Recent advances in evaluation complexity for nonconvex optimization

Philippe Toint
(with S. Bellavia, C. Cartis, N. Gould, G. Gurioli and B. Morini)

Namur Center for Complex Systems (naXys), University of Namur, Belgium

(philippe.toint@unamur.be)

VASO Workshop, PolyU, Hong Kong, November 2018

The problem (again)

We consider the unconstrained nonlinear programming problem:

$$\text{minimize} \quad f(x)$$

for $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ smooth.

For now, focus on the

unconstrained case

but we are also interested in the case featuring

inexpensive constraints

An overestimating model

Note the following: if

- f has gradient g and globally Lipschitz continuous Hessian H with constant $2L$

Taylor, Cauchy-Schwarz and Lipschitz imply

$$\begin{array}{rcl} f(x+s) & = & f(x) + \langle s, g(x)\rangle + \frac{1}{2}\langle s, H(x)s\rangle \\ & & + \int_0^1 (1-\alpha)\langle s, [H(x+\alpha s) - H(x)]s\rangle \, d\alpha \\ & \leq & \underbrace{f(x) + \langle s, g(x)\rangle + \frac{1}{2}\langle s, H(x)s\rangle + \frac{1}{3}L\|s\|_2^3}_{m(s)} \end{array}$$

$\Longrightarrow$ reducing m from $s = 0$ improves f since $m(0) = f(x)$.

Griewank, 1981

Approximate model minimization

Lipschitz constant L unknown $\Rightarrow$ replace by adaptive parameter σ_k in the model :

$$m(s) \stackrel{\text{def}}{=} f(x) + s^T g(x) + \tfrac{1}{2} s^T H(x) s + \tfrac{1}{3}\sigma_k \|s\|_2^3 = T_{f,2}(x,s) + \tfrac{1}{3}\sigma_k \|s\|_2^3$$

Computation of the step:

1. minimize $m(s)$ until an approximate first-order minimizer is obtained:

$$\|\nabla_s m(s)\| \leq \kappa_{\text{stop}} \|s\|^2$$

Note: no global optimization involved.

Second-order Adaptive Regularization (AR2)

Algorithm 1.1: The AR2 Algorithm

Step 0: Initialization: x_0 and $\sigma_0 > 0$ given. Set $k = 0$

Step 1: Termination: If $\|g_k\| \leq \epsilon$, terminate.

Step 2: Step computation:

Compute s_k such that $m_k(s_k) \leq m_k(0)$ and $\|\nabla_s m(s_k)\| \leq \kappa_{\rm stop}\|s_k\|^2$.

Step 3: Step acceptance:

Compute $\rho_k = \dfrac{f(x_k) - f(x_k + s_k)}{f(x_k) - T_{f,2}(x_k, s_k)}$

and set $x_{k+1} = \begin{cases} x_k + s_k & \text{if } \rho_k > 0.1 \\ x_k & \text{otherwise} \end{cases}$

Step 4: Update the regularization parameter:

$$\sigma_{k+1} \in \begin{cases} [\sigma_{\min}, \sigma_k] = \frac{1}{2}\sigma_k & \text{if } \rho_k > 0.9 & \textit{very successful} \\ [\sigma_k, \gamma_1\sigma_k] = \sigma_k & \text{if } 0.1 \leq \rho_k \leq 0.9 & \textit{successful} \\ [\gamma_1\sigma_k, \gamma_2\sigma_k] = 2\sigma_k & \text{otherwise} & \textit{unsuccessful} \end{cases}$$

Evaluation complexity: an important result

How many function evaluations (iterations) are needed to ensure that

$$\|g_k\| \leq \epsilon?$$

If H is globally Lipschitz and the s-rule is applied, the AR2 algorithm requires at most

$$\left\lceil \frac{\kappa_{\rm S}}{\epsilon^{3/2}} \right\rceil \text{ evaluations}$$

for some $\kappa_{\rm S}$ independent of ϵ.

"Nesterov & Polyak",
Cartis, Gould, T., 2011, Birgin, Gardenghi, Martinez, Santos, T., 2017

Note: an $O(\epsilon^{-3})$ bound holds for convergence to second-order critical points.

