Second-order methods for nonconvex optimization with complexity guarantees

Stephen Wright (UW-Madison)

September, 2020

Outline

- **0.** Introduction and Motivation
- **1.** Unconstrained Optimization
  - **1a.** Line-Search
  - **1b.** Trust-Region
  - **1c.** Strict Saddles
- **2.** Bound-constrained Optimization
  - **2a.** Log-Barrier
  - **2b.** Projected Newton-CG
- **3.** Equality constrained Optimization
  - **3a.** Proximal-Augmented-Lagrangian

Collaborators

Clément Royer (Université Paris-Dauphine)
Mike O'Neill (Wisconsin $\rightarrow$ Lehigh)
Yue Xie (Wisconsin)
Frank Curtis (Lehigh)
Daniel Robinson (Lehigh)

0. Complexity in Optimization

Complexity: Finding bounds on the amount of computation for an algorithm and / or problems in a certain class.

Computation can be measured in several ways, including

- Iterations (sometimes broken down into different levels of iteration: outer and inner);
- Oracle: how many queries for information about the functions are required to find an approximate solution. Sometimes separated into function evaluations, first-derivative, second-derivative information.

In this talk, we deal with upper bounds on the computation required by a given algorithm for all problems in a certain class.

(There's also interest in lower bounds.)

Complexity in Optimization: Nemirovski and Yudin

[Nemirovski and Yudin, 1983] is a founding document in complexity of continuous optimization problems and algorithms. Identified 3 ingredients:

- class of problems
- sources of information about the problem e.g. values of $f(x)$ and $\nabla f(x)$ for a given x: oracles (attributed to Bakhvalov)
- methods for definining error / solution accuracy.

Discusses different types of complexity:

- iterations ("laboriousness")
- elementary operations
- memory.

Complexity $N(\epsilon)$ depends on (relative) error ϵ in approximate solution.

Convex only: general, strongly convex, mirror descent, nonlinear conjugate gradient...

(They comment that global minimization of nonconvex functions is exponential in the dimension.)

Complexity in Optimization: Nonlinear

Complexity results in convex optimization.

- polynomial interior-point for LP, convex QP, feasible sets with self-concordant barriers (80s-90s).
- momentum methods for nonlinear convex (heavy ball, Nesterov): faster rates than steepest descent (80s, then 2010-)
- subgradient and stochastic subgradient: convergence rates for averaged iterates.

Interest in complexity for nonconvex optimization is more recent, and focuses on finding approximate second-order points (or higher-order), rather than global solutions.

- Enhances the theory, possibly the practice too.
- Nonconvex applications from machine learning (e.g. matrix optimization) have nice properties such as
  - all saddle points are strict, or
  - all local minima are global.

Philosophy

- There's a rich collection of practical algorithms for nonlinear nonconvex optimization — unconstrained, simple constraints, nonlinear constraints.
- Typical convergence theory is about *local convergence rates* or *accumulation points are stationary* or perhaps *accumulation points satisfy second-order necessary conditions*.
- Can we build on these algorithms, modifying to equip them with global complexity properties without sacrificing practical appeal?

Interested mostly in algorithms that find approximate second-order points but that don't require evaluation of second-derivative information (Hessians) explicitly.

Then can use computational differentiation to obtain Hessian-vector products, based on code for first derivatives: Apply computational differentiation to $\nabla f(x)^T d$ to get $\nabla^2 f(x) d$, for a given d.

1. Smooth Nonconvex Unconstrained Optimization

$\min_{x \in \mathbb{R}^n} f(x)$ where f is smooth and nonconvex.

Seek a second-order necessary (2oN) point:

$$\nabla f(x) = 0, \quad \nabla^2 f(x) \succeq 0.$$

Let $\mathcal{D}$ be an open set containing level set $\{x \mid f(x) \leq f(x^0)\}$. Assume

- f is bounded below: $f(x) \geq f_{\text{low}}$ for all x.
- Gradient and Hessian are Lipschitz continuous: For all $y, z \in \mathcal{D}$, have

$$\|\nabla f(y) - \nabla f(z)\| \leq L_g \|y - z\|, \quad \|\nabla^2 f(y) - \nabla^2 f(z)\| \leq L_H \|y - z\|.$$

At any x, have quadratic and cubic upper bounds on f over all $\mathcal{D}$:

$$f(x+p) \leq f(x) + \nabla f(x)^T p + \frac{L_g}{2}\|p\|^2,$$
$$f(x+p) \leq f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x) p + \frac{L_H}{6}\|p\|^3.$$

Approximate 2oN Points & Guarantees

Seek approximate 2oN points satisfying

$$\|\nabla f(x)\| \leq \epsilon_g, \quad \nabla^2 f(x) \succeq -\epsilon_H I,$$

where ϵ_g and ϵ_H are small positive tolerances.

