R}^p$

Choice of subspace: we need to make sure we search in 'good' subspaces (where there is potential to decrease f sufficiently):

$$\mathbb{P}\left[\|P_k^T \nabla f(\boldsymbol{x}_k)\|_2 \geq \alpha \|\nabla f(\boldsymbol{x}_k)\|_2\right] \geq 1 - \delta, \qquad \text{for some } \alpha > 0.$$

i.e. if there is still work to do, then we (probably) know this by only inspecting f in the subspace.

Subspace methods

We use a subspace method: only search in low-dimensional subspaces of $\mathbb{R}^n$

Subspace framework:

- Generate subspace of dimension $p \ll n$ given by $\text{col}(P_k)$ for random $P_k \in \mathbb{R}^{n \times p}$
- Choose $\mathcal{D}_k \subset \mathbb{R}^p$ which is κ-descent for $P_k^T \nabla f(\boldsymbol{x}_k) \in \mathbb{R}^p$

Choice of subspace: we need to make sure we search in 'good' subspaces (where there is potential to decrease f sufficiently):

$$\mathbb{P}\left[\|P_k^T \nabla f(\boldsymbol{x}_k)\|_2 \geq \alpha \|\nabla f(\boldsymbol{x}_k)\|_2\right] \geq 1 - \delta, \qquad \text{for some } \alpha > 0.$$

i.e. if there is still work to do, then we (probably) know this by only inspecting f in the subspace. Using J-L lemma, choose $p = \Omega(1)$ independent of n.

Subspace DFO — Complexity

Theorem (R. & Royer, 2023)

If f is sufficiently smooth and bounded below and ϵ sufficiently small, then with probability at least $1 - \mathcal{O}(e^{-c\epsilon^{-2}})$ we find $\boldsymbol{x}_k$ with $\|\nabla f(\boldsymbol{x}_k)\|_2 \leq \epsilon$ after at most $\mathcal{O}(m\kappa^{-2}\epsilon^{-2})$ evaluations of f.

Using standard κ-descent choices in the subspaces, this bound matches the $\mathcal{O}(n\epsilon^{-2})$ bounds from random direct search, but any choice of $\mathcal{D}_k$ is fine (including random unit vectors).

Subspace DFO — Complexity

Theorem (R. & Royer, 2023)

If f is sufficiently smooth and bounded below and ϵ sufficiently small, then with probability at least $1 - \mathcal{O}(e^{-c\epsilon^{-2}})$ we find $\boldsymbol{x}_k$ with $\|\nabla f(\boldsymbol{x}_k)\|_2 \leq \epsilon$ after at most $\mathcal{O}(m\kappa^{-2}\epsilon^{-2})$ evaluations of f.

Using standard κ-descent choices in the subspaces, this bound matches the $\mathcal{O}(n\epsilon^{-2})$ bounds from random direct search, but any choice of $\mathcal{D}_k$ is fine (including random unit vectors).

For example, using P_k random Gaussian and $\mathcal{D}_k = \{\pm\boldsymbol{e}_1, \ldots, \pm\boldsymbol{e}_p\}$, the evaluation complexity is $\mathcal{O}(pn\epsilon^{-2})$.

Subspace DFO — Complexity

Theorem (R. & Royer, 2023)

If f is sufficiently smooth and bounded below and ϵ sufficiently small, then with probability at least $1 - \mathcal{O}(e^{-c\epsilon^{-2}})$ we find $\boldsymbol{x}_k$ with $\|\nabla f(\boldsymbol{x}_k)\|_2 \leq \epsilon$ after at most $\mathcal{O}(m\kappa^{-2}\epsilon^{-2})$ evaluations of f.

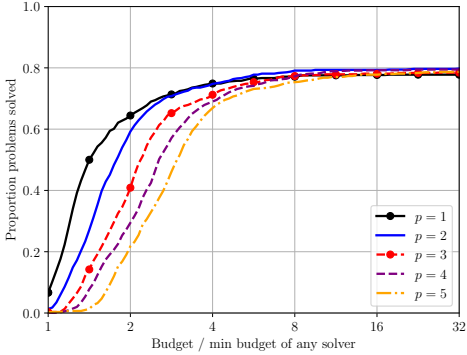
Using standard κ-descent choices in the subspaces, this bound matches the $\mathcal{O}(n\epsilon^{-2})$ bounds from random direct search, but any choice of $\mathcal{D}_k$ is fine (including random unit vectors).

For example, using P_k random Gaussian and $\mathcal{D}_k = \{\pm\boldsymbol{e}_1, \ldots, \pm\boldsymbol{e}_p\}$, the evaluation complexity is $\mathcal{O}(pn\epsilon^{-2})$.