Evaluation complexity: proof (1)

$$f(x_k + s_k) \le T_{f,2}(x_k, s_k) + \frac{L_f}{p}\|s_k\|^3$$

$$\|g(x_k + s_k) - \nabla_s T_{f,2}(x_k, s_k)\| \le L_f\|s_k\|^2$$

Lipschitz continuity of $H(x) = \nabla_x^2 f(x)$

$$\forall k \ge 0 \qquad f(x_k) - T_{f,2}(x_k, s_k) \ge \tfrac{1}{6}\sigma_{\min}\|s_k\|^3$$

$$f(x_k) = m_k(0) \ge m_k(s_k) = T_{f,2}(x_k, s_k) + \tfrac{1}{6}\sigma_k\|s_k\|^3$$

Evaluation complexity: proof (2)

$$\exists \sigma_{\max} \quad \forall k \geq 0 \qquad \sigma_k \leq \sigma_{\max}$$

Assume that $\sigma_k \geq \dfrac{L_f(p+1)}{p\,(1-\eta_2)}$. Then

$$|\rho_k - 1| \leq \frac{|f(x_k+s_k) - T_{f,2}(x_k,s_k)|}{|T_{f,2}(x_k,0) - T_{f,2}(x_k,s_k)|} \leq \frac{L_f(p+1)}{p\,\sigma_k} \leq 1-\eta_2$$

and thus $\rho_k \geq \eta_2$ and $\sigma_{k+1} \leq \sigma_k$.

Evaluation complexity: proof (3)

$$\forall k \ \text{ successful} \qquad \|s_k\| \geq \left(\frac{\|g(x_{k+1})\|}{L_f + \kappa_{\text{stop}} + \sigma_{\max}} \right)^{\frac{1}{2}}$$

$$\begin{aligned}
\|g(x_k + s_k)\| &\leq \|g(x_k + s_k) - \nabla_s T_{f,2}(x_k, s_k)\| \\
&\quad + \left\| \nabla_s T_{f,2}(x_k, s_k) + \sigma_k \|s_k\| s_k \right\| + \sigma_k \|s_k\|^2 \\
&\leq L_f \|s_k\|^2 + \|\nabla_s m(s_k)\| + \sigma_k \|s_k\|^2 \\
&\leq [L_f + \kappa_{\text{stop}} + \sigma_k] \, \|s_k\|^2
\end{aligned}$$

Evaluation complexity: proof (4)

$$\|g(x_{k+1})\| \leq \epsilon \text{ after at most } \frac{f(x_0) - f_{\rm low}}{\kappa}\, \epsilon^{-3/2} \text{ successful iterations}$$

Let $\mathcal{S}_k = \{j \leq k \geq 0 \mid \text{iteration } j \text{ is successful}\}$.

$$\begin{aligned}
f(x_0) - f_{\rm low} &\geq f(x_0) - f(x_{k+1}) \geq \textstyle\sum_{i\in\mathcal{S}_k} \Big[f(x_i) - f(x_i + s_i) \Big] \\
&\geq \tfrac{1}{10} \sum_{i\in\mathcal{S}_k} \Big[f(x_i) - T_{f,2}(x_i, s_i) \Big] \geq |\mathcal{S}_k| \frac{\sigma_{\min}}{60} \min_i \|s_i\|^3 \\
&\geq |\mathcal{S}_k| \frac{\sigma_{\min}}{60\Big(L_f + \kappa_{\rm stop} + \sigma_{\max}\Big)^{3/2}} \min_i \|g(x_{i+1})\|^{3/2} \\
&\geq |\mathcal{S}_k| \frac{\sigma_{\min}}{60\Big(L_f + \kappa_{\rm stop} + \sigma_{\max}\Big)^{3/2}} \epsilon^{3/2}
\end{aligned}$$

Evaluation complexity: proof (5)

$$k \leq \kappa_u |\mathcal{S}_k|, \quad \text{where} \quad \kappa_u \stackrel{\rm def}{=} \left(1 + \frac{|\log \gamma_1|}{\log \gamma_2}\right) + \frac{1}{\log \gamma_2} \log\left(\frac{\sigma_{\max}}{\sigma_0}\right),$$

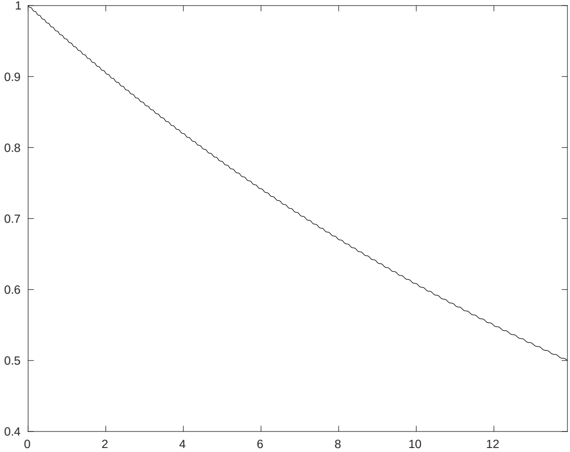
$\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$ + mechanism of the σ_k update.

