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`)

WOMBAT/WICO, University of Sydney
11 December 2023

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. **Derivative-Free Optimization**
2. Random Subspace Methods
3. New Analysis

Nonlinear Optimization

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.

- f is possibly nonconvex and/or 'black-box'
  - In practice, allow inaccurate evaluations of f, e.g. noise, outcome of iterative process
- Seek local minimizer (actually, approximate stationary point: $\|\nabla f(\boldsymbol{x})\|_2 \le \epsilon$)

Lots of high-quality algorithms available:

- Linesearch, $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k H_k^{-1}\nabla f(\boldsymbol{x}_k)$ (e.g. GD, Newton, BFGS)
- Trust-region methods (adapt well to derivative-free setting)
- Others: cubic regularization, nonlinear CG, ...

Derivative-Free Optimization

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k [\nabla^2 f(\boldsymbol{x}_k)]^{-1} \nabla f(\boldsymbol{x}_k)$$

- How to calculate derivatives of f in practice?
  - Write code by hand
  - Finite differences
  - Algorithmic differentiation/backpropagation

Derivative-Free Optimization

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k [\nabla^2 f(\boldsymbol{x}_k)]^{-1} \nabla f(\boldsymbol{x}_k)$$

- How to calculate derivatives of f in practice?
  - Write code by hand
  - Finite differences
  - Algorithmic differentiation/backpropagation
- Difficulties when function evaluation is
  - Black-box
  - Noisy
  - Computationally expensive

Derivative-Free Optimization

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k [\nabla^2 f(\boldsymbol{x}_k)]^{-1} \nabla f(\boldsymbol{x}_k)$$

- How to calculate derivatives of f in practice?
  - Write code by hand
  - Finite differences
  - Algorithmic differentiation/backpropagation
- Difficulties when function evaluation is
  - Black-box
  - Noisy
  - Computationally expensive
- Alternative — derivative-free optimization (DFO)
- Several approaches, here focus on direct search (simple & flexible)

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

- Find perturbations of neural network inputs which are misclassified
- Neural network structure assumed to be unknown (black-box!)
- Want to test very few examples ($\approx$ expensive!)

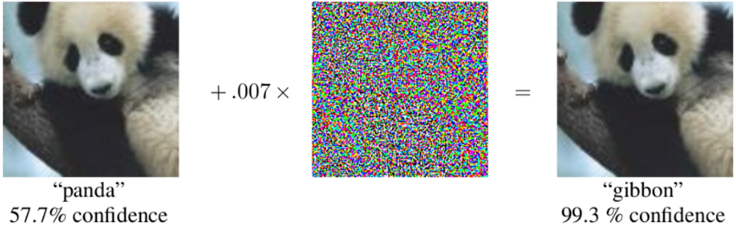


Image from [Goodfellow et al., 2015]

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 (FT; 12x more memory), DFO (MeZO) gives comparable performance

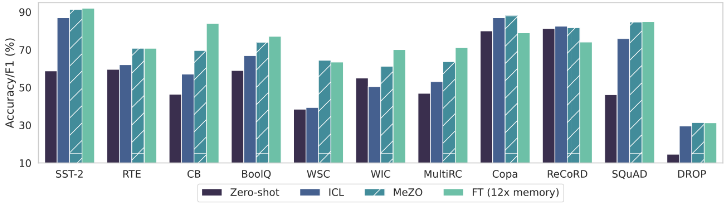


Image from [Malladi et al., 2023]

Method: Direct Search (simple & easily generalised)

Method: Direct Search (simple & easily generalised)

- 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 increase Δ_k
  - Otherwise, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$ and decrease Δ_k

Direct Search

Method: Direct Search (simple & easily generalised)

- 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 increase Δ_k
  - Otherwise, set $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$ and decrease Δ_k

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

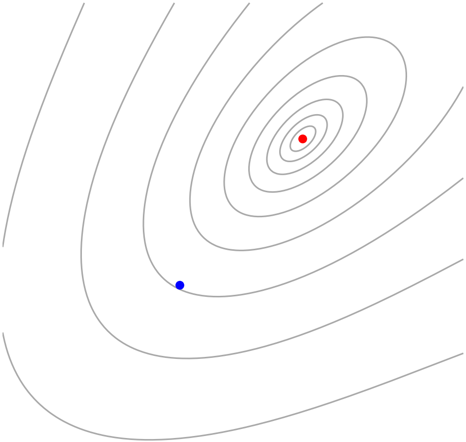
$$\max_{\boldsymbol{d} \in \mathcal{D}_k} \frac{-\boldsymbol{d}^T \nabla f(\boldsymbol{x}_k)}{\|\boldsymbol{d}\|_2 \cdot \|\nabla f(\boldsymbol{x}_k)\|_2} \geq \kappa \in (0, 1]$$

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 $\kappa \sim 1/n$.