For J-L to hold, need $p = \Omega(1)$, but unclear how to pick p in practice.

Example Results

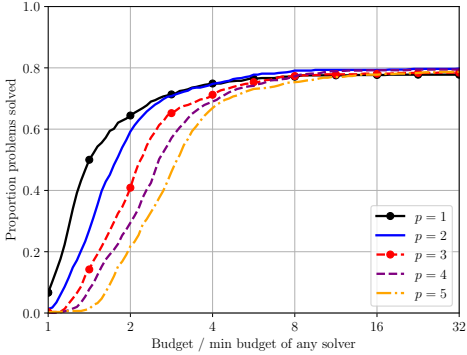
Example results for different choices of p.



Performance profiles: fraction of test problems solved vs. computational work (# evaluations of f) — higher is better.

Example Results

Example results for different choices of p.



Theory says $p = \Omega(1)$ works, numerical results say $p \to 1$ optimal. Why might this be true?

Outline

1. Large-Scale DFO
2. Random Subspace Methods
3. **Expected Decrease Analysis**

Average-Case Analysis

All the analysis above is worst-case: e.g. "for all objectives f in a given class, get $\|\nabla f(\boldsymbol{x}_k)\|_2 \leq \epsilon$ after at most $k = \mathcal{O}(\epsilon^{-2})$ iterations".

Does this capture realistic behaviour?

Average-Case Analysis

All the analysis above is worst-case: e.g. "for all objectives f in a given class, get $\|\nabla f(\boldsymbol{x}_k)\|_2 \leq \epsilon$ after at most $k = \mathcal{O}(\epsilon^{-2})$ iterations".

Does this capture realistic behaviour?

- Not for linear programming! Simplex method takes exponentially many iterations (worst-case) but on average is polynomial time [Spielman & Teng, 2004]
- Gradient descent-type methods designed for (convex) average-case Hessian spectra can outperform "worst-case optimal" methods [Pedregosa & Scieur, 2020]
- For nonconvex optimization, can do worst-case analysis in different regions of the domain separately [Curtis & Robinson, 2021]

New here: average-case analysis for nonconvex optimization algorithms.

Average-Case Analysis

What is a tractable model to analyze average-case behavior for these algorithms?

Average-Case Analysis

What is a tractable model to analyze average-case behavior for these algorithms?

- Pick random linear function $f(\boldsymbol{x}) = \boldsymbol{v}^T \boldsymbol{x}$
- At $\boldsymbol{x}_k$, pick random p-dimensional subspace
- Follow subspace direct search with $2p$ directions (i.e. $\mathcal{D}_k = \{\pm \boldsymbol{e}_1, \ldots, \pm \boldsymbol{e}_p\}$)
  - Using complete polling
- Look at expected decrease over one iteration as function of relevant dimensions

$$\mathbb{E}(p, n) := \mathbb{E}[f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1})]$$

with expectation over uniformly distributed objective functions (unit vectors $\boldsymbol{v}$) and subspaces (Stiefel manifold).

Average-Case Analysis

Assuming f is linear?

Average-Case Analysis

Assuming f is linear?

- Simplest starting model: allows us to do the relevant calculations
- Results independent of starting point $\boldsymbol{x}_k$ and scale linearly with step size Δ_k
- All steps are successful ($\boldsymbol{x}_{k+1} \neq \boldsymbol{x}_k$)
- Linear interpolation gives exact gradient (model-based)

Average-Case Analysis

Assuming f is linear?

- Simplest starting model: allows us to do the relevant calculations
- Results independent of starting point $\boldsymbol{x}_k$ and scale linearly with step size Δ_k
- All steps are successful ($\boldsymbol{x}_{k+1} \neq \boldsymbol{x}_k$)
- Linear interpolation gives exact gradient (model-based)

Alternative motivation: if ∇f is L-Lipschitz then

$$f(\boldsymbol{x}_k + \Delta_k \boldsymbol{d}_k) - f(\boldsymbol{x}_k) \leq \Delta_k \nabla f(\boldsymbol{x}_k)^T \boldsymbol{d}_k + \frac{L}{2}\Delta_k^2 \|\boldsymbol{d}_k\|^2$$

f linear $\iff L = 0$, approximately equivalent to $\Delta_k \ll 1$ (i.e. close to a solution)

Average-Case Analysis

Calculating expected decrease leads to an interesting problem:

Lemma

For direct search, $\mathbb{E}(p, n) = \mathbb{E}_{\boldsymbol{g} \sim \mathbb{S}^{n-1}}[\max(|g_1|, \ldots, |g_p|)]$

i.e. for a randomly distributed unit vector $\boldsymbol{g} \in \mathbb{R}^n$, $\|\boldsymbol{g}\|_2 = 1$, what is the expected ∞-norm of its first p coordinates?