Seek iteration complexities for finding such points. Also seek operation complexities in terms of the number of fundamental operations required. Bound these in terms of ϵ_g and ϵ_H. (Also L_g and L_H.)

We take the "fundamental operations" to be

- gradient evaluations $\nabla f(x)$, and
- Hessian-vector products $\nabla^2 f(x)d$ for arbitrary d.

whose cost is comparable — see earlier discussion about computational differentiation.

Explicit knowledge of $\nabla^2 f(x)$ is not required!

A Basic Algorithm with Pretty Good Complexity

When L_g and L_H are known, there is an elementary steepest-descent + negative curvature method that finds an approximate 2oN point in $O(\max(\epsilon_g^{-2}, \epsilon_H^{-3}))$ iterations.

For $k = 0, 1, 2, \ldots$:

- If $\|\nabla f(x^k)\| > \epsilon_g$, take a short steepest-descent step:

$$x^{k+1} = x^k - \tfrac{1}{L_g} \nabla f(x^k).$$

  Use quadratic upper bound to get a decrease of $\geq \epsilon_g^2/(2L_g)$.

- Otherwise, if $\nabla^2 f(x^k) \not\succeq -\epsilon_H I$, find direction d^k such that

$$\|d^k\| = 1, \quad (d^k)^T \nabla^2 f(x^k) d^k = \lambda_{\min}^k < -\epsilon_H, \quad \nabla f(x^k)^T d^k \leq 0.$$

  Take a step of length $2|\lambda_{\min}^k|/L_H$ along d^k to get decrease of $\geq \frac{2}{3}\epsilon_H^3/L_H^2$, using cubic upper bound.

Iteration and Operation Complexity

Because of the lower bound f_{low}, the number of iterations is at most

$$\max\left(2L_g\epsilon_g^{-2}, \frac{3}{2}L_H^2\epsilon_H^{-3}\right)(f(x^0) - f_{\text{low}}).$$

Each iteration in this scheme requires gradient evaluation and (sometimes) cost of finding the most negative eigenvalue of $\nabla^2 f(x^k)$.

In fact, for the negative curvature direction, need only d such that

$$d^T\nabla^2 f(x^k)d \leq -\tfrac{1}{2}\epsilon_H\|d\|^2.$$

If $\lambda_{\min}(\nabla^2 f(x^k)) \leq -\epsilon_H$, this can be computed to probability $1-\delta$ using randomly-started Lanczos iteration at a cost of

$$\min\left\{n, O\left(\sqrt{\frac{L_g}{\epsilon_H}}|\log\delta|\right)\right\}$$

Hessian-vector products (without necessarily knowing $\nabla^2 f$ explicitly).

Operation complexity is a factor of $\epsilon_H^{-1/2}$ greater than iteration complexity.

Literature on Complexity for Second Order Points

[Nesterov and Polyak, 2006] proposed adding a cubic regularization term to the second-order Taylor-series model, to find approximate second-order points with complexity guarantees:

$$f(x+p) \leq f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x) p + \frac{M}{6} \|p\|^3.$$

(Cubic regularization previously proposed by [Griewank, 1981].)

Many later papers on iteration complexity for second-order points:

- Cubic regularization and trust-region methods
  [Nesterov and Polyak, 2006, Birgin and Martínez, 2017, Cartis et al., 2012, Curtis et al., 2017a, Curtis et al., 2017b, Martínez and Raydan, 2017, Cartis et al., 2019a]
- p-order necessary points ($p \geq 2$)
  [Cartis et al., 2020a, Cartis et al., 2020b]

Algorithms with good operation complexity described in

- Cubic regularization [Agarwal et al., 2017];
- Adapting accelerated gradient in various ways [Carmon et al., 2017a, Carmon et al., 2017b];
- Gradient descent (+ acceleration), with noise injection to escape from saddles [Jin et al., 2017a, Jin et al., 2017b].