$\|g(x_{k+1})\| \leq \epsilon$ after at most $\dfrac{f(x_0) - f_{\rm low}}{\kappa}\, \epsilon^{-3/2}$ successful iterations

One evaluation per iteration (successful or unsuccessful).

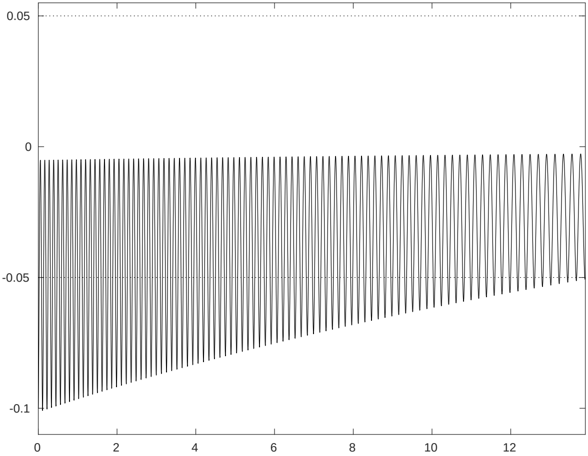
Evaluation complexity: sharpness

Is the bound in $O(\epsilon^{-3/2})$ sharp? YES!!!



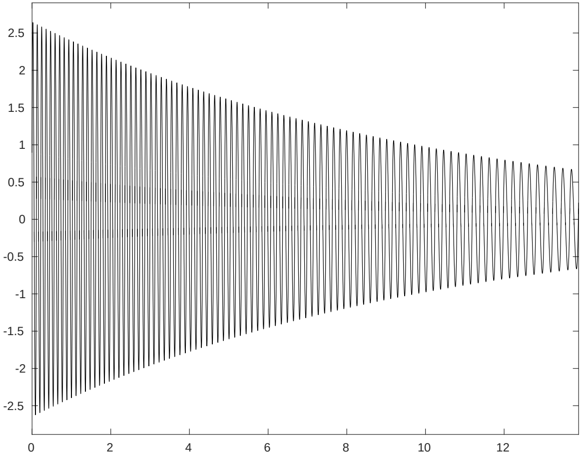
The objective function

An example of slow AR2 (2)



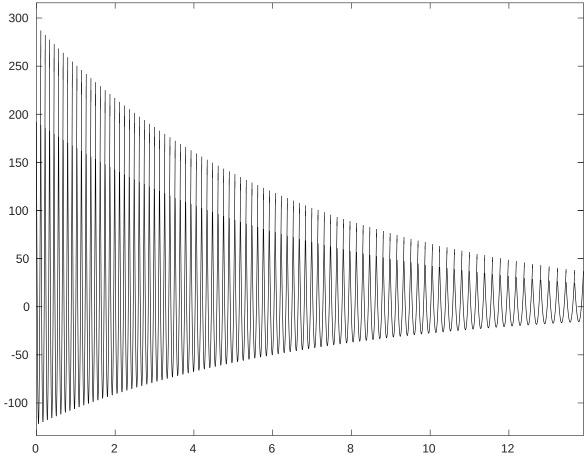
The first derivative

An example of slow AR2 (3)



The second derivative

An example of slow AR2 (4)



The third derivative

Slow steepest descent (1)

The steepest descent method with requires at most

$$\left\lceil \frac{\kappa_{\rm C}}{\epsilon^2} \right\rceil \text{ evaluations}$$

for obtaining $\|g_k\| \leq \epsilon$.

Nesterov

Sharp??? YES

Newton's method (when convergent) requires at most

$$O(\epsilon^{-2}) \text{ evaluations}$$

for obtaining $\|g_k\| \leq \epsilon$!!!!

High-order models for first-order points (1)

What happens if one considers the model

$$m_k(s) = T_{f,p}(x_k, s) + \frac{\sigma_k}{p!}\|s\|_2^{p+1}$$

where

$$T_{f,p}(x,s) = f(x) + \sum_{j=1}^{p} \frac{1}{j!}\nabla_x^j f(x)[s]^j$$

terminating the step computation when

$$\|\nabla_s m(s_k)\| \le \kappa_{\rm stop}\|s_k\|^p$$

now the first-order ARp method!

High-order models for first-order points (2)

unconstrained ϵ-approximate 1rst-order-necessary minimizer after at most

$$\frac{f(x_0) - f_{\rm low}}{\kappa} \, \epsilon^{-\frac{p+1}{p}}$$

function and gradient evaluations

Birgin, Gardhenghi, Martinez, Santos, T., 2017

Technique of proof very similar to that used above.

Derivative tensors for partially separable problems

f is partially separable if

$$f(x) = \sum_{i=1}^{m} f_i(U_i x) = \sum_{i=1}^{m} f_i(x_i) \quad \text{where} \quad \mathrm{rank}(U_i) \ll n$$

Then

$$\nabla_x^p f(x)[s]^p = \sum_{i=1}^{m} \nabla_{x_i}^p f_i(x)[U_i x]^p$$

Note:

$$\text{size}(\ \nabla_{x_i}^p f_i(x)\ ) \ll \text{size}(\ \nabla_x^p f(x)\ )!!!$$

One then wonders. . .

If one uses a model of degree p ($T_{f,p}(x,s)$), why be satisfied with first- or second-order critical points???

What do we mean by critical points of order larger than 2 ???

What are necessary optimality conditions for order larger than 2 ???

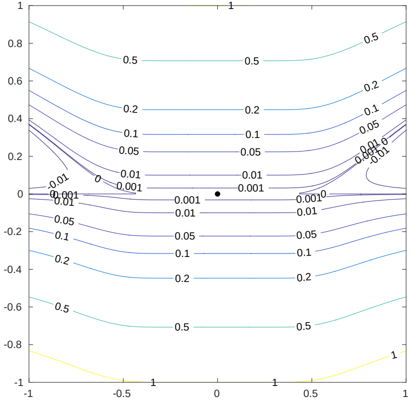
Not an obvious question!

A sobering example (1)

Consider the unconstrained minimization of

$$f(x_1, x_2) = \begin{cases} x_2\left(x_2 - e^{-1/x_1^2}\right) & \text{if } x_1 \neq 0, \\ x_2^2 & \text{if } x_1 = 0, \end{cases}$$

Peano (1884), Hancock (1917)



A sobering example (2)

Conclusions:

- looking at optimality along straight lines is not enough
- depending on Taylor's expansion for necessary conditions is not always possible

Even worse:

$$f(x_1, x_2) = \begin{cases} x_2 \left(x_2 - \sin(1/x_1) e^{-1/x_1^2} \right) & \text{if } x_1 \neq 0, \\ x_2^2 & \text{if } x_1 = 0, \end{cases}$$

(no continuous descent path from 0, although not a local minimizer!!!)

Hopeless?

A new (approximate) optimality measure

Define, for some small $\delta > 0$, $(\mathcal{F} = \mathbb{R}^n)$

$$\phi^{\delta}_{f,q}(x) \stackrel{\rm def}{=} f(x) - \text{globmin}_{\substack{x+d\in\mathcal{F}\\ \|d\|\leq\delta}} T_{f,q}(x,d),$$

and

$$\chi_q(\delta) \stackrel{\rm def}{=} \sum_{\ell=1}^{q} \frac{\delta^\ell}{\ell!}$$

x is a (ϵ,δ)-approximate qth-order-necessary minimizer

$$\Longleftrightarrow$$

$$\phi^{\delta}_{f,q}(x) \leq \epsilon\,\chi_q(\delta)$$

- $\phi^{\delta}_{f,q}(x)$ is continuous as a function of x for all q.
- $\phi^{\delta}_{f,q}(x) = o\big(\chi_q(\delta)\big)$ is a necessary optimality condition

Approximate unconstrained optimality

Familiar results for low orders: when $q = 1$

$$\left.\begin{array}{c} \phi_{f,1}^{\delta}(x) = \|\nabla_x f(x)\|\, \delta \\ \chi_1(\delta) = \delta \end{array}\right\} \Rightarrow \|\nabla_x f(x)\| \le \epsilon$$

while, for $q = 2$,

$$\left.\begin{array}{c} \|\nabla_x f(x)\| \le \epsilon \\ \lambda_{\min}(\nabla_x^2 f(x)) \ge -\epsilon \end{array}\right\} \Rightarrow \phi_{f,2}^{\delta}(x) \le \epsilon \chi_2(\delta)$$

Suppose that $\nabla_x^q f$ is β-Hölder continous near x_ϵ and that

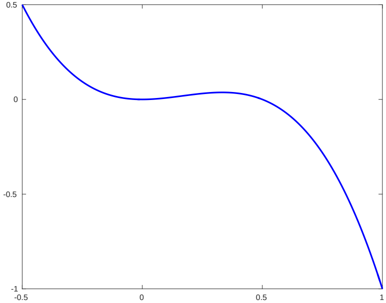
$$\phi_{f,q}^{\delta}(x_\epsilon) \le \epsilon \chi_q(\delta).$$

Then

$$f(x_\epsilon + d) \ge f(x_\epsilon) - 2\epsilon\chi_q(\delta) \quad \forall d \mid \|d\| \le \min\left[\delta, \left(\frac{(q+1)!\,\epsilon}{L_{f,q}}\right)^{\frac{1}{q-1+\beta}}\right]$$

The need for δ

Let $x = 0$ and $T(x, s) = s^2 - 2s^3$



Then

- the origin is a local minimizer of T
- $\phi^1_{T,3}(1) = -1 \neq 0$ but $\phi^\delta_{T,3}(x) = 0$ for all $\delta \leq 4/7$.

Introducing inexpensive constraints

Constraints are inexpensive

$\Leftrightarrow$

their evaluation/enforcement has negligible cost
(compared with that of evaluating f)

- evaluation complexity for the constrained problem well measured in counting evaluations of f and its derivatives
- many well-known and important examples
  - bound constraints
  - convex constraints with cheap projections
  - parametric constraints
  - ...

From now on: $\mathcal{F} \stackrel{\text{def}}{=}$ (inexpensive) feasible set

A very general optimization problem

Our aim:

Compute an (ϵ, δ)-approximate qth-order-necessary minimizer for the problem

$$\min_{x \in \mathcal{F}} f(x)$$

where

- $p \geq q \geq 1$,
- $\nabla_x^p f(x)$ is β-Hölder continuous ($\beta \in (0, 1]$)
- $\mathcal{F}$ is an inexpensive feasible set

Note:

1. no convexity assumption of f
2. no convexity assumption on $\mathcal{F}$ (not even connectivity)
3. reduces to Lipschitz continuous $\nabla_x^p f(x)$ when $\beta = 1$.

A (theoretical) regularization algorithm

Algorithm 3.1: The ARp algorithm for qth-order optimality

Step 0: Initialization: x_0, δ_{-1} and $\sigma_0 > 0$ given. Set $k = 0$

Step 1: Termination: If $\phi_{f,q}^{\delta_{k-1}}(x_k) \le \epsilon \chi_q(\delta)$, terminate.

Step 2: Step computation:

Compute* s_k such that $x_k + s_k \in \mathcal{F}$, $m_k(s_k) < m_k(0)$ and

$$\|s_k\| \ge \kappa_s \, \epsilon^{\frac{1}{p-q+\beta}} \quad \text{or} \quad \phi_{m_k,q}^{\delta_k}(x_k + s_k) \le \frac{\theta \, \|s_k\|^{p-q+\beta}}{(p-q+\beta)!} \chi_q(\delta_k)$$

Step 3: Step acceptance:

Compute $\rho_k = \dfrac{f(x_k) - f(x_k + s_k)}{f(x_k) - T_{f,p}(x_k, s_k)}$

and set $x_{k+1} = x_k + s_k$ if $\rho_k > 0.1$ or $x_{k+1} = x_k$ otherwise.

Step 4: Update the regularization parameter:

$$\sigma_{k+1} \in \begin{cases} [\sigma_{\min}, \sigma_k] & = \frac{1}{2}\sigma_k \quad \text{if } \rho_k > 0.9 & \textit{very successful} \\ [\sigma_k, \gamma_1 \sigma_k] & = \sigma_k \quad \text{if } 0.1 \le \rho_k \le 0.9 & \textit{successful} \\ [\gamma_1 \sigma_k, \gamma_2 \sigma_k] & = 2\sigma_k \quad \text{otherwise} & \textit{unsuccessful} \end{cases}$$

Finding a step

Compute*: does a suitable step always exists?

Either

$$\operatorname*{globmin}_{x_k+s\in\mathcal{F}} m_k(s) = 0$$

or there exists $\delta_k \in (0,1]$ and a neighbourhood of

$$s_k^* = \arg \operatorname*{globmin}_{x_k+s\in\mathcal{F}} m_k(s)$$

such that, for all s in that neighbourhood

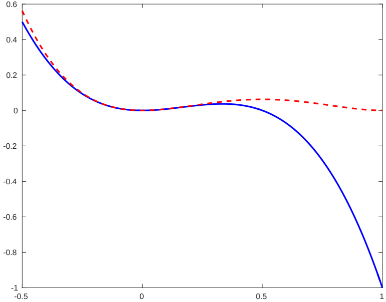
$$m_k(s) < m_k(0) \quad \text{and} \quad \phi_{m_k,q}^{\delta_k}(x_k+s) \le \epsilon\chi_q(\delta_k).$$

Note: (ϵ,δ)-approximate pth-order-necessary minimizer in the first case!

Need for the first case

Let $x = 0$, $T(x,s) = s^2 - 2s^3$ (as above) and $\sigma_k = 24$, yielding

$$m(s) = s^2 - 2s^3 + s^4 = s^2(s-1)^2 \geq 0$$



Further comments on the algorithm

1. when $\|s_k\| \geq \kappa_s \, \epsilon^{\frac{1}{p-q+\beta}}$, no need for computing $\phi^{\delta_k}_{m_k,q}(x_k + s_k)$!
2. for $p = 1$ and $p = 2$, computing it is easy
   - $p = 1$: analytic solution
   - $p = 2$: trust-region subproblem with unit radius

   $\Rightarrow$ practical algorithm
3. for $p > 2$: hard problem in general

   $\Rightarrow$ conceptual algorithm

The main result

The ARp algorithm finds an (ϵ, δ)-approximate qth-order-necessary minimizer for the problem

$$\min_{x \in \mathcal{F}} f(x)$$

in at most

$$O\left(\epsilon^{-\frac{p+\beta}{p-q+\beta}}\right)$$

iterations and evaluations of the objective function and its p first derivatives. Moreover, this bound is sharp.

What this theorem does

1. generalizes ALL known complexity results for regularization methods to

   arbitrary degree p, arbitrary order q and arbitrary smoothness $p+\beta$

2. applies to very general constrained problems
3. generalizes the lower complexity bound of Carmon at al., 2018, to arbitrary dimension, arbitrary order and to constrained problems
4. provides a considerably better complexity order than the bound

   $$O\left(\epsilon^{-(q+1)}\right)$$

   known for unconstrained trust-region algorithms (Cartis, Gould, T., 2017)

   Note: linesearch methods all fail for $q > 3$!
5. is provably optimal within a wide class of algorithms (Cartis, Gould, T., 2018 for $p \leq 2$)

A slide from the ICM in August 2018...

Where do we stand (for convexly constrained problems)?

$\vdots$	–	–	–	–		**?**
q	–	–	–	$O(\epsilon^{-(q+1)})$	**?**	**?**
$\vdots$	–	–		**?**	**?**	**?**
2		$O(\epsilon^{-3})$	$\cdots$	$\cdots$	$[O(\epsilon^{-(p+1)/(p-1)})]$	$\cdots$
1	$O(\epsilon^{-2})$	$O(\epsilon^{-3/2})$	$\cdots$	$\cdots$	$O(\epsilon^{-(p+1)/p})$	$\cdots$
$\uparrow q/p \rightarrow$	1	2	$\cdots$	$\cdots$	p	$\cdots$
	$\leftarrow$ *sharp* $\rightarrow$					

Complexity of optimality order q as a function of model degree p

Trust-region algo   Regularization algo

[ ] for unconstrained problems only!

Moving on: allowing inexact evaluations

A common observation:

In many applications, it is necessary/useful to evaluate $f(x)$ and/or $\nabla_x^j f(x)$ inexactly

1. complicated computations involving truncated iterative processes
2. variable accuracy schemes
3. sampling techniques (machine learning)
4. noise
5. ...

Focus on the case where f and all its derivatives are inexact

The dynamic accuracy framework (1)

How are the values of $f(x)$ and $\nabla_x^j f(x)$ used in the ARp algorithm?

- $f(x_k)$ and $f(x_k+s_k)$ are used in order to accept/reject the step when computing

$$\rho_k = \frac{f(x_k)-f(x_k+s_k)}{f(x_k)-T_{f,p}(x_k,s_k)} = \frac{f(x_k)-f(x_k+s_k)}{\Delta T_{f,p}(x_k,s_k)}$$

where

$$\Delta T_{f,p}(x_k,s_k) = f(x_k)-T_{f,p}(x_k,s_k) = -\sum_{\ell=1}^{p} \nabla_x^p f(x_k)[s_k]^p$$

is the Taylor's increment

$\Delta T_{f,p}(x_k,s_k)$ is independent of $f(x_k)$

Hence we need

Absolute error in $f(x_k)$ and $f(x_k+s_k)$ $'' \leq ''$ $\Delta T_{f,p}(x_k,s_k)$

The dynamic accuracy framework (2)

- $\nabla_x^j f(x_k)$ used in
  - computing

$$\begin{aligned}\phi_{f,q}^{\delta_{k-1}}(x_k) &= \min\left\{0, \operatorname*{globmin}_{\substack{x_k+d\in\mathcal{F}\\ \|d\|\le\delta}} [f(x_k) - T_{f,q}(x_k,d)]\right\}\\ &= \max\left\{0, \operatorname*{globmax}_{\substack{x_k+d\in\mathcal{F}\\ \|d\|\le\delta}} \Delta T_{f,q}(x_k,d)\right\}\end{aligned}$$

  - defining the model $m_k(s)$ which is minimized to compute s_k, i.e.

$$\max_{x_k+s\in\mathcal{F}} \Delta T_{f,p}(x_k,s)$$

  - computing

$$\phi_{f,q}^{\delta_{k-1}}(x_k) = \max\left\{0, \operatorname*{globmax}_{\substack{x_k+d\in\mathcal{F}\\ \|d\|\le\delta}} \Delta T_{m_k,q}(x_k,d)\right\}$$

Relative error in $\Delta T_{\bullet,\bullet} < 1$

The dynamic accuracy framework (3)

Denote inexact quantities with overbars.

Note: $\overline{\Delta T}_{\bullet,\bullet} \geq 0$

Accuracy conditions ($\kappa_1, \kappa_2 \in [0,1)$):

$$\max\left[|\overline{f}(x_k) - f(x_k)|, |\overline{f}(x_k + s_k) - f(x_k)|\right] \leq \kappa_1 \overline{\Delta T}_{f,p}(x_k, s_k)$$

$$|\overline{\Delta T}_{\bullet,\bullet} - \Delta T_{\bullet,\bullet}| \leq \kappa_2 \overline{\Delta T}_{\bullet,\bullet}$$

The latter relative error bound can be obtained by

iteratively decreasing the absolute error until satisfied

Only impose absolute error levels ε on $\{\nabla_x^j f(x_k)\}_{j=0}^p$

The ARpDA algorithm

Algorithm 4.1: The ARpDA algorithm for qth-order optimality

Step 0: Initialization: x_0, δ_{-1} and $\sigma_0 > 0$ given. Set $k = 0$

Step 1: Termination: If $\overline{\phi}_{f,q}^{\delta_{k-1}}(x_k) \leq \frac{1}{2}\epsilon\chi_q(\delta)$, terminate.

Step 2: Step computation:

Compute* s_k such that $x_k + s_k \in \mathcal{F}$, $m_k(s_k) < m_k(0)$ and

$$\|s_k\| \geq \kappa_s\, \epsilon^{\frac{1}{p-q+\beta}} \quad \text{or} \quad \overline{\phi}_{m_k,q}^{\delta_k}(x_k + s_k) \leq \frac{\theta\, \|s_k\|^{p-q+\beta}}{(p-q+\beta)!}\chi_q(\delta_k)$$

Step 3: Step acceptance:

$$\text{Compute } \rho_k = \frac{\overline{f}(x_k) - \overline{f}(x_k + s_k)}{\overline{\Delta T}_{f,p}(x_k, s_k)}$$

and set $x_{k+1} = x_k + s_k$ if $\rho_k > 0.1$ or $x_{k+1} = x_k$ otherwise.

Step 4: Update the regularization parameter:

(as in ARp)

Evaluation complexity for the ARpDA algorithm

And then (sweeping some dust under the carpet)...

The ARpDA algorithm finds an (ϵ, δ)-approximate qth-order-necessary minimizer for the problem

$$\min_{x \in \mathcal{F}} f(x)$$

in at most

$$O\left(\epsilon^{-\frac{p+\beta}{p-q+\beta}}\right)$$

iterations and at most

$$O\left(|\log(\epsilon)|\epsilon^{-\frac{p+\beta}{p-q+\beta}}\right)$$

(inexact) evaluations of the objective function and its p first derivatives.

A probabilistic complexity bound

Suppose that absolute evaluation errors are random and independent, and that, for given ε,

$$Pr\left[\|\,\overline{\nabla_x^j f}\,(x_k) - \nabla_x^j f(x_k)\| \leq \varepsilon\right] \geq 1 - t \quad (j \in \{1, \ldots, p\})$$

where

$$t = O\left(\frac{t_{\rm final}\,\epsilon^{\frac{p+1}{p-q+\beta}}}{p+q+2}\right)$$

Then the ARpDA algorithm finds an (ϵ,δ)-approximate qth-order-necessary minimizer for the problem $\min_{x\in\mathcal{F}} f(x)$ in at most $O\left(\epsilon^{-\frac{p+\beta}{p-q+\beta}}\right)$ iterations and at most $O\left(|\log(\epsilon)|\epsilon^{-\frac{p+\beta}{p-q+\beta}}\right)$ (inexact) evaluations of the objective function and its p first derivatives, with probability $1 - t_{\rm final}$.

Selecting a sample size in subsampling methods (1)

Now consider $p = 2,\ \beta = 1,\ \mathcal{F} = \mathbb{R}^n$ and (as in machine learning)

$$f(x) = \frac{1}{N}\sum_{i=1}^{N}\psi_i(x)$$

Estimating the values of $\{\nabla_x^j f(x_k)\}_{j=0}^{2}$ by sampling:

$$\overline{f}\,(x_k) = \frac{1}{|\mathcal{D}_k|}\sum_{i\in\mathcal{D}_k}\psi_i(x_k), \quad \overline{\nabla_x^1 f}\,(x_k) = \frac{1}{|\mathcal{G}_k|}\sum_{i\in\mathcal{G}_k}\nabla_x^1\psi_i(x_k),$$

$$\overline{\nabla_x^2 f}\,(x_k) = \frac{1}{|\mathcal{H}_k|}\sum_{i\in\mathcal{H}_k}\nabla_x^2\psi_i(x_k),$$

and applying the Operator-Bernstein matrix concentration inequality. . .

Selecting a sample size in subsampling methods (2)

Suppose that $\beta = 1 \le q \le 2 = p$, that, for all k and $j \in \{0,1,2\}$,

$$\max_{i \in \{1,\ldots,N\}} \|\nabla_x^j \psi_i(x_k)\| \le \kappa_j(x_k)$$

and that, for given ε,

$$|\mathcal{D}_k| \ge \vartheta_{0,k}(\varepsilon) \log\left(2/t\right), \quad |\mathcal{G}_k| \ge \vartheta_{1,k}(\varepsilon) \log\left((n+1)/t\right),$$

$$|\mathcal{H}_k| \ge \vartheta_{2,k}(\varepsilon) \log\left(2n/t\right),$$

where

$$\vartheta_{j,k}(\varepsilon) \stackrel{\rm def}{=} \frac{4\kappa_j(x_k)}{\varepsilon}\left(\frac{2\kappa_j(x_k)}{\varepsilon} + \frac{1}{3}\right) \quad \text{and} \quad t = O\left(\frac{t_{\rm final}\, \epsilon^{\frac{3}{3-q}}}{4+q}\right).$$

Then the AR2DA algorithm finds an ϵ-approximate qth-order-necessary minimizer for the problem $\min_{x \in \mathbb{R}^n} f(x)$ in at most $O\left(\epsilon^{-\frac{3}{3-q}}\right)$ iterations and at most $O\left(|\log(\epsilon)|\epsilon^{-\frac{3}{3-q}}\right)$ subsampled evaluations of f, $\nabla_x^1 f$ and $\nabla_x^2 f$, with probability $1 - t_{\rm final}$.

Conclusions

Evaluation complexity for qth order approximate minimizers using degree p models for β-Hölder continuous $\nabla_x^p f$

$$O\left(\epsilon^{-\frac{p+\beta}{p-q+\beta}}\right) \quad \text{(unconstrained, inexpensive constraints)}$$

This bound is sharp!

Extension to inexact evaluations:

$$O\left(|\log(\epsilon)|\epsilon^{-\frac{p+\beta}{p-q+\beta}}\right) \quad \text{(unconstrained, inexpensive constraints)}$$

Consequences in probabilistic complexity and subsampling strategies

Perspectives

Complexity for expensive constraints for $q > 1$?

Subsampling of derivative tensors

Optimization in variable arithmetic precision

etc., etc., etc.

Thank you for your attention!

Some references

C. Cartis, N. Gould and Ph. L. Toint,
"Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints",
arXiv:1811.01220.

S. Bellavia, G. Gurioli, B. Morini and Ph. L. Toint,
"Deterministic and stochastic inexact regularization algorithms for nonconvex optimization with optimal complexity",
arXiv:1811.03831.

C. Cartis, N. Gould and Ph. L. Toint,
"Second-order optimality and beyond: characterization and evaluation complexity in convexly-constrained nonlinear optimization",
FoCM, to appear.

Also see http://perso.fundp.ac.be/~phtoint/toint.html