Average-Case Analysis

Calculating expected decrease leads to an interesting problem:

Lemma

For direct search, $\mathbb{E}(p, n) = \mathbb{E}_{\boldsymbol{g} \sim \mathbb{S}^{n-1}}[\max(|g_1|, \ldots, |g_p|)]$

i.e. for a randomly distributed unit vector $\boldsymbol{g} \in \mathbb{R}^n$, $\|\boldsymbol{g}\|_2 = 1$, what is the expected ∞-norm of its first p coordinates?

Theorem (Hare, R. & Royer, 2023)

$$\mathbb{E}(p, n) = \frac{p 2^{p-1}}{\pi^{p/2}} \cdot \frac{\Gamma\left(\frac{n}{2}\right) \Gamma\left(\frac{p+1}{2}\right)}{\Gamma\left(\frac{n+1}{2}\right)} \cdot \mathcal{I}(p)$$

where $\mathcal{I}(p)$ *is a (nasty)* $(p-1)$*-dimensional integral.*

Nasty Integral

$$\mathcal{I}(p) = \int_R \left[\prod_{j=1}^{p-1} \sin^j(\varphi_j) \right] d\varphi_{p-1} \cdots d\varphi_1$$

where

$$R = \left\{ (\varphi_1, \ldots, \varphi_{p-1}) \in \left[\frac{\pi}{4}, \frac{\pi}{2}\right] \times \prod_{j=2}^{p-1} \left[\arctan\left(\prod_{k=1}^{j-1} \frac{1}{\sin(\varphi_k)} \right), \frac{\pi}{2} \right] \right\}$$

Nasty Integral

$$\mathcal{I}(p) = \int_R \left[\prod_{j=1}^{p-1} \sin^j(\varphi_j) \right] d\varphi_{p-1} \cdots d\varphi_1$$

where

$$R = \left\{ (\varphi_1, \ldots, \varphi_{p-1}) \in \left[\frac{\pi}{4}, \frac{\pi}{2}\right] \times \prod_{j=2}^{p-1} \left[\arctan\left(\prod_{k=1}^{j-1} \frac{1}{\sin(\varphi_k)} \right), \frac{\pi}{2} \right] \right\}$$

p	$\mathcal{I}(p)$
1	1
2	$1/\sqrt{2}$
3	$\left(4\arctan(\sqrt{2}) + \arctan(460\sqrt{2}/329)\right)/(8\sqrt{2})$
4	$\arctan(1/(2\sqrt{2}))/\sqrt{2}$

Average-Case Analysis

Although $\mathcal{I}(p)$ is nasty, we can still get bounds on it...

$$\mathcal{I}(p+1) < \frac{\sqrt{\pi}}{2} \frac{\Gamma\left(\frac{p+1}{2}\right)}{\Gamma\left(\frac{p}{2}+1\right)} \mathcal{I}(p) < \frac{\sqrt{\pi}}{\sqrt{2p}} \mathcal{I}(p)$$

...and then look at "expected decrease per objective evaluation"

Average-Case Analysis

Although $\mathcal{I}(p)$ is nasty, we can still get bounds on it...

$$\mathcal{I}(p+1) < \frac{\sqrt{\pi}}{2}\frac{\Gamma\left(\frac{p+1}{2}\right)}{\Gamma\left(\frac{p}{2}+1\right)}\mathcal{I}(p) < \frac{\sqrt{\pi}}{\sqrt{2p}}\mathcal{I}(p)$$

...and then look at "expected decrease per objective evaluation"

Theorem (Hare, R. & Royer, 2023)

For any n, the expected decrease per objective evaluation, $\mathbb{E}(p,n)/(2p)$, is strictly decreasing in p for $p = 1, \ldots, n$.

Average-Case Analysis

Although $\mathcal{I}(p)$ is nasty, we can still get bounds on it...

$$\mathcal{I}(p+1) < \frac{\sqrt{\pi}}{2} \frac{\Gamma\left(\frac{p+1}{2}\right)}{\Gamma\left(\frac{p}{2}+1\right)} \mathcal{I}(p) < \frac{\sqrt{\pi}}{\sqrt{2p}} \mathcal{I}(p)$$

...and then look at "expected decrease per objective evaluation"

Theorem (Hare, R. & Royer, 2023)

For any n, the expected decrease per objective evaluation, $\mathbb{E}(p,n)/(2p)$, is strictly decreasing in p for $p = 1, \ldots, n$.