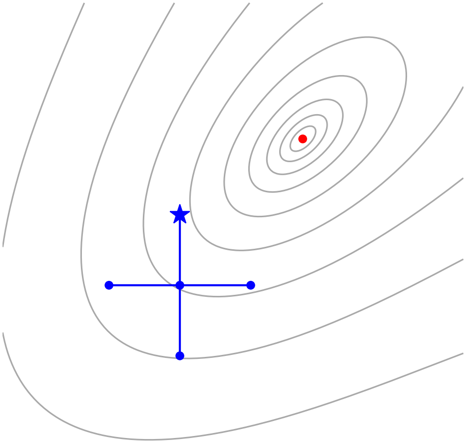
[Kolda, Lewis & Torczon, 2003; Conn, Scheinberg & Vicente, 2009]

Example: Direct Search



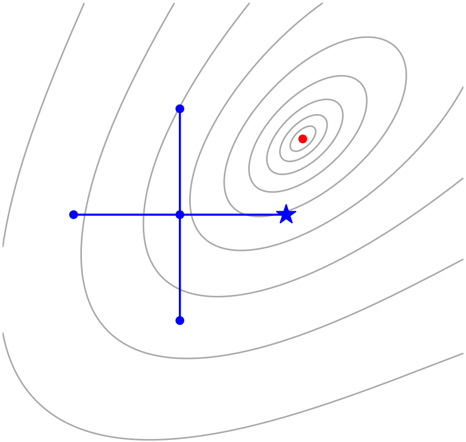
Modified from [Kolda, Lewis & Torczon, 2003]

Example: Direct Search



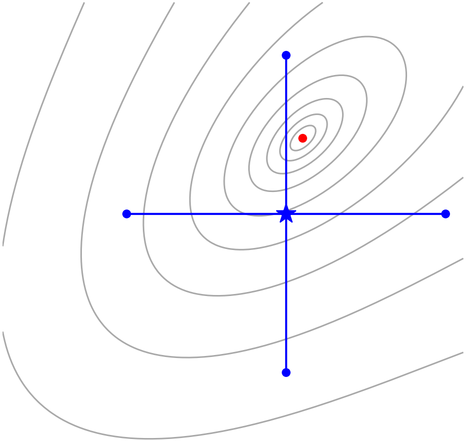
Modified from [Kolda, Lewis & Torczon, 2003]

Example: Direct Search



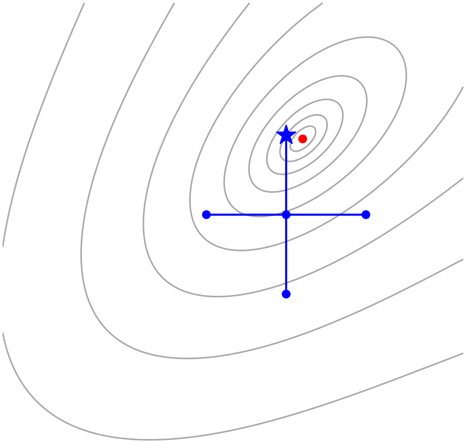
Modified from [Kolda, Lewis & Torczon, 2003]

Example: Direct Search



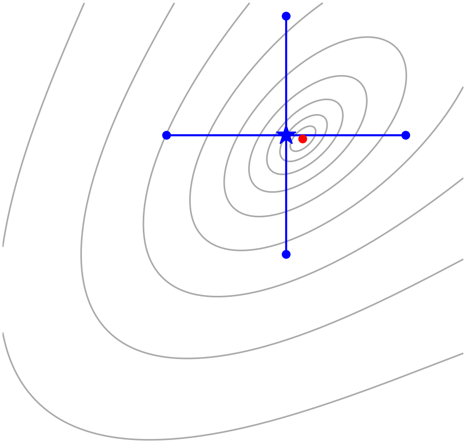
Modified from [Kolda, Lewis & Torczon, 2003]

Example: Direct Search



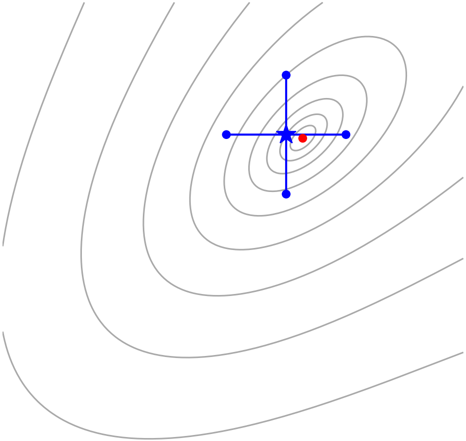
Modified from [Kolda, Lewis & Torczon, 2003]

Example: Direct Search



Modified from [Kolda, Lewis & Torczon, 2003]

Example: Direct Search



Modified from [Kolda, Lewis & Torczon, 2003]

Complexity Theory

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

Complexity Theory

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

Theorem (Vicente, 2013)

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

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

Analyse 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 randomisation.

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 $1 - \mathcal{O}(e^{-c\epsilon^{-2}})$.

Complexity Theory

Analyse 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 randomisation.

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 generate suitable random directions $\mathcal{D}_k$?

