Derivative-Free Optimization of Noisy Functions via Quasi-Newton Methods

Jorge Nocedal

Northwestern University

Huatulco, Jan 2018

Collaborators

Albert Berahas
Northwestern University

Richard Byrd
University of Colorado

Discussion

1. The BFGS method continues to surprise
2. One of the best methods for nonsmooth optimization (Lewis-Overton)
3. Leading approach for (deterministic) DFO derivative-free optimization
4. This talk: Very good method for the minimization of noisy functions

We had not fully recognized the power and generality of quasi-Newton updating until we tried to find alternatives!

Subject of this talk:

1. Black-box noisy functions
2. No known structure
3. Not the finite sum loss functions arising in machine learning, where cheap approximate gradients are available

Outline

Problem 1: $\min f(x)$ f smooth but derivatives not available: DFO

1. f contains no noise
2. Scalability, Parallelism
3. Robustness

Problem 2: $\min f(x;\xi)$ $f(\cdot;\xi)$ smooth

$$\min f(x) = \phi(x) + \epsilon(x) \qquad f(x) = \phi(x)(1+\epsilon(x))$$

- Propose method build upon classical quasi-Newton updating using finite-difference gradients
- Estimate good finite-difference interval h
- Use noise estimation techniques (More'-Wild)
- Deal with noise adaptively
- Can solve problems with thousands of variables
- Novel convergence results – to neighborhood of solution (Richard Byrd)

DFO: Derivative free deterministic optimization (no noise)

$$\min f(x) \qquad f \text{ is smooth}$$

- Direct search/pattern search methods: not scalable
- Much better idea:
  - Interpolation based models with trust regions (Powell, Conn, Scheinberg,…)

$$\min \quad m(x) = x^T Bx + g^T x \quad \text{s.t.} \quad \| x \|_2 \leq \Delta$$

1. Need *(n+1)(n+2)/2* function values to define quadratic model by pure interpolation
2. Can use O(*n*) points and assume minimum norm change in the Hessian
3. Arithmetic costs high: n^4 ← *scalability*
4. Placement of interpolation points is important
5. Correcting the model may require many function evaluations
6. Parallelizable? ←

Why not simply BFGS with finite difference gradients?

$$x_{k+1} = x_k - \alpha_k H_k \nabla f(x_k)$$

$$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(x + he_i) - f(x)}{h}$$

- Invest significant effort in estimation of gradient
- Delegate construction of model to BFGS
- Interpolating gradients
- Modest linear algebra costs *O(n)* for L-BFGS
- Placement of sample points on an orthogonal set
- BFGS is an overwriting process: no inconsistencies or ill conditioning *with* Armijo-Wolfe line search
- Gradient evaluation parallelizes easily

Why now?

- Perception that n function evaluations per step is too high
- Derivative-free literature rarely compares with FD – quasi-Newton
- Already used extensively: fminunc MATLAB
- Black-box competition and KNITRO

Some numerical results

Compare:

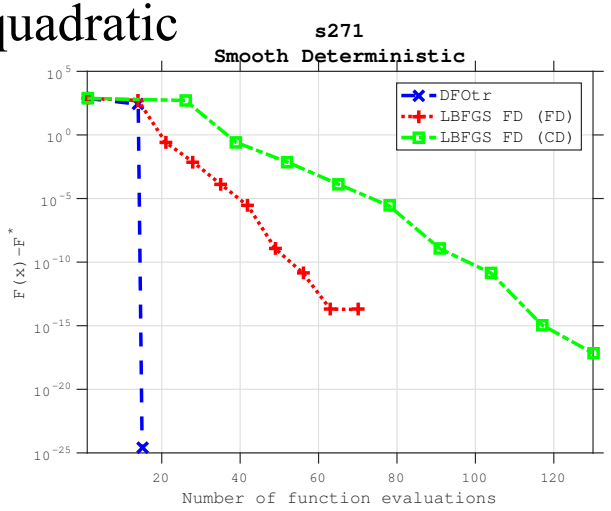
Model based trust region code DFOtr by Conn, Scheinberg, Vicente

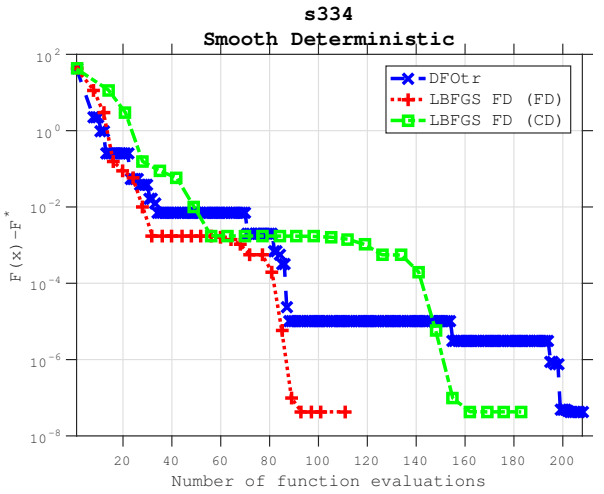
vs FD-L-BFGS with forward and central differences

