Recent results in worst-case evaluation complexity for smooth and non-smooth, exact and inexact, nonconvex optimization

Philippe Toint

(with S. Bellavia, C. Cartis, X. Chen, N. Gould, S. Gratton, G. Gurioli, B. Morini and E.Simon)

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

(philippe.toint@unamur.be)

ICCOPT 2019, Berlin

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

Adaptive regularization

Adaptive regularization methods iteratively compute steps by mimizing

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

until an approximate first-order minimizer is obtained:

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

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

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

for some κ_S independent of ϵ.

"Nesterov & Polyak",

Cartis, Gould, T., 2011, Birgin, Gardenghi, Martinez, Santos, T., 2017

Note:

- The above result is sharp (in order of ϵ)!
- An $O(\epsilon^{-3})$ bound holds for convergence to second-order critical points.

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

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

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

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 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 weak (ϵ,δ)-approximate qth-order-necessary minimizer

$$\Leftrightarrow$$

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

- $\phi^{\delta}_{f,q}(x)$ 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 $q = 1$

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

while, for $q = 2$,

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

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: $\mathcal{F} \stackrel{\rm def}{=}$ (inexpensive) feasible set

A very general optimization problem

Our aim:

Compute an weak (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 2.1: The ARqp 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:

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

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:

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

The main result

The ARp algorithm is well-defined and

The ARp algorithm finds a strong (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.

Same complexity for achieving the strong optimality condition

$$\phi_{f,j}^{\delta_j}(x) \le \epsilon_j \frac{\delta_j^j}{j!} \quad j \in \{1, \ldots, q\}$$

under stronger smoothness assumptions and $p \le 2q$.

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 $p \leq 2$)

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 3.1: 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 a strong (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 (inexact) evaluations of the objective function, and at most

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

(inexact) evaluations of 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 a strong (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 (inexact) evaluations of the objective function, and at most $O\left(|\log(\epsilon)| + \epsilon^{-\frac{p+\beta}{p-q+\beta}}\right)$ (inexact) evaluations of 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 $\beta = 1 \leq q \leq 2 = p$, that, for all k and $j \in \{0,1,2\}$,

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

and that, for given ε,

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

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

where

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

Then the AR2DA algorithm finds a strong 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 subsampled evaluations of f, and at most $O\left(|\log(\epsilon)| + \epsilon^{-\frac{3}{3-q}}\right)$ subsampled evaluations $\nabla_x^1 f$ and $\nabla_x^2 f$, with probability $1 - t_{\text{final}}$.

Turning to non-smooth problems: non-Lipschitzian singularities 1

Now consider

$$\min_{x\in\mathcal{F}} f(x) + \sum_{i\in\mathcal{H}} |x_i|^a, \qquad a \in (0,1)$$

with $\mathcal{F}$ convex and "kernel centered"
Define

$$\mathcal{C}(x) = \{ i \in \mathcal{H} \mid x_i = 0 \} \;\text{ and }\; \mathcal{R}(x) = \bigcap_{i\in\mathcal{H}\setminus\mathcal{R}(x)} \text{span}\,\{e_i\}$$

Criticality measure

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

Non-Lipschitzian singularities 2

- define a Lipschitzian model of the non-Lipschitzian singularities based on inherent symmetry
- prove that the related Lipschitz constant is independent of ϵ
- assemble the singular and non-singular complexity estimates

$O\left(\epsilon^{-\frac{p+\beta}{p-q+\beta}}\right)$ evaluations of f and its derivatives

Non-smooth Lipschitzian composite problems

Finally, consider

$$\min_x f(x) + h(c(x))$$

where f and c have Lipschitz gradients but are inexact, and h is convex, Lispchitz and exact.

- not a special case of smooth inexact case because $\overline{\Delta f}$ now involves h as well as $\overline{\nabla_x^1 f}$ and $\overline{\nabla_x^1 c}$
- simpler termination for step computation possible

$O\left(|\log(\epsilon)| + \epsilon^{-2}\right)$ evaluations of f, h, c, $\nabla_x^1 f$ and $\nabla_x^1 c$

Also for problems with inexpensive constraints

Conclusions 1

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!

Also valid for a class of function with non-Lipschitz singularities

Conclusions 2

Allows partially-separable structure within the objective function

Extension to inexact evaluations for smooth problems:

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

Extension to inexact evaluations for non-smooth Lispchitzian composite problems:

$$O\left(|\log(\epsilon)| + \epsilon^{-2}\right) \text{ (unconstrained, inexpensive constraints)}$$

Conclusions 3

Consequences in probabilistic complexity and subsampling strategies

Other results available for first-order optimality in problems with expensive constraints

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", SIOPT, to appear, 2019.

C. Cartis, N. Gould and Ph. L. Toint,
"Second-order optimality and beyond: characterization and evaluation complexity in convexly-constrained nonlinear optimization", FoCM, vol. 18(5), pp. 1083-1107, 2018.

X. Chen, Ph. L. Toint and H. Wang,
"'"Partially separable convexly-constrained optimization with non-Lipschitzian singularities and its complexity", SIOPT, to appear, 2019.

S. Gratton, E. Simon and Ph. L. Toint,
"Minimization of nonsmooth nonconvex functions using inexact evaluations and its worst-case complexity", arXiv:1902.10406.

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