So, the smallest subspace dimension $p = 1$ gives the best 'bang for your buck'. This is exactly what the numerical results said!

Algorithmic Variations

If we look at minor algorithmic variations of direct search, we get some interesting results:

- **Opportunistic polling:** if search in order $\boldsymbol{e}_1, -\boldsymbol{e}_1, \boldsymbol{e}_2, -\boldsymbol{e}_2, \ldots$ then either $\boldsymbol{e}_1$ or $-\boldsymbol{e}_1$ gives decrease, so on average try $3/2$ directions (independent of p)
- This gives better 'expected decrease per evaluation' than complete polling with any p (in particular $p = 1$)

Algorithmic Variations

If we look at minor algorithmic variations of direct search, we get some interesting results:

- **Opportunistic polling:** if search in order $\boldsymbol{e}_1, -\boldsymbol{e}_1, \boldsymbol{e}_2, -\boldsymbol{e}_2, \ldots$ then either $\boldsymbol{e}_1$ or $-\boldsymbol{e}_1$ gives decrease, so on average try $3/2$ directions (independent of p)
- This gives better 'expected decrease per evaluation' than complete polling with any p (in particular $p = 1$)
- **Parallel evaluations:** if you can do c parallel evaluations, the best choice is $p = c/2$ (i.e. smallest p where you can do all poll evaluations simultaneously)

Model-Based Methods

What about model-based methods?

Random subspace methods for model-based DFO have the same improved complexity bounds: build low-dimensional fully linear models for $\boldsymbol{s} \mapsto f(\boldsymbol{x}_k + P_k \boldsymbol{s})$. [Cartis & R., 2023]

Model-Based Methods

What about model-based methods?

Random subspace methods for model-based DFO have the same improved complexity bounds: build low-dimensional fully linear models for $\boldsymbol{s} \mapsto f(\boldsymbol{x}_k + P_k \boldsymbol{s})$. [Cartis & R., 2023]

Using linear interpolation models, the expected decrease analysis gives

Lemma

For model-based, $\mathbb{E}(p, n) = \mathbb{E}_{\boldsymbol{g} \sim \mathbb{S}^{n-1}}[\sqrt{g_1^2 + \cdots + g_p^2}]$

What about model-based methods?

Random subspace methods for model-based DFO have the same improved complexity bounds: build low-dimensional fully linear models for $\boldsymbol{s} \mapsto f(\boldsymbol{x}_k + P_k \boldsymbol{s})$. [Cartis & R., 2023]

Using linear interpolation models, the expected decrease analysis gives

Lemma

For model-based, $\mathbb{E}(p, n) = \mathbb{E}_{\boldsymbol{g} \sim \mathbb{S}^{n-1}}[\sqrt{g_1^2 + \cdots + g_p^2}]$

This is a nicer probability question than for direct search, with a nicer answer:

$$\mathbb{E}(p, n) = \frac{\Gamma\left(\frac{n}{2}\right) \cdot \Gamma\left(\frac{p+1}{2}\right)}{\Gamma\left(\frac{n+1}{2}\right) \cdot \Gamma\left(\frac{p}{2}\right)} \qquad \approx \frac{\sqrt{p}}{\sqrt{n}} \text{ for } p, n \text{ large}$$

Model-Based Methods

The main result for model-based methods (with linear interpolation models) is:

Theorem (Hare, R. & Royer, 2023)

For any n, the expected decrease per objective evaluation, $\mathbb{E}(p,n)/(p+1)$, satisfies

$$\frac{\mathbb{E}(2,n)}{3} > \left[\frac{\mathbb{E}(1,n)}{2} = \frac{\mathbb{E}(3,n)}{4}\right] > \frac{\mathbb{E}(4,n)}{5} > \cdots > \frac{\mathbb{E}(n,n)}{n+1}$$

The main result for model-based methods (with linear interpolation models) is:

Theorem (Hare, R. & Royer, 2023)

For any n, the expected decrease per objective evaluation, $\mathbb{E}(p,n)/(p+1)$, satisfies

$$\frac{\mathbb{E}(2,n)}{3} > \left[\frac{\mathbb{E}(1,n)}{2} = \frac{\mathbb{E}(3,n)}{4}\right] > \frac{\mathbb{E}(4,n)}{5} > \cdots > \frac{\mathbb{E}(n,n)}{n+1}$$

So $\mathbb{E}(p,n)/(p+1)$ is strictly decreasing in p for $p \geq 2$, not $p \geq 1$.