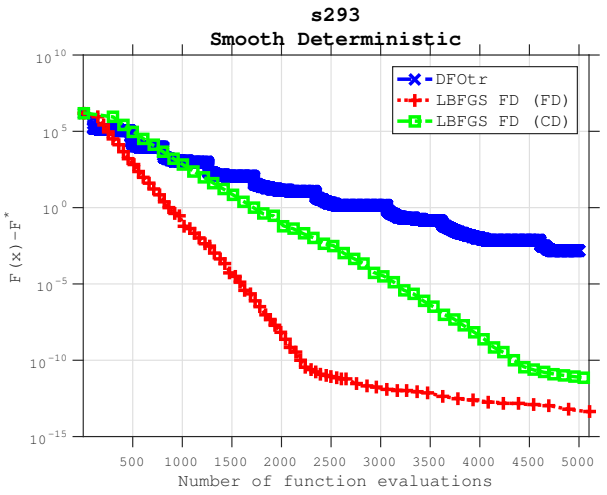
Plot function decrease vs total number of function evaluations

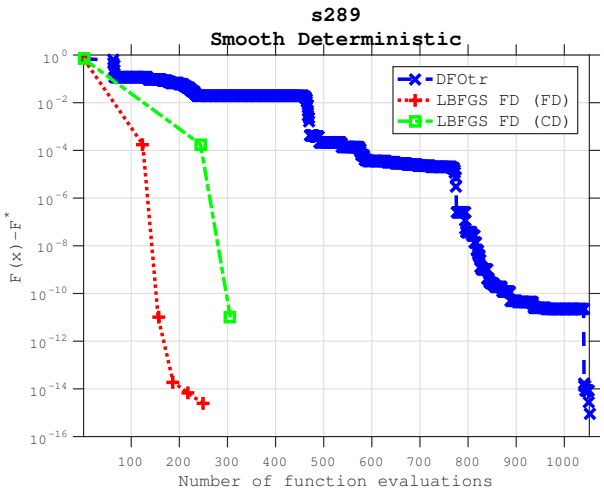
Comparison: function decrease vs total # of function evaluations

quadratic









Conclusion: DFO without noise

Finite difference BFGS is a real competitor of DFO method based on function evaluations
Can solve problems with thousands of variables

… but really nothing new.

Optimization of Noisy Functions

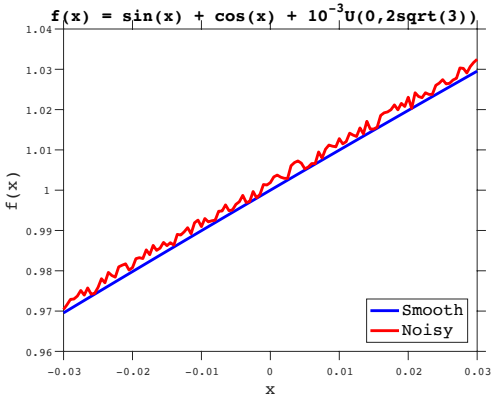
$$\min f(x;\xi) \quad \text{where} \quad f(\cdot;\xi) \text{ is smooth}$$