Outline

1. Derivative-Free Optimization

2. **Random Subspace Methods**

3. New Analysis

Randomisation for Dimensionality Reduction

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,*

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

Randomisation for Dimensionality Reduction

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,*

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

- 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 $\mathbb{R}^n$

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 $\mathrm{col}(P_k)$ 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 $\mathrm{col}(P_k)$ 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 $\mathrm{col}(P_k)$ 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.

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 with many ways to pick $\mathcal{D}_k$.

Subspace DFO — Complexity

Theorem (R. & Royer, 2023)

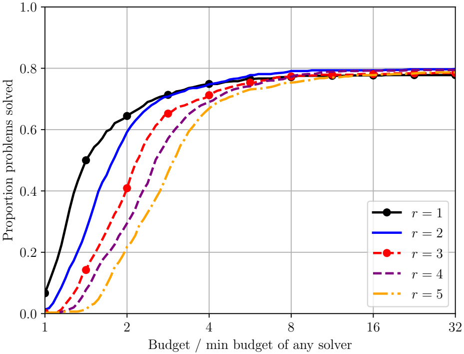
If f is sufficiently smooth and bounded below and ϵ sufficiently small, then with probability at least $1 - \mathcal{O}(e^{-c\epsilon^{-2}})$ we find $\boldsymbol{x}_k$ with $\|\nabla f(\boldsymbol{x}_k)\|_2 \leq \epsilon$ after at most $\mathcal{O}(m\kappa^{-2}\epsilon^{-2})$ evaluations of f.

Using standard κ-descent choices in the subspaces, this bound matches the $\mathcal{O}(n\epsilon^{-2})$ bounds from random direct search, but with many ways to pick $\mathcal{D}_k$.

For J-L to hold, need $p = \Omega(1)$, but unclear how small p can be.

Example Results

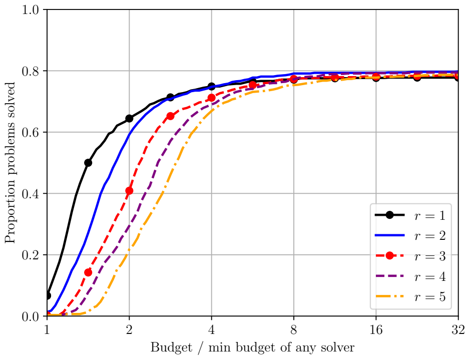
Example results: direct search for different choices of p.



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

Example Results

Example results: direct search 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. Derivative-Free Optimization
2. Random Subspace Methods
3. **New Analysis**

Average-Case Analysis

Previous analysis was worst-case (over all functions f in a smoothness class). Instead look at average-case performance.

Average-Case Analysis

Previous analysis was worst-case (over all functions f in a smoothness class). Instead look at average-case performance.

- 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. $\mathcal{D}_k = \{\pm \boldsymbol{e}_1, \ldots, \pm \boldsymbol{e}_p\}$)
- Look at expected decrease 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

Previous analysis was worst-case (over all functions f in a smoothness class). Instead look at average-case performance.

- 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. $\mathcal{D}_k = \{\pm\boldsymbol{e}_1, \ldots, \pm\boldsymbol{e}_p\}$)
- Look at expected decrease 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).

Tractable model, assumes f is linear (or $\Delta_k \ll 1$, i.e. close to a solution).

Calculating expected decrease leads to an interesting problem:

Lemma

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

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

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

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)

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

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	$\mathcal{I}(p)$	Approx.
1	1	1.0000
2	$1/\sqrt{2}$	0.7071
3	$\left(4\arctan(\sqrt{2}) + \arctan(460\sqrt{2}/329)\right)/(8\sqrt{2})$	0.4352
4	$\arctan(1/(2\sqrt{2}))/\sqrt{2}$	0.2403

Although $\mathcal{I}(p)$ is nasty, we can still get bounds on it and then look at "expected decrease per objective evaluation"

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

Average-Case Analysis

Random subspace methods based on finite differencing for $\nabla f(\boldsymbol{x}_k)$ give a similar question: look at expected 2-norm of first p components of random unit vector (much nicer than ∞-norm) to get a similar result:

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

Average-Case Analysis

Random subspace methods based on finite differencing for $\nabla f(\boldsymbol{x}_k)$ give a similar question: look at expected 2-norm of first p components of random unit vector (much nicer than ∞-norm) to get a similar result:

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

Theorem (Hare, R. & Royer, 2023)

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

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

Average-Case Analysis

Random subspace methods based on finite differencing for $\nabla f(\boldsymbol{x}_k)$ give a similar question: look at expected 2-norm of first p components of random unit vector (much nicer than ∞-norm) to get a similar result:

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

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 $p \geq 1$.

Conclusions & Future Work

Conclusions

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

Conclusions & Future Work

Conclusions

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

Future Work

- Second-order analysis (second-order stationarity conditions, random quadratic objectives)
- Problems with constraints

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. 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.

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.

References ii

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).

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

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