The main result for model-based methods (with linear interpolation models) is:

Theorem (Hare, R. & Royer, 2023)

For any n, the expected decrease per objective evaluation, $\mathbb{E}(p,n)/(p+1)$, satisfies

$$\frac{\mathbb{E}(2,n)}{3} > \left[\frac{\mathbb{E}(1,n)}{2} = \frac{\mathbb{E}(3,n)}{4}\right] > \frac{\mathbb{E}(4,n)}{5} > \cdots > \frac{\mathbb{E}(n,n)}{n+1}$$

So $\mathbb{E}(p,n)/(p+1)$ is strictly decreasing in p for $p \geq 2$, not $p \geq 1$.

(parallel evaluations: $p = c$ is best, i.e. largest p where you can do all evaluations simultaneously)

Conclusions

- Randomized projections can be effective for dimensionality reduction
- Novel average-case analysis can give fine-grained understanding of algorithm performance

Conclusions & Future Work

Conclusions

- Randomized projections can be effective for dimensionality reduction
- Novel average-case analysis can give fine-grained understanding of algorithm performance

Future Work

- Average-case analysis for quadratic objectives
- Impact of noisy objective evaluations

References i

M. Alzantot, Y. Sharma, S. Chakraborty, H. Zhang, C.-J. Hsieh, and M. B. Srivastava, *GenAttack: Practical black-box attacks with gradient-free optimization*, in Proceedings of the Genetic and Evolutionary Computation Conference, Prague, Czech Republic, 2019, ACM, pp. 1111–1119.

A. S. Berahas, L. Cao, K. Choromanski, and K. Scheinberg, *A theoretical and empirical comparison of gradient approximations in derivative-free optimization*, Foundations of Computational Mathematics, 22 (2022), pp. 507–560.

C. Cartis and L. Roberts, *Scalable subspace methods for derivative-free nonlinear least-squares optimization*, Mathematical Programming, 199 (2023), pp. 461—524.

A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, vol. 8 of MPS-SIAM Series on Optimization, MPS/SIAM, Philadelphia, 2009.

F. E. Curtis and D. P. Robinson, *Regional complexity analysis of algorithms for nonconvex smooth optimization*, Mathematical Programming, 187 (2021), pp. 579–615.

S. Ghadimi and G. Lan, *Stochastic first- and zeroth-order methods for nonconvex stochastic programming*, SIAM Journal on Optimization, 23 (2013), pp. 2341–2368.

References ii

I. J. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and harnessing adversarial examples*, in 3rd International Conference on Learning Representations ICLR, San Diego, 2015.

S. Gratton, C. W. Royer, L. N. Vicente, and Z. Zhang, *Direct search based on probabilistic descent*, SIAM Journal on Optimization, 25 (2015), pp. 1515–1541.

W. Hare, L. Roberts, and C. W. Royer, *Expected decrease for derivative-free algorithms using random subspaces*, arXiv preprint arXiv:2308.04734, (2023).

W. B. Johnson and J. Lindenstrauss, *Extensions of Lipschitz mappings into a Hilbert space*, in Contemporary Mathematics, R. Beals, A. Beck, A. Bellow, and A. Hajian, eds., vol. 26, American Mathematical Society, Providence, Rhode Island, 1984, pp. 189–206.

T. G. Kolda, R. M. Lewis, and V. Torczon, *Optimization by direct search: New perspectives on some classical and modern methods*, SIAM Review, 45 (2003), pp. 385–482.

S. Malladi, T. Gao, E. Nichani, A. Damian, J. D. Lee, D. Chen, and S. Arora, *Fine-tuning language models with just forward passes*, arXiv preprint arXiv:2305.17333, (2023).

References iii

Y. Nesterov and V. Spokoiny, *Random gradient-free minimization of convex functions*, Foundations of Computational Mathematics, 17 (2017), pp. 527–566.

F. Pedregosa and D. Scieur, *Average-case acceleration through spectral density estimation*, Proceedings of the 37th International Conference on Machine Learning, (2020).

L. Roberts and C. W. Royer, *Direct search based on probabilistic descent in reduced spaces*, SIAM Journal on Optimization, 33 (2023), pp. 3057–3082.

S. Shan, W. Ding, J. Passananti, H. Zheng, and B. Y. Zhao, *Prompt-specific poisoning attacks on text-to-image generative models*, arXiv preprint arXiv:2310.13828, (2023).

D. A. Spielman and S.-H. Teng, *Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time*, Journal of the ACM, 51 (2004), pp. 385–463.

L. N. Vicente, *Worst case complexity of direct search*, EURO Journal on Computational Optimization, 1 (2013), pp. 143–153.

Example Results (Model-Based)

Example results: model-based (linear interpolation) random subspace methods for different choices of p.