Unconstrained Derivative-Free Optimization of Functions with Additive Noise

Elise Reed

University of Colorado Denver

April 12, 2018

OUTLINE

1. Overview of derivative-free optimization
   - Motivation for using zero order methods
   - Features of derivative-free algorithms
   - Basic framework of stochastic derivative-free schemes
2. Review of three random derivative-free methods
   - STARS
   - RG
   - RP
3. Numerical Experiments
   - Accuracy vs. Number of Function Evaluations
   - STARS with variance estimation
4. Conclusion
5. References

$$\min \tilde{f}(x)$$

- $\tilde{f}(x) = f(x) + \nu(\sigma)$
- f is convex and continuously differentiable
- $\nu(\sigma)$ is a noise term with zero mean and bounded variance
- the problem is unconstrained

Motivation

1. Objective function is a black-box
2. Objective function is not differentiable
3. Calculation of the derivative is intractable
4. Estimations of the derivative are inaccurate

Features of derivative-free algorithms

1. Ability to impose descent when not at a stationary point

2. Ability to control the geometry of the set where the objective function is evaluated

3. Ability to generate a sequence of step-sizes that converges down to zero

Random Search

- random search is one of the most basic ideas for optimizing an objective function *f*
  1. start at a point *x*, and choose another point *y* at random
  2. if f(y) < f(x), move to *y*.
  3. if not, stay at *x* and choose a new point *z*

- this method is inefficient given the details above

- improvements can be made to make the scheme tractable for certain problems

- Random Pursuit is the algorithm presented later that uses a type of random search

Random Derivative Approximations

- main ideas
  1. the directional derivative is much more well behaved than the gradient in many senses
  2. using directions drawn from the normal distribution allows for accurate estimation of the gradient

- general framework
  - given a current iterate x_k, step-size h_k, a 'smoothing' step-size μ_k (where $\mu_k \to 0$), and a random direction u

$$x_{k+1} = x_k - h_k \frac{f(x_k + \mu_k u) - f(x_k)}{\mu_k} u$$

- STARS and RG are the algorithms presented later that use this type of gradient approximation

Intro to Gaussian Smoothing

- the scheme presented in the previous slide relies on a smoothed version of the objective function

$$f_\mu(x) = \mathbb{E}_u[f(x + \mu u)]$$

and a finite difference approximation of the directional derivative of f in the direction u

$$g_\mu(x) := \frac{f(x + \mu u) - f(x)}{\mu} u$$

- when f is differentiable at x, we have the important result

$$\mathbb{E}_u[g_\mu(x)] = \nabla f_\mu(x)$$

Step-size Approximation in Randomized Search

- introduced by Ruobing Chen and Stefan M. Wild in 2015
- a variation on the random derivative approximation scheme introduced above for noisy objective functions
  - we are only considering the case of additive noise
- requires the objective function to have the following properties
  1. continuously differentiable
  2. convex
  3. Lipschitz continuous gradient:

$$||\nabla f(x) - \nabla f(y)|| \leq L_1 ||x - y||$$

for all $x, y \in \mathbb{R}^n$, where L_1 is the Lipschitz constant of the gradient

Step-size Approximation in Randomized Search

- uses noise-adjusted smoothing step-size to minimize the least-squares error between the finite-difference approximation and the directional derivative

$$\mu = \left[\frac{8\sigma^2 n}{L_1^2 (n+6)^3} \right]^{\frac{1}{4}}$$

- converges in

$$N \sim \mathcal{O}\left(\frac{n}{\epsilon} L_1 R^2 \right)$$

  iterations, where n is the dimension of the problem, ϵ is the target accuracy, L_1 is the Lipschitz constant of the gradient, and R is the norm of the difference $||x^0 - x^*||$ between the initial point x^0 and the minimum x^*

Random Search for Smooth Optimization

- introduced by Yurii Nesterov in 2011
- one of the first variations on the general random derivative approximation scheme introduced above, not designed for an objective function contaminated with noise
- requires the objective function to have the following properties
    1. continuously differentiable
    2. convex
    3. Lipschitz continuous gradient:

$$||\nabla f(x) - \nabla f(y)|| \leq L_1 ||x - y||$$

for all $x, y \in \mathbb{R}^n$, where L_1 is the Lipschitz constant of the gradient

Random Search for Smooth Optimization

- uses target accuracy based smoothing step-size

$$\mu = \frac{5}{3(n+4)}\sqrt{\frac{\epsilon}{2L_1}}$$

- converges in

$$N \sim \mathcal{O}\left(\frac{n}{\epsilon}L_1 R^2\right)$$

iterations, where n is the dimension of the problem, ϵ is the target accuracy, L_1 is the Lipschitz constant of the gradient, and R is the norm of the difference $||x^0 - x^*||$ between the initial point x^0 and the minimum x^*