1a. Line-Search Newton-CG Procedure

The method in [Royer et al., 2020] uses two kind of directions p_k:

- "sufficient" negative curvature for $\nabla^2 f(x_k)$;
- approximate (slightly) damped Newton $-(\nabla^2 f(x_k) + 2\epsilon_H I)^{-1} \nabla f(x_k)$ (provided it's a descent direction).

Does a backtracking line search along each such direction.

- CG is used to find damped Newton step OR detect and return a direction of "sufficient negative curvature"
- Monitor the CG procedure to ensure that no more than $O(\epsilon_H^{-1/2})$ steps are taken. (Requires some complicated termination tests.)
- As a backup to CG (rarely needed), use randomized Lanczos to search for the "sufficient negative curvature" direction for $\nabla^2 f(x_k)$.

Follows the traditional line-search Newton-CG approach, but the new features yield provable complexity, and exploit negative curvature explicitly.

Modified CG (Modifications in red)

Define

$$\bar{H} := H + 2\epsilon I,\ \kappa := \frac{M + 2\epsilon}{\epsilon},\ \ T := \frac{4\kappa^4}{(1-\sqrt{1-\tau})^2},\ \ \tau := \frac{1}{\sqrt{\kappa}+1};$$

$y_0 \leftarrow 0,\ r_0 \leftarrow g,\ p_0 \leftarrow -g,\ j \leftarrow 0;$
if $p_0^\top \bar{H} p_0 < \epsilon \|p_0\|^2$ **then**
  Set $d = p_0$ and terminate with d_type=NC;
end if
while TRUE **do**
  $\alpha_j \leftarrow r_j^\top r_j / p_j^\top \bar{H} p_j;$
  $y_{j+1} \leftarrow y_j + \alpha_j p_j;$
  $r_{j+1} \leftarrow r_j + \alpha_j \bar{H} p_j;$
  $\beta_{j+1} \leftarrow (r_{j+1}^\top r_{j+1})/(r_j^\top r_j);$
  $p_{j+1} \leftarrow -r_{j+1} + \beta_{j+1} p_j;$
  $j \leftarrow j + 1;$
  Perform Termination Tests;
end while

Modified CG Properties

Four Termination Tests!

- **1.** $\|r_j\| \leq \hat{\zeta}\|r_0\|$: approx damped Newton step.
- **2,3.** Either y_j or p_j has negative curvature, less than $-\epsilon$. Return it!
- **4.** "Residual norm $\|r_j\|$ is decreasing slowly and this is an indication that H has an eigenvalue less than $-\epsilon$." Moreover, a direction of negative curvature can be recovered by combining two of the iterates y_i encountered so far.

Case **4** uses an idea of [Carmon et al., 2017b] for accelerated gradient (but requires significant modification for CG). Based on an earlier argument of Bubeck (2014).

Complexity of Modified CG:

$$\min\left(n, \tilde{\mathcal{O}}\left(\sqrt{\frac{L_g}{\epsilon}}\right)\right).$$

(Similar to accelerated gradient.)

Since we usually set $\epsilon = \epsilon_H = \epsilon_g^{1/2}$, this is $O(\epsilon_g^{-1/4})$.

Minimum Eigenvalue Oracle (MEO)

Inputs: Symmetric $H \in \mathbb{R}^{n \times n}$, scalar M with $\lambda_{\max}(H) \leq M$, and $\epsilon > 0$; Set parameter $\delta \in [0, 1)$;
Outputs: Estimate λ of $\lambda_{\min}(H)$ such that $\lambda \leq -\epsilon/2$, and vector v with $\|v\| = 1$ such that $v^\top H v = \lambda$ OR certificate that $\lambda_{\min}(H) \geq -\epsilon$.

(If the certificate is output, it is false with probability δ.)

Need MEO, as Modified CG alone may not suffice to identify negative curvature directions, e.g. when $\nabla f(x_k) = 0$ (a possible saddle point).

Can be implemented with randomized Lanczos. Theory from [Kuczyński and Woźniakowski, 1992, Kuczyński and Woźniakowski, 1994] shows that this requires $\tilde{\mathcal{O}}((L_g/\epsilon)^{1/2})$ matrix-vector multiplications with H.