$$\min f(x) = \phi(x) + \epsilon(x) \qquad f(x) = \phi(x)(1+\epsilon(x))$$

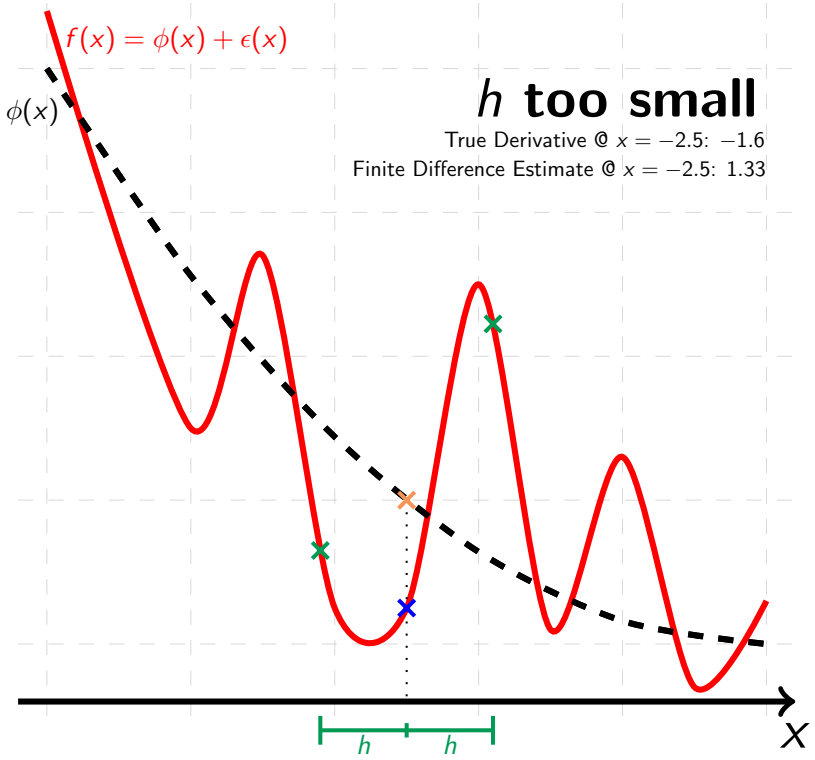
Focus on additive noise

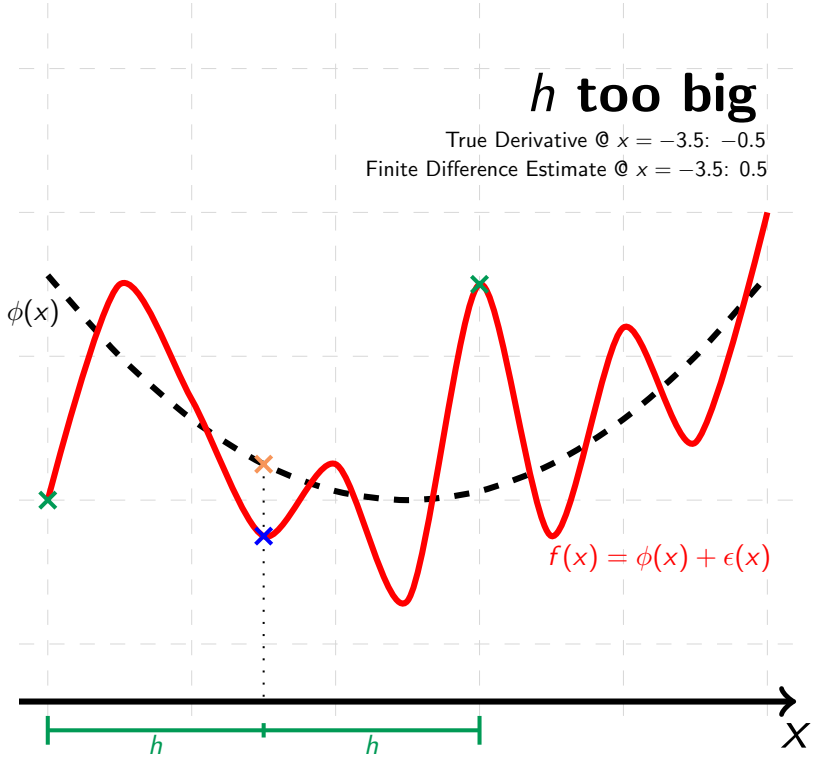
Finite-difference BFGS should not work!

1. Difference of noisy functions dangerous
2. Just one bad update once in a while: disastrous
3. Not done to the best of our knowledge



Finite Differences – Noisy Functions





A Practible Algorithm

$$\min f(x) = \phi(x) + \epsilon(x) \qquad f(x) = \phi(x)(1 + \epsilon(x))$$

Outline of adaptive finite-difference BFGS method

1. Estimate noise $\epsilon(x)$ at every iteration -- More'-Wild
2. Estimate h
3. Compute finite difference gradient
4. Perform line search (?!)
5. Corrective Procedure when case line search fails
   - (need to modify line search)
   - Re-estimate noise level

Will require very few extra f evaluations/iteration – even none

Noise estimation

More'-Wild (2011)

$$\min f(x) = \phi(x) + \epsilon(x)$$

Noise level: $\sigma = [var(\epsilon(x))]^{1/2}$

Noise estimate: ϵ_f

At x choose a random direction v

evaluate f at $q+1$ equally spaced points $x + i\beta v, \quad i = 0,...,q$

Compute function differences:

$$\Delta^0 f(x) = f(x)$$

$$\Delta^{j+1} f(x) = \Delta^j [\Delta f(x)] = \Delta^j [f(x+\beta)] - \Delta^j [f(x)]]$$

