Zero-Order Methods for the Optimization of Noisy Functions

Jorge Nocedal

Northwestern University

Simons Institute, October 2017

Collaborators

Albert Berahas
Northwestern University

Richard Byrd
University of Colorado

Motivation

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

1. Scalability
2. Parallelism
3. Beyond linear models
4. But should not aim for fully quadratic model
5. Spread function evaluations effectively

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

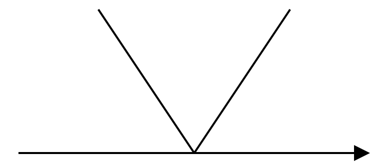
1. Use noise estimation techniques (Hamming 1960s)
2. Estimate good finite-difference interval h
3. Classical quasi-Newton updating using finite-difference gradients
4. Deal with noise adaptively
5. Can solve problems with thousands of variables
6. Convergence to a neighborhood of solution

Nonsmooth Optimization

Lewis and Overton

BFGS method with the right line search is more effective in practice than bundle methods or any other approach they tried (Curtis)

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



$$f(x) = |x|$$

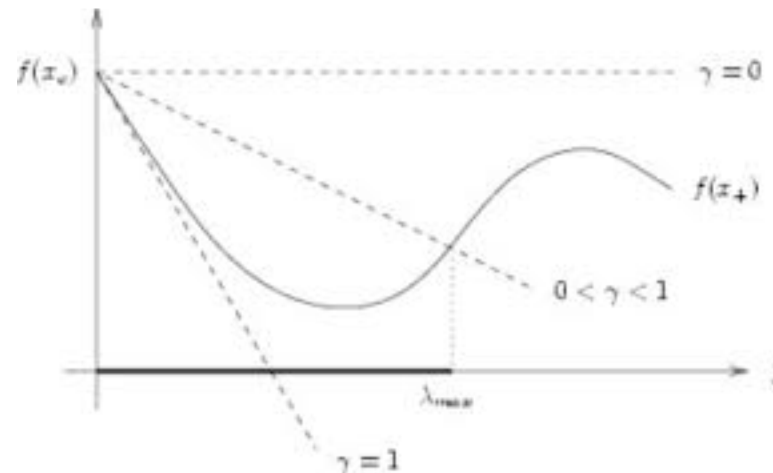
The Wolfe line search ensures that a convex model can be created. Only assume function is bounded below

Gradient exists almost everywhere:

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

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

$$0 < c_1 < c_2 < 1$$



Nonsmooth Optimization

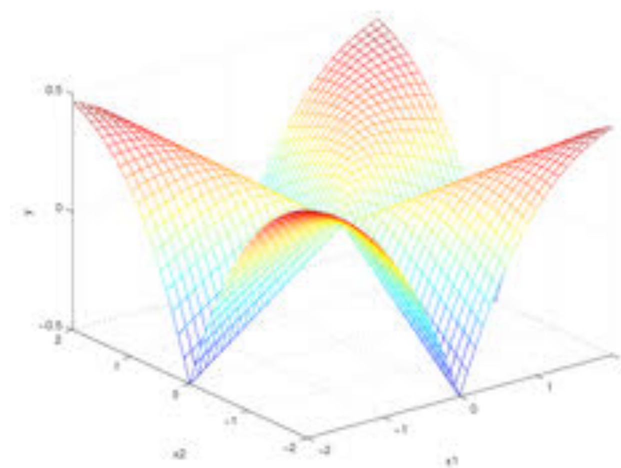
Lewis and Overton:

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

The BFGS matrix captures the U-V structure of the objective

Hessian approximation blows up (good thing)

Never observed failures
Very limited convergence results
Where do we go from here?



Power of Armijo-Wolfe line search not appreciated by the convex analysis community
Rather than constructing a majorizing function, one constructs a convex model along the search direction

Discussion

1. The BFGS method continues to surprise
2. One of the leading algorithms for **nonsmooth** optimization
3. Leading approach for (deterministic) derivative-free optimization
4. This talk: Leading method for the minimization of noisy functions

These observations do not apply to:

1. Structured nonsmooth optimization (e.g. lasso)
2. Stochastic objectives with cheap gradient, as in machine learning
3. Nonlinear least squares objectives; Gauss-Newton is the right approach

We had not fully recognized the power and generality of quasi-Newton updating

Derivative free deterministic optimization (no noise)

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

- Interpolation based models with trust regions (Katya)

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

1. Need *(n+1)(n+2)/2* function values to define quadratic model by pure interpolation
2. Can use n points and assume minimum norm change in the Hessian
3. Arithmetic costs high: n^4
4. Placement of interpolation points is important
5. Trust region constraint needed – and natural
6. Parallelizable?

BFGS with finite difference gradients: deterministic case

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

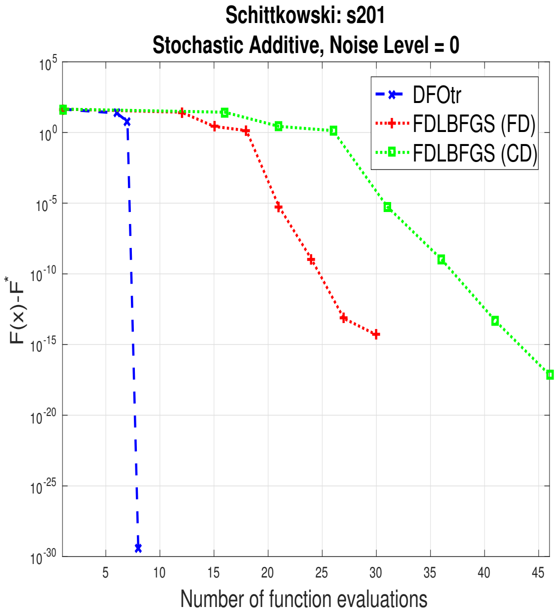
$$\boxed{\frac{\partial f(x)}{\partial x_i} \approx \frac{f(x + he_i) - f(x)}{h}}$$

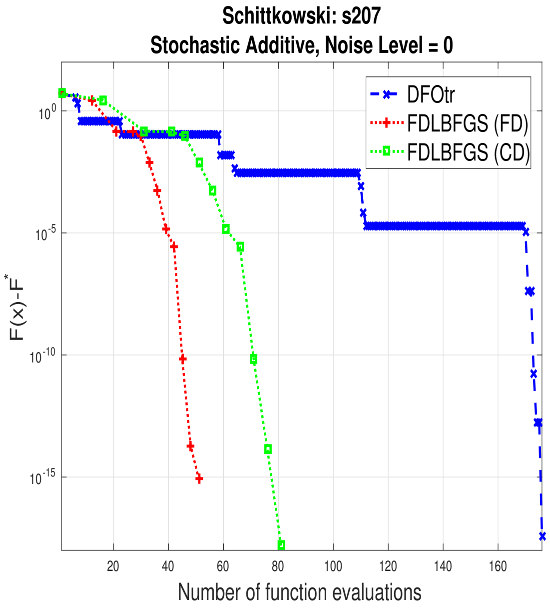
- Invest significant effort in estimation of gradient
- Delegate construction of model to BFGS
- Interpolating gradients
- Modest linear algebra costs *O(n)*
- Placement of sample points on an orthogonal set
- BFGS is an overwriting process: no inconsistencies or ill conditioning *with* Armijo-Wolfe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

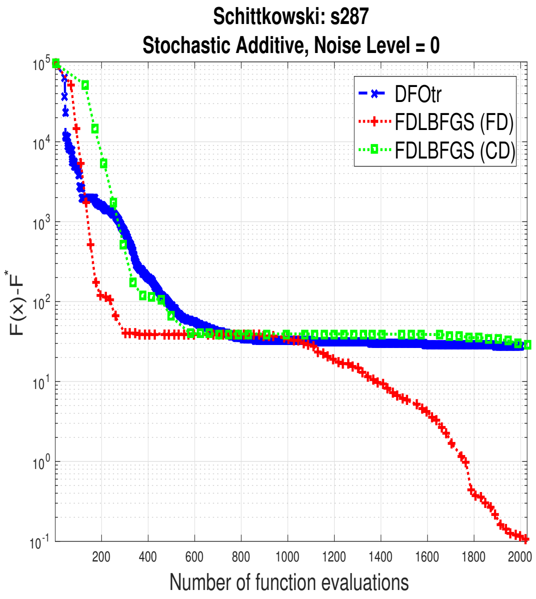
Why now?

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

Comparison: function decrease vs total # of function evaluations







Optimization of Noisy Functions

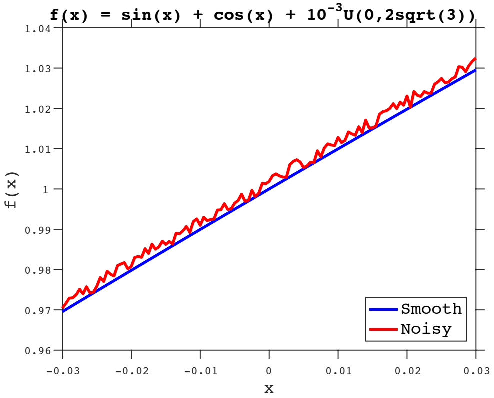
$$\min f(x;\xi) \qquad \text{where} \quad f(\cdot;\xi) \text{ is smooth}$$

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

Additive and multiplicative noise. Focus on additive

Outline of adaptive finite-difference BFGS method

1. Estimate noise *e(x)* at every iteration,
2. Possibly change h
3. Corrective Procedure in case line search fails
4. (need to modify line search)



Noise estimation

More'-Wild (2011)

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

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

Noise estimate: ϵ_f

At x choose a random direction v

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

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:

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

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

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

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

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	$1.20e-2$	$-3.35e-3$	$3.05e-3$	$-1.61e-3$	2	
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 make cause problems (not often)

Adaptive Finite Difference L-BFGS Method

Estimate noise ϵ_f

Compute h by 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 $s_k^T y_k \leq 0$

$k = k + 1$

endwhile

Adaptive Finite Difference L-BFGS Method

Estimate noise ϵ_f

Compute h by 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

Corrective Procedure

Compute new noise estimate $\bar{\epsilon}_f$ along search direction d_k;

Compute corresponding $\bar{h}$

IF $\bar{h} \notin [0.7h, 1.5h]$

$h = \bar{h}$, $x_+ = x_k$, $f_+ = f_k$ [update h; do not move]

ELSE

$x_+ = x_k + h d_k / \| d_k \|$, $f_+ = f(x_+)$ [perturbation]

If x_+ satisfies the relaxed Armijo condition

return x_+, h

else

if $f_+ \le f_s$ and $f_+ \le f_k$ accept x_+

else if $f_k > f_s$ and $f_+ > f_s$ $x_+ = x_s$, $f_+ = f_s$

else $x_+ = x_k$ $f_+ = f_k$

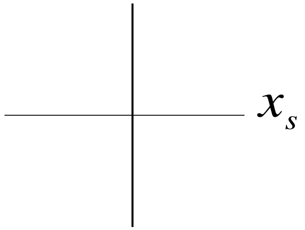
compute new ϵ_f, h [random v]

end if

end if

ENDIF

Finite difference Stencil (Kelley)



Line Search

BFGS method requires Armijo-Wolfe line search

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

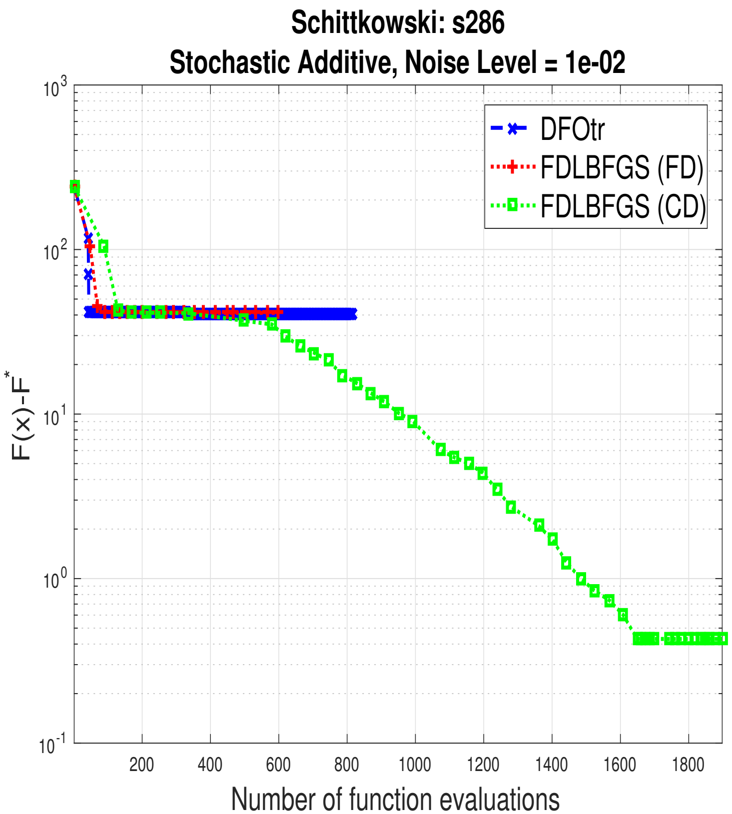
$$\nabla f(x_k + \alpha d)^T d \ge c_2 \nabla f(x_k)^T d \qquad \text{Wolfe}$$

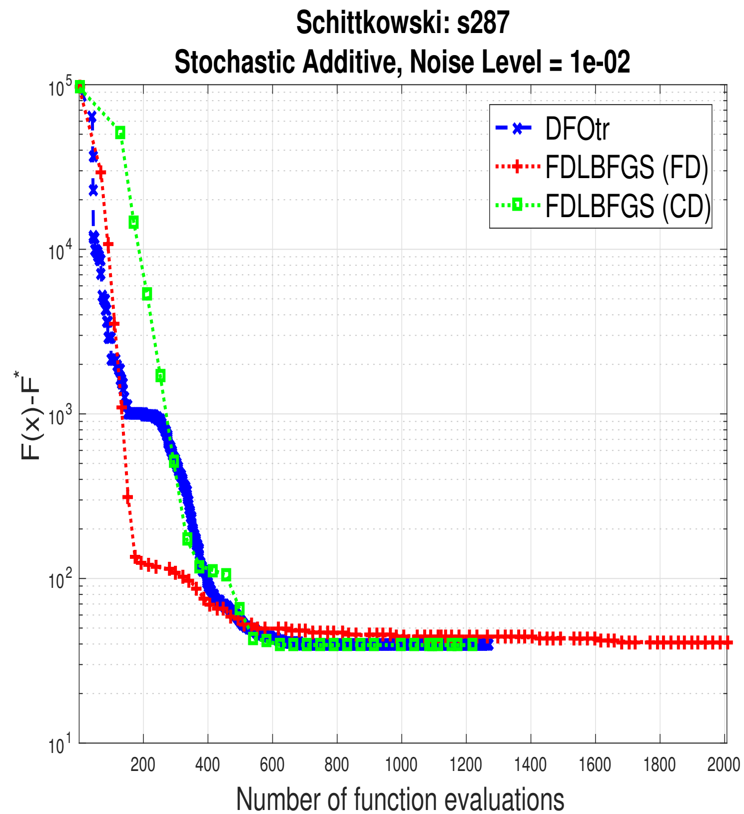
Deterministic case: always possible if f is bounded below

- Can be problematic in the noisy case. Direction d may not be a descent direction for smooth underlying function
- 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


Schittkowski: s286
Stochastic Additive, Noise Level = 1e-02
DFOtr
FDLBFGS (FD)
FDLBFGS (CD)
F(x)-F*
Number of function evaluations


Schittkowski: s287
Stochastic Additive, Noise Level = 1e-02
DFOtr
FDLBFGS (FD)
FDLBFGS (CD)
F(x)-F*
Number of function evaluations

A simple convergence result: constant steplength

Assumptions:

1. $f(x) = \phi(x) + \epsilon(x)$. Function ϕ is twice differentiable
2. Strong convexity. $\mu I \prec \nabla^2 \phi(x) \prec LI \quad \forall x \in R^n$
3. H_k has bounded eigenvalues
4. Bounded noise. $\| \epsilon(x) \| \le \bar{\epsilon} \quad \forall x \in R^n$

Theorem. If

$$\alpha < \frac{1-\beta}{(1-\beta)L + \beta^2 L} \qquad \text{for any } \beta \in (0,1)$$

Then for all k

$$\phi(x_k) - \phi^* \le (1-\rho)^k (\phi(x_0) - \phi^* - \eta / \rho) + \eta / \rho$$

$$\rho = 1 - \alpha\mu(1-\beta)$$

$$\eta = \alpha\left[\frac{1-\alpha L}{\beta} + \frac{\alpha L}{2}\right]\bar{\epsilon}^2$$

END