Derivative-Free Optimization via Proximal Point Methods

Yves Lucet & Warren Hare

a place of mind

UBC

July 24, 2013

Outline

Derivative Free Optimization

Problem

$$\min\{f(x) : x \in \mathbb{R}^n\}, \;\; f \text{ not analytically available.}$$

Typical framework

- Gradients are unavailable or too expensive (simulation...)
- The objective function is noisy (values/gradients are inexact)
- The algorithm is going to be used on a wide variety of problems
  - need wide convergence properties
  - desire easy to follow structure

Derivative Free Optimization

Methods

- Numerical Differentiation — Stability
- Automatic Differentiation — Need Source Code
- Evolutionary/Genetic Algorithms, simulated Annealing — No or weak convergence results
- Derivative-Free Methods — Robust, Convergence results

Outline

Proximal Point Method

Proximal Map

$$P_r f(x) = \arg\min_y \left\{ f(y) + \frac{r}{2}\|y - x\|^2 \right\}.$$

Proximal Point Method

If P_r is well-defined, then the proximal point method is

$$x^{k+1} \in P_{r^k} f(x^k).$$

Basic Results

Theorem: [Moreau, '63]
If f is convex and $0 \in \partial f(\bar{x})$, then $P_1 f(\bar{x}) = \{\bar{x}\}$.

Theorem: [Martinet, '72]
If f is convex, then $x^{k+1} \in P_1 f(x^k)$ converges to a minimizer.

Theorem: [Rockafellar, '76]
If f is convex and $\bar{x}$ is a strict critical point, then $x^{k+1} \in P_{r^k} f(x^k)$ converges to $\bar{x}$ in a finite number of iterations.

Theorem: [Poliquin & Rockafellar '96 : H. & Lewis '04]
If f is prox-regular and r^k is sufficiently large, then the above still works.

Stability

Theorem: [Moreau, '65]
If f is convex, then $P_1 f$ is Lipschitz.

Theorem: [Poliquin & Rockafellar '96]
If f is prox-regular, then $P_r f$ is Lipschitz for large r.

Theorem: [Hare & Poliquin, '07]
If f_λ is para-prox-regular, then for r large

$$\begin{array}{rcl} |P_r f_\lambda(x) - P_r f_\lambda(\tilde{x})| & \leq & C_x |x - \tilde{x}| \\ |P_r f_\lambda(x) - P_{\tilde{r}} f_\lambda(x)| & \leq & C_r |r - \tilde{r}| \\ |P_r f_\lambda(x) - P_r f_{\tilde{\lambda}}(x)| & \leq & C_f |\lambda - \tilde{\lambda}| \end{array}$$

Robustness

Theorem: [Kiwiel, '90]
Approximating convex f with piecewise linear cutting-planes models creates a convergent algorithm.

Theorem: [Noll et al. '08 : Hare & Sagastizábal, '09]
For nonconvex f, approximating $f + \eta\frac{1}{2}|\cdot|^2$ with piecewise linear cutting planes models creates a convergent algorithm.

Theorem: [Kiwiel, '10]
Approximating convex f with *inexact* piecewise linear models creates a convergent algorithm.

Outline

1. Derivative-Free Optimization
2. Proximal Point Methods
3. **Derivative-Free Proximal Point**
4. Conclusion

Sample Radius

$$\Delta(Y) = \max_{y^i \in Y} \|y^i - y^0\|$$

A Derivative-Free Proximal Point Method

1. INITIALIZE: Input starting point and parameters.
2. MODEL AND STOPPING CONDITIONS: Create a quadratic interpolation model of f over sample radius Δ^k

   $$q^k(x) := \alpha^k + \langle g^k, x\rangle + \langle x, H^k x\rangle.$$

   Check stopping conditions ($\|\nabla q^k(x^k)\|$ small and Δ^k small)
3. PROX-FEASIBLITY CHECK:
   If $r^k \leq -\lambda_n(H^k)$, then ($q^k + r^k \frac{1}{2}\|\cdot\|^2$ is not convex) increase r^k
4. PROX TRIAL POINT:

   $$\tilde{x}^k = P_{r^k} q^k(x^k) = (2H_k + r^k Id)^{-1}(r_k x_k - g_k)$$

5. SERIOUS/NULL CHECK:
   If $\tilde{x}^k$ is good, then (declare a serious step) line search in direction $x^k - \tilde{x}^k$
   Else (declare a null step), either increase r^k or decrease Δ^k or both
6. LOOP