Compute finite diverence table:

$$T_{ij} = \Delta^j f(x + i\beta v)$$

$$1 < j < q \quad 0 < i < j - q$$

$$\sigma_j = \frac{\gamma_j}{q-1-j} \sum_{i=0}^{q-j} T_{i,j}^2 \qquad \gamma_j = \frac{(j!)^2}{(2j)!}$$

Noise estimation

More'-Wild (2011)

$$\min f(x) = \sin(x) + \cos(x) + 10^{-3}\mathrm{U}(0, 2\sqrt{3}) \qquad q = 6 \qquad \beta = 10^{-2}$$

x	f	Δf	$\Delta^2 f$	$\Delta^3 f$	$\Delta^4 f$	$\Delta^5 f$	$\Delta^6 f$
$-3 \cdot 10^{-2}$	1.003	$7.54e-3$	$2.15e-3$	$1.87e-4$	$-5.87e-3$	$1.46e-2$	$-2.49e-2$
$-2 \cdot 10^{-2}$	1.011	$9.69e-3$	$2.33e-3$	$-5.68e-3$	$8.73e-3$	$-1.03e-3$	
-10^{-2}	1.021	$1.20e-2$	$-3.35e-3$	$3.05e-3$	$-1.61e-3$		
0	1.033	$8.67e-3$	$-2.96e-3$	$1.44e-3$			
10^{-2}	1.041	$8.38e-3$	$1.14e-3$				
$2 \cdot 10^{-2}$	1.050	$9.52e-3$					
$3 \cdot 10^{-2}$	1.059						
σ_k		$6.65e-3$	$8.69e-4$	$7.39e-4$	$7.34e-4$	$7.97e-4$	$8.20e-4$

High order differences of a smooth function tend to zero rapidly, while differences in noise are bounded away from zero. Changes in sign, useful.

Procedure is scale invariant!

Finite difference itervals

Once noise estimate ϵ_f has been chosen:

Forward difference: $h = 8^{1/4}(\frac{\epsilon_f}{\mu_2})^{1/2}$ $\mu_2 = \max_{x \in I} |f''(x)|$

Central difference: $h = 3^{1/3}(\frac{\epsilon_f}{\mu_3})^{1/3}$ $\mu_3 \approx |f'''(x)|$

Bad estimates of second and third derivatives can cause problems (not often)

Adaptive Finite Difference L-BFGS Method

Estimate noise ϵ_f

Compute h for forward or central differences [(4-8) function evaluations]

Compute g_k

While convergence test not satisfied:

$d = -H_k g_k$ [L-BFGS procedure]

$(x_+, f_+, flag) =$ LineSearch$(x_k, f_k, g_k, d_k, f_s)$

IF flag=1 [line search failed]

$(\mathrm{x}_+, f_+, h) =$ Recovery$(x_k, f_k, g_k, d_k, max_{iter})$

endif

$x_{k+1} = x_+, \; f_{k+1} = f_+$

Compute g_{k+1} [finite differences using h]

$s_k = x_{k+1} - x_k, \; y_k = g_{k+1} - g_k$

Discard (s_k, y_k) if $s_k^T y_k \le 0$

$k = k + 1$

endwhile

Line Search

BFGS method requires Armijo-Wolfe line search

$$f(x_k + \alpha d) \le f(x_k) + \alpha c_1 \nabla f(x_k) d \qquad \text{Armijo}$$

$$\nabla f(x_k + \alpha d)^T d \ge c_2 \nabla f(x_k)^T d \qquad \text{Wolfe}$$

Deterministic case: always possible if f is bounded below

- Can be problematic in the noisy case.
- Strategy: try to satisfy both but limit the number of attempts
- If first trial point (unit steplength) is not acceptable relax:

$$f(x_k + \alpha d) \le f(x_k) + \alpha c_1 \nabla f(x_k) d \; + 2\epsilon_f \qquad \text{relaxed Armijo}$$

Three outcomes: a) both satisfied; b) only Armijo; c) none

Corrective Procedure

Compute a new noise estimate $\overline{\epsilon}_f$ along search direction d_k

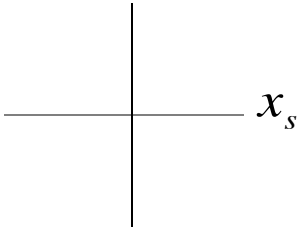
Compute corresponding $\overline{h}$

If $\hat{h} \neq h$ use new estimat $h \leftarrow \overline{h}$; return w.o. changing x_k

Else compute new iterate (various options):

small perturbation; stencil point

Finite difference Stencil (Kelley)



Some quotes

I believe that eventually the better methods will not use derivative approximations…
[Powell, 1972]

f is … somewhat noisy, which renders most methods based on finite differences of little or no use [X,X,X]. [Rios & Sahinidis, 2013]

END