Line-Search Newton-CG (Step k)

if $\|\nabla f(x_k)\| > \epsilon_g$ **then**
  Call CG to obtain d and step type;
  if step type = "negative curvature" **then**
    Scale and flip sign of d to get d_k;
  else {step type is "approx damped Newton"}
    $d_k \leftarrow d$;
  end if
else
  Call MEO to output v;
  if MEO certifies that $\lambda_{\min}(\nabla^2 f(x_k)) \geq -\epsilon_H$ **then**
    Terminate;
  else
    Scale and flip sign of v to get d_k; (negative curvature direction)
  end if
end if
Backtrack to find α_k s.t. $f(x_k + \alpha_k d_k) < f(x_k) - \frac{\eta}{6}\alpha_k^3 \|d_k\|^3$;
$x_{k+1} \leftarrow x_k + \alpha_k d_k$;

Operation Complexity Result

If $\epsilon_H = \epsilon_g^{1/2}$, the method finds an approximate 2oN point in $\tilde{\mathcal{O}}(\epsilon_g^{-7/4}|\log\delta|)$ operations (with $\delta =$ probability of failure).

Also, $\tilde{\mathcal{O}}(\epsilon_g^{-7/4})$ operations needed to find a point satisfying approximate 1oN conditions $\|\nabla f(x)\| \leq \epsilon_g$, with no probability of failure: deterministic!

($\tilde{\mathcal{O}}$ hides log factors.)

Independent of dimension n, for large n. (But the constants in $\tilde{\mathcal{O}}$ depend on Lipschitz constants.)

1b. Trust-Region Newton-CG

Trust-region Newton methods for minimizing smooth f solve at k:

$$s^k = \arg \min_{\|s\| \le \delta_k} m_k(s) := \nabla f(x^k)^T s + \tfrac{1}{2} s^T \nabla^2 f(x^k) s,$$

where δ_k is the trust-region (TR) radius.

Define ratio ρ_k of actual to predicted decrease in f:

$$\rho_k := \frac{f(x^k) - f(x^k + s^k)}{m_k(0) - m_k(s^k)}.$$

If $\rho_k \ge \eta$ (for some small positive η), take step $x^{k+1} = x^k + s^k$ and choose $\delta_{k+1} > \delta_k$. Otherwise decrease δ_k and compute a new s^k.

- Line-search methods choose direction first, then steplength.
- Trust-region methods choose steplength bound first, then direction.

Steihaug's method (1980)

Steihaug (1980) applies CG to minimization of model $m_k(s)$.

- Start from $s = 0$;
- If it crosses the TR boundary, stop at the TR boundary and return;
- If negative curvature direction in $\nabla^2 f(x^k)$ is detected, move along that direction to the TR boundary, then return.
- If TR boundary does not interfere, keep iterating to the minimum of m_k. (At most n iterations.)

Properties:

- Popular and practical.
- First step of CG is to the "Cauchy point," which is enough to guarantee overall convergence to a first-order point.
- Each CG step reduces model m_k, and moves further away from 0.
- (No second-order guarantees; method does not move away from a saddle point.)
- No complexity guarantees.

TR Newton-CG: Modifying for Complexity Guarantees

[Curtis et al., 2019]: Keep the spirit of Steihaug's method, but modify to enable convergence guarantees.

- Add regularization term to model function:

$$m_k(s) := \nabla f(x^k)^T s + \tfrac{1}{2} s^T \nabla^2 f(x^k) s + \epsilon_H s^T s.$$

- Use the same CG method as in the line-search method, but with explicit cap on the number of iterations, and modified to stay inside the trust region.
- Add the minimum eigenvalue oracle (MEO) to check explicitly for negative curvature. Such directions are almost always found in CG, so in practice MEO is usually invoked only as a final check, at the last iteration.

Complexity Results

The approach broadly follows the line-search method:

- CG called at every iteration to compute a step s^k — either approx solution to the damped trust-region subproblem OR negative curvature direction.
- MEO called as check when CG does not return a useful result and gradient $\nabla f(x^k)$ is small (typically only at last iteration).