Comparison with quasi-Newton trust region methods

Similarities

- quadratic model to approximate the function
- minimize model to obtain the next iterate

Differences

- QN: minimizes other a ball
- PP: minimizes using a quadratic penalty
  - convexify the approximating quadratic so all subproblems are easily solvable
  - automatically enforces an Armijo-like descent that provides a clean convergence analysis

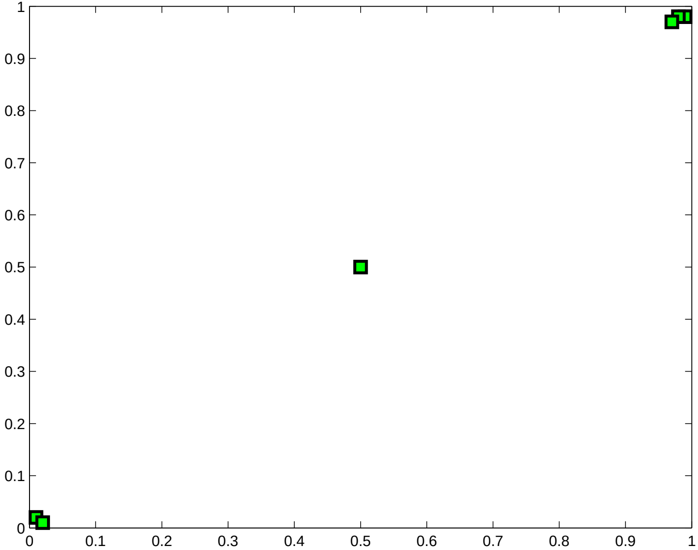
Poisedness

$Y = \{y^0, y^1, \dots y^p\}$ is *poised* for quadratic interpolation over f if $(p+1=)|Y| = (n+1)(n+2)/2$ and there is a *unique* quadratic function q such that

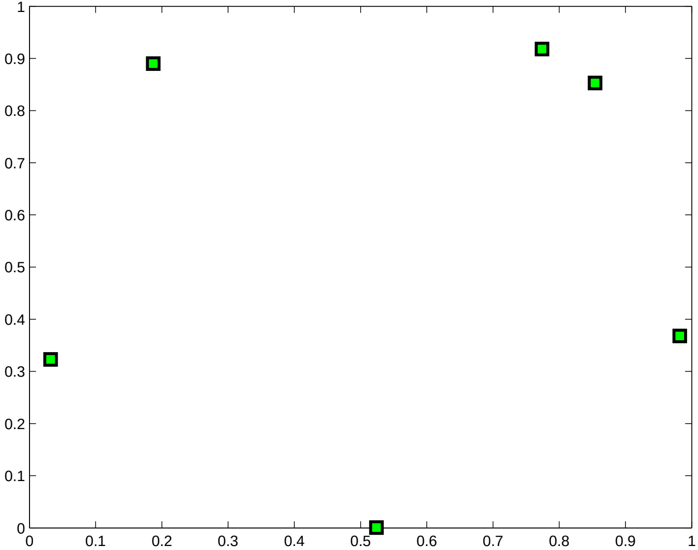
$$q(y^i) = f(y^i) \text{ for each } y^i \in Y.$$

To say the interpolation points Y are poised for quadratic interpolation implies that the points provide reasonable coverage across the full dimension of $\mathbb{R}^n$.

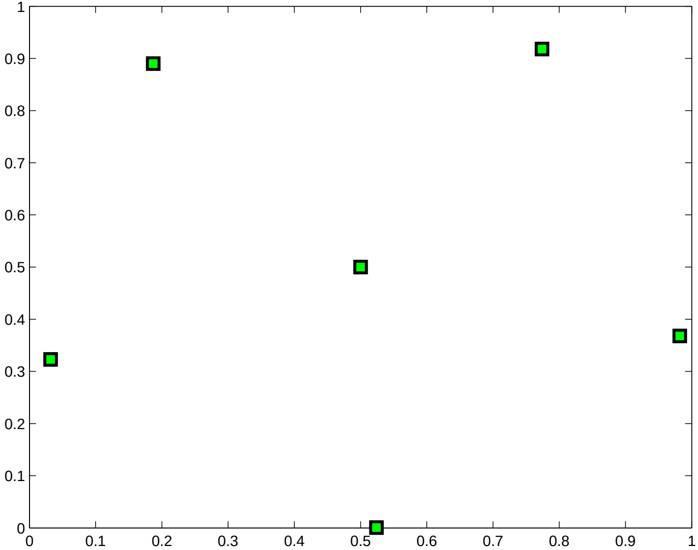
DFPP Algorithm



DFPP Algorithm



DFPP Algorithm



over- and under-determined quadratic models

Quadratic interpolation can be replaced with

- quadratic regression: use more than $(n+1)(n+2)/2$ points and replace the exact interpolation with a least-squares regression to determine the quadratic model.
- underdetermined quadratic models: use less than $(n+1)(n+2)/2$ points and use pick the Lagrange polynomials that generate the minimum Frobenius norm.

Can be done with $O(n)$ points

Stopping

Theorem:
Suppose f is smooth, bounded below, and good interpolation sets are used.

If $\|\nabla q^k(x^k)\|$ is small and Δ^k is small, then $\|\nabla f(x^k)\|$ is small.

Serious/Null choice

$$\text{Predicted decrease: } \delta^k = q^k(x^k) - q^k(\tilde{x}^k).$$

If $f(\tilde{x}^k) \leq f(x^k) - m\delta^k$, SERIOUS STEP:

- (line search) set $x^{k+1} = x^k + \alpha(\tilde{x}^k - x^k)$
- update Y^{k+1} st
  $x^{k+1} \in Y^{k+1}$ and $\Delta(Y^{k+1}) \leq \Delta(Y^k)$

Else NULL STEP:

- if $\tilde{x}^k \notin B_{\Delta(Y^k)}(x^k)$, $r^{k+1} \to 2r^k$: set $x^{k+1} = x^k$ and $Y^{k+1} = Y^k$.
  if $r^{k+1} > r_{\mathtt{tol}}$, STOP
- else set $x^{k+1} = x^k$ and update Y^{k+1} st
  $x^{k+1} \in Y^{k+1}$ and $\Delta(Y^{k+1}) \leq \Gamma\Delta(Y^k)$.

Convergence

Theorem:
Suppose f is smooth, bounded below, and good interpolation sets are used.

If an infinite number of serious steps occur and r^k is bounded above, then

$$\lim \nabla f(x^k) = 0.$$

Convergence

Theorem:
Suppose f is smooth, bounded below, and good interpolation sets are used.

If a finite number of serious steps occurs and an infinite number of null steps occur, then

$$\nabla f(x^k) = 0,$$

where x^k is the result of the last serious step.

Numerical Tests

- The algorithm was implemented in MATLAB.
- The Moré-Garbow-Hillstrom test set was used.
- Successful on 29 of 35 problems.
- The majority of failures were on badly scaled functions
    - e.g., $f(x, y) = (10^4 xy - 1)^2 + (e^{-x} + e^{-y} - 1.0001)^2$

Outline

Summary and Future Directions

- ☑ Proof of concept algorithm that uses ideas from quasi-Newton trust region methods and the proximal point algorithm.
- ☑ DFO Prox-point method convergence proof.
- ☐ Refinement of the implementation to improve numerical results (line search)
- ☐ Developing methods to deal with badly scaled problems.
- ☐ Bundle approaches? Limited memory approaches?
- ☐ (Split) Bregman methods?

Reference

Hare, W. & Lucet, Y. Derivative-Free Optimization Via Proximal Point Methods. Journal of Optimization Theory and Applications, Springer US, 2013, 1-17
http://link.springer.com/article/10.1007%2Fs10957-013-0354-0

Thank you