Random Pursuit

- introduced by S. U. Stich, C. L. Muller, and B. Gartner in 2012

- a variation on the general random search scheme introduced above, not designed for an objective function contaminated with noise

- requires the objective function to have the following properties
  1. differentiable
  2. convex
  3. bounded curvature in the following sense:

$$f(y) - f(x) - \langle \nabla f(x), y - x \rangle \leq \frac{L_1}{2} ||x - y||^2$$

  for all $x, y \in \mathbb{R}^n$.

  - it should be noted that it is not necessary to know L_1 in order to run the algorithm, it is merely for theoretical purposes.

Random Pursuit

- uses line search oracle

$$\mathrm{LS}(x,u) \in \arg\min_{h\in\mathbb{R}} f(x+hu)$$

  it must be noted that using this line search is very computationally expensive. It can be difficult to know how many times the oracle calls the objective function.

- a target accuracy ϵ can be achieved in

$$N \sim \mathcal{O}\left(\frac{n}{\epsilon}\right)$$

  iterations, where n is the dimension of the problem and we assume line search accuracy μ

$$\mathrm{LS}(x,u) - \mu \leq \mathrm{LS}\textsc{approx}_{\mu}(x,u) \leq \mathrm{LS}(x,u) + \mu$$

Results from Chen and Wild

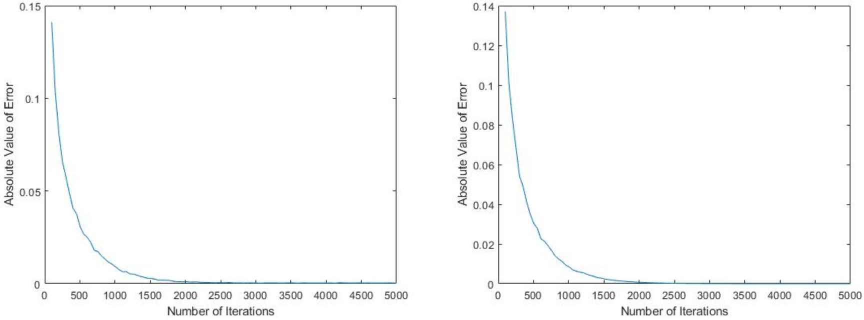
- reproduce the results given in the STARS paper to make sure the algorithms are being implemented correctly

- we currently consider the case of an objective function with additive noise generated from a uniform distribution of zero mean and given standard deviation

- I was able to reproduce the level of accuracy seen by Chen and Wild when the mean over 20 random seeds is found for number of function evaluations between 100 and $10,000$

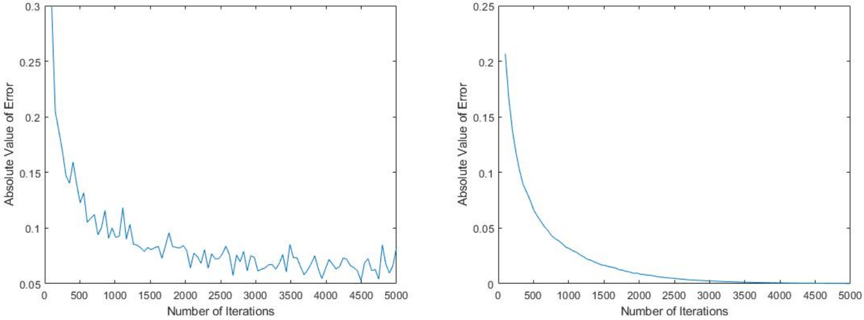
STARS with Uniform Noise

STARS: $\sigma_1 = 10^{-3}$, $\sigma_2 = 10^{-6}$



RG with Uniform Noise

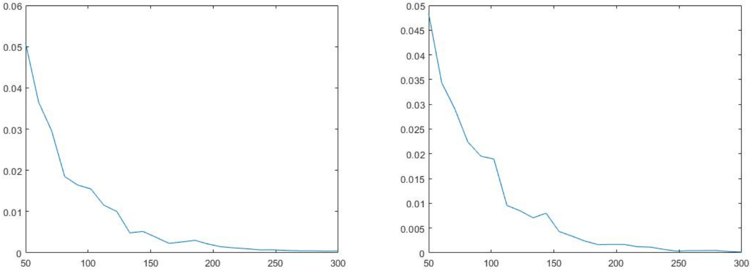
RG: $\sigma_1 = 10^{-3}$, $\sigma_2 = 10^{-6}$