Complexity results are broadly the same as line-search too:

To find a point x with

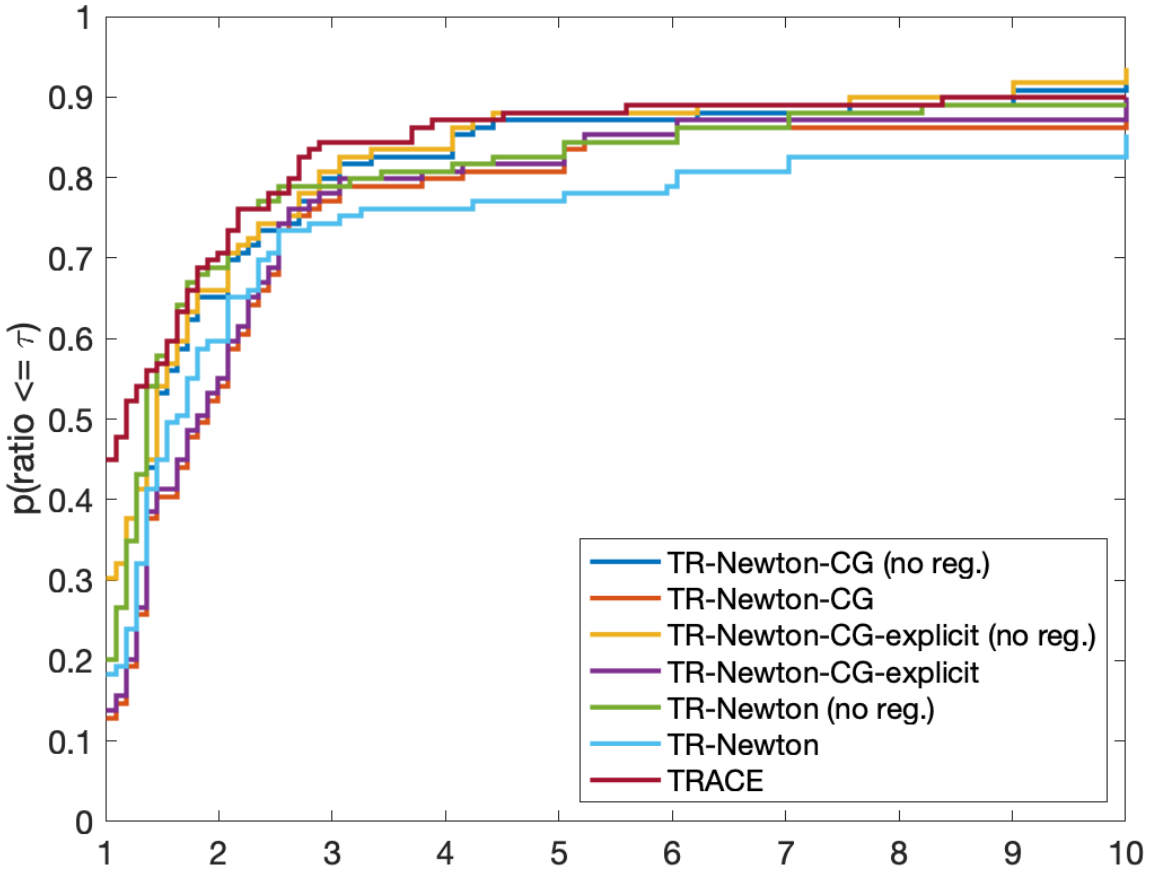
$$\|\nabla f(x)\| \leq \epsilon_g, \quad \nabla^2 f(x) \succeq -\epsilon_H I,$$

with $\epsilon_H = \sqrt{\epsilon_g}$, and with high probability, need $\tilde{\mathcal{O}}(\epsilon_g^{-3/2})$ iterations and $\tilde{\mathcal{O}}(\epsilon_g^{-7/4})$ operations.

Computational Results: Iterations

Problems from the CUTEst set with dimension $n \in [100, 1000]$ All variants solve $\geq 101/109$ problems within $10^4 n$ iterations. $\epsilon_g = 10^{-5}$, $\epsilon_H = 10^{-2.5}$.