Random Pursuit with Uniform Noise

- RP is not considered in the STARS paper referenced previously, but I also compare its performance in this project
- recall that RP requires fewer iterations to converge than STARS or RG, but its line search oracle makes each iteration computationally expensive

RP: $\sigma_1 = 10^{-3}$, $\sigma_2 = 10^{-6}$

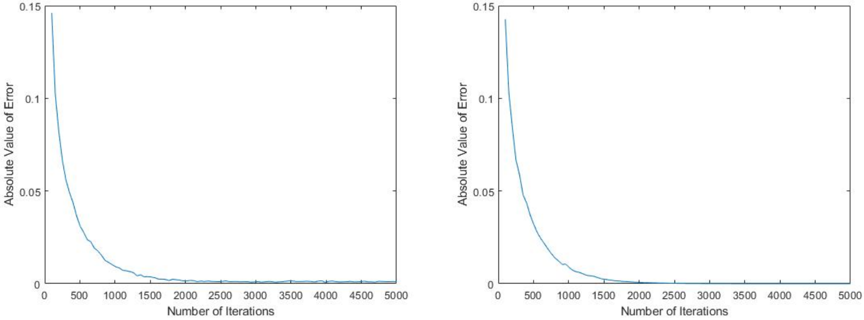


Noise with Other Distributions

- Chen and Wild only consider the case of noise generated from a uniform distribution
- I test the "robustness" of the algorithms in the sense of their ability to handle noise drawn from various distributions
- the distributions tested are a normal distribution and a Gaussian mixture model, both with zero mean and the same variances as used previously

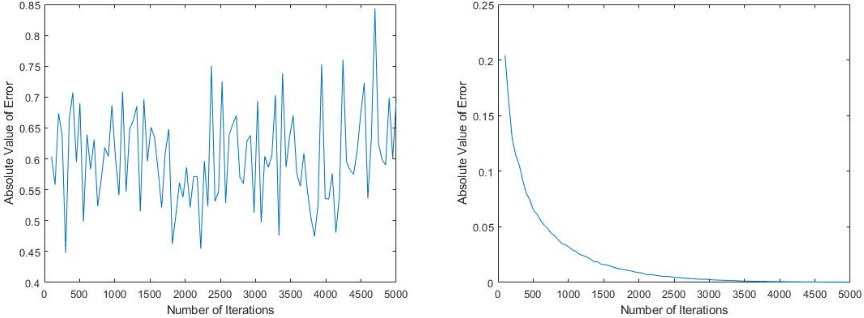
STARS with Normal Noise

$$\sigma_1 = 10^{-3}, \sigma_2 = 10^{-6}$$



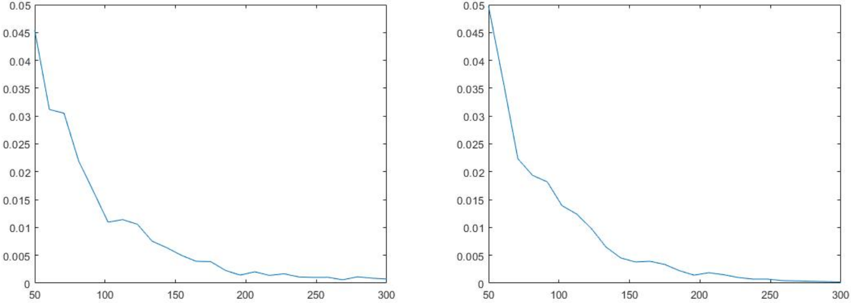
RG with Normal Noise

$$\sigma_1 = 10^{-3}, \sigma_2 = 10^{-6}$$



RP with Normal Noise

$$\sigma_1 = 10^{-3}, \sigma_2 = 10^{-6}$$



The Gaussian Mixture Model

- a mixture model is a weighted combination of multiple distributions
- here, two identical Gaussian distributions are combined with equal weight

- the aim is to create a model that have the same standard deviations as we have been using thus far ($\sigma_1 = 10^{-3}$, $\sigma_2 = 10^{-6}$)

- the variance σ^2 of a mixture of two distributions is given by

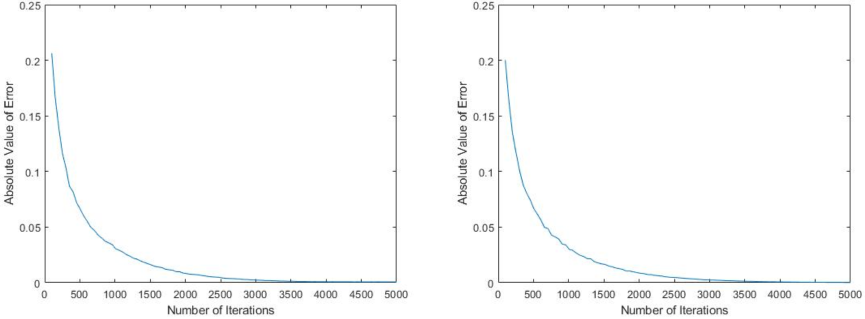
$$\sigma^2 = p_A \sigma_A^2 + p_B \sigma_B^2 + \left[p_A \mu_A^2 + p_B \mu_B^2 - (p_A \mu_A + p_B \mu_B)^2 \right]$$

  where p_I is the weight of the component, σ_I is the standard deviation of the component, and μ_I is the mean of the component

- setting $p_A = p_B = .5$, $\sigma_A = \sigma_B$, $\mu_A = -\mu_B$, and $\sigma = \sigma_i$, we can determine appropriate values for the mean and variance of each component

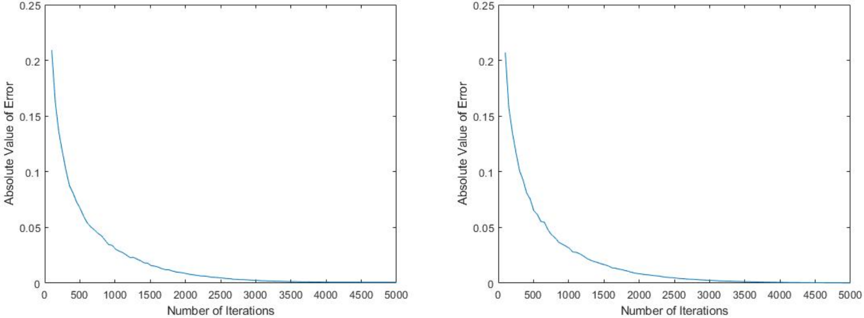
STARS with Mixed Gaussian Noise

$$\sigma_1 = 10^{-3}, \sigma_2 = 10^{-6}$$



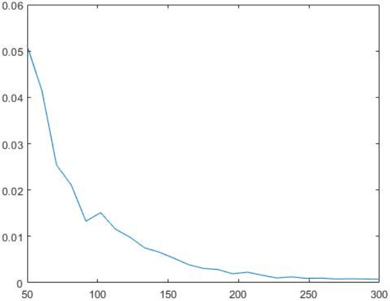
RG with Mixed Gaussian Noise

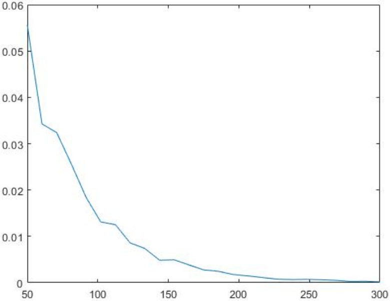
$$\sigma_1 = 10^{-3}, \sigma_2 = 10^{-6}$$



RP with Mixed Gaussian Noise

$$\sigma_1 = 10^{-3},\ \sigma_2 = 10^{-6}$$





Another Test Function

- results given by Chen and Wild, and Nesterov are from experiments using only one test function

$$f_1(x) = \frac{1}{2}(x^{(1)})^2 + \frac{1}{2}\sum_{i=1}^{n-1}(x^{(i+1)} - x^{(i)})^2 + \frac{1}{2}(x^{(n)})^2 - x^{(1)}$$

- the aim is to confirm the results given above hold for different test functions
  - it is difficult, especially in higher dimensions, to find a function that has the necessary properties
- I use the sphere function to rerun the above experiments

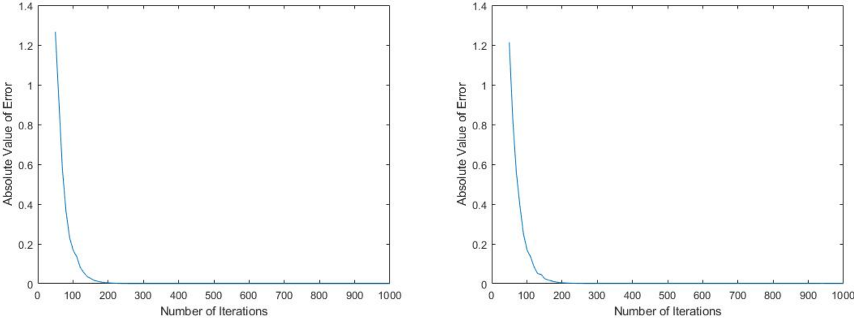
$$\sum_{i=1}^{n} x_i^2$$

- not surprisingly, all three algorithms perform well on this simple function

Another Test Function

- STARS performed so well on the sphere test function that it was tested on a much larger variance $\sigma = 1$
  - to achieve the accuracy seen above, 100 seeds were used

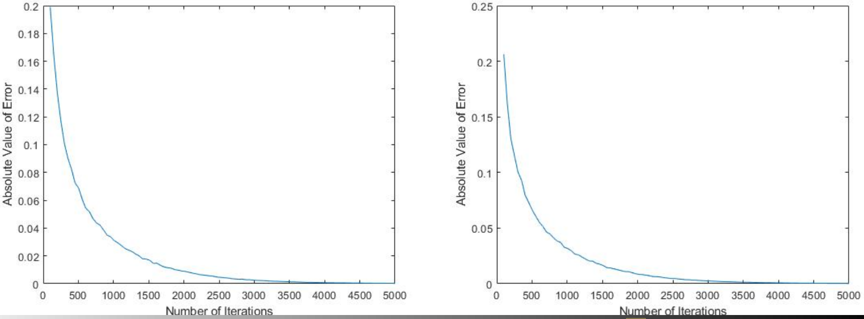
Uniform Noise, Normal Noise



Sampling the Noisy Function for Variance Estimation

- the smoothing step-size in STARS depends on the variance of the noise
- STARS is tested using only an estimator for the variance instead of the true variance
  - the variance is estimated by sampling $\tilde{f}$ 50 times and calculating the variance of this set of samples

STARS: Mixed Gaussian Noise, Normal Noise



Recap of Derivative-Free Optimization

- zero-order methods have become important as models have become more complicated and computational methods have changed
- these methods share some features of derivative-based methods and have a few unique features of their own
  - ability to impose descent when not at a stationary point
  - ability to control the geometry of the set where the objective function is evaluated
  - ability to generate sequences of step-sizes that converge down to zero
- considering random methods has greatly improved the performance of basic zero-order schemes such as direct search and derivative approximation

Takeaways from the Numerical Experiments

- the noise-adjusted smoothing step-size used in STARS gives the scheme the ability to handle many different types of noise

- RG performs as well as STARS when the variance of the noise is small enough. It was found that with larger variance this method is not guaranteed to converge given a noise term of arbitrary distribution

- RP performs well when given additive noise of various distributions. However, the line search oracle called in each iteration makes the computations very expensive. Even with calling less than one third of the number of iterations as with STARS or RG, RP takes significantly longer (in the sense of run time) to converge

Future Work

- find more complicated test functions that still satisfy the requirements of the algorithms

- find more exotic distributions to test

- find a model for the noise term when the variance is unknown, and use the variance of this model as an estimator to the true variance

1. Chen, Ruobing, and Stefan M. Wild. ”Randomized Derivative-Free Optimization of Noisy Convex Functions.” *U.S. Department of Energy, Office of Science*, 12 July 2015.
2. ”Introduction.” *Introduction to Derivative-Free Optimization* , by Andrew R Conn et al., 2009, pp. 1 - 12. MOS-SIAM Series on Optimization.
3. Nesterov, Yurii, and Vladimir Spokoiny. ”Random Gradient-Free Minimization of Convex Functions.” *Foundations of Computational Mathematics*, vol. 17, no. 2, 2015, pp. 527-566., doi:10.1007/s10208-015-9296-2.
4. B.T. Polyak, "Introduction to Optimization". *Optimization Software*, 1987.
5. Stich, S. U., et al. ”Optimization of Convex Functions with Random Pursuit.” *SIAM Journal on Optimization*, vol. 23, no. 2, 2013, pp. 1284-1309., doi:10.1137/110853613.

6. Surjanovic, Sonja, and Derek Bingham. ”Virtual Library of Simulation Experiments” *Optimization Test Functions and Datasets*, Simon Fraser University, 2013, `www.sfu.ca/~ssurjano/optimization.html`.
7. `https://stats.stackexchange.com/users/919/whuber`, whuber. ”What Is the Variance of the Weighted Mixture of Two Gaussians?” *What Is the Variance of the Weighted Mixture of Two Gaussians?*, 20 May 2015, `stats.stackexchange.com/questions/16608/what-is-the-variance-of-the-weighted-mixture-of-two-gaussians?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa`.