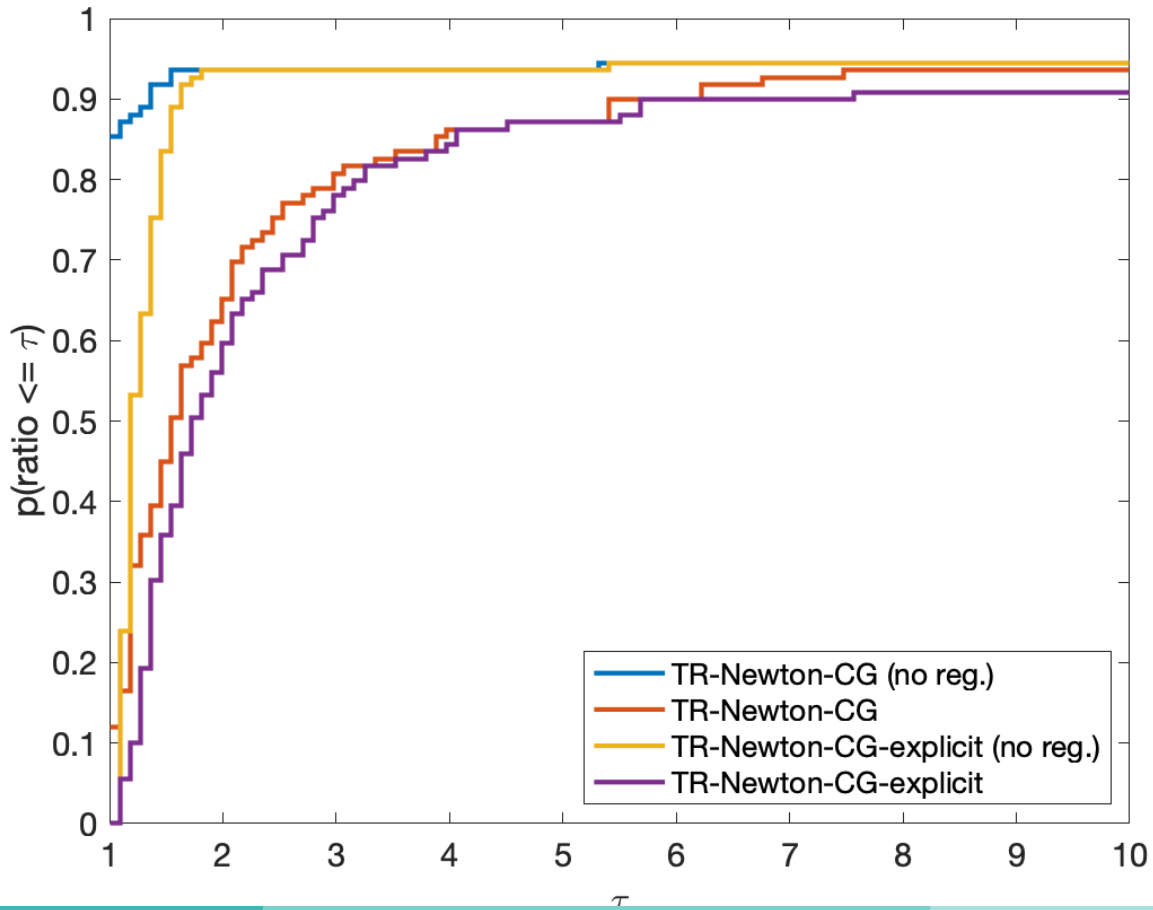
Variants include inexact (CG) and exact subproblems solution. (The analyzed method is TR-Newton-CG-explicit — purple line.)



Computational Results: Hessian-vector products

- Two variants have damped subproblems, consistent with our analysis; the others omit the damping.
- Two variants have explicit caps on CG iterations; the other two don't.

The damped variants use slightly more CG iterations.



1c. Matrix Optimization with Strict Saddles

Low-Rank Matrix Problems: Want to find a rank r solution to the matrix optimization problem:

$$\min_{X \in \mathbb{R}^{n \times m}} f(X), \quad \text{subject to } \mathrm{rank}(X) = r,$$

Matrix completion:

$$\min_X \frac{1}{2} \|(X - X^*)_\Omega\|_F^2, \quad \text{subject to } X \text{ low-rank},$$

where Ω is the set of observed entries and X^*_Ω are given.

Matrix sensing:

$$\min_X \frac{1}{2} \|\mathcal{A}(X - X^*)\|^2, \quad \text{subject to } X \text{ low-rank},$$

where $\mathcal{A} : \mathbb{R}^{n \times m} \to \mathbb{R}^p$ is a known linear measurement operator and $\mathcal{A}(X^*) \in \mathbb{R}^p$ is given.

Reformulation and Strict Saddle Property

Commonly reformulated in terms of $U \in \mathbb{R}^{n\times r}$ and $V \in \mathbb{R}^{m\times r}$:

$$\min_W F(W) := f(UV^T) \quad \text{where} \quad W = \begin{bmatrix} U \\ V \end{bmatrix} \in \mathbb{R}^{(n+m)\times r}.$$

Add a regularization term to eliminate scaling ambiguity:

$$G(W) = F(W) + \tfrac{1}{8}\|U^T U - V^T V\|_F^2.$$

While nonconvex, this formulation satisfies the robust strict saddle property: For any $W \in \mathbb{R}^{(n+m)\times r}$ at least one of the following holds:

1. W is in the neighborhood of a local minimizer W^* and $G(W)$ obeys a regularity condition for all W in this neighborhood (for $\alpha, \beta > 0$):

$$\langle \nabla G(W), W - W^* \rangle \geq \alpha \,\mathrm{dist}(W, W^*)^2 + \beta \|\nabla G(W)\|_F^2.$$

2. $\lambda_{\min}(\nabla^2 G(W)) \leq -\gamma$; or
3. $\|\nabla G(W)\|_F \geq \rho$.

Related Work

We use the global geometry of [Zhu et al., 2017].

Spectral Initialization: Specialized initialization followed by gradient descent. Overall complexity is cost of initialization (e.g. a rank r SVD) plus $\mathcal{O}(\log \max\{\epsilon_g^{-1}, \epsilon_H^{-1}\})$ iterations of gradient descent.

Nonconvex Optimization: Descent method with a saddle point escaping mechanism. When the strict saddle parameters are known at runtime, complexity of gradient descent with occasional perturbations is $\mathcal{O}(\log \max\{\epsilon_g^{-1}, \epsilon_H^{-1}\})$.

Issue: Strict saddle parameters for low-rank matrix problems depend on the singular values of the optimal solution — so we can't realistically assume advance knowledge.

Question: Can we design an algorithm without expensive initialization that does not depend on the strict saddle parameters with a worst case complexity of $\mathcal{O}(\log \max\{\epsilon_g^{-1}, \epsilon_H^{-1}\})$?

Approach

- Maintain γ_k as an upper estimate of strict saddle parameter γ (separation of $\lambda_{\min}\nabla^2 G(W)$ from 0 around strict saddles).
- Define estimates of the regularity parameters α_k and β_k (used in Local Phase) in terms of γ_k.
- Use these estimates to check the large gradient condition **3** and the large negative curvature condition **2**.
- If either of these are satisfied, the appropriate type of step is taken.
- Otherwise, we enter the Local Phase, within which gradient descent on G converges at a linear rate.
- (If we entered the local phase prematurely — because our estimate γ_k was too large — we return to the main algorithm expeditiously.)

Line-Search Algorithm

initial point W^0; initial guess $\gamma_0 \geq \sigma_r(X^*)$;
Iteration k:
if $\|\nabla G(W^k)\|_F \geq \gamma_k^{3/2}$ **then** {Large Gradient; Steepest Descent}
  Find W^{k+1} by backtracking on $-\nabla G(W^k)$; $\gamma_{k+1} \leftarrow \gamma_k$;
else
  Call MEO to approximate $\lambda_{\min}(\nabla^2 G(W^k))$;
  if MEO certifies $\lambda_{\min}(\nabla^2 G(W^k)) \geq -\gamma_k$ **then** {Local Phase}
    if Local Phase successful **then**
      TERMINATE;
    else
      $\gamma_{k+1} \leftarrow \gamma_k/2$;
    end if
  else {MEO Found Negative Curvature Direction D_k}
    Find W^{k+1} by backtracking on D^k;
    $\gamma_{k+1} \leftarrow \gamma_k$;
  end if
end if

MEO: A Complication?

The usual MEO implementation based on randomized Lanczos introduces a term $O(\epsilon^{-1/2})$ into the complexity (for finding a direction with curvature less than $-\epsilon$ in $\nabla^2 G(W)$).

This seems to destroy our hope of $\log \epsilon$ operation complexity!

But we note two things:

1. We can bound the curvature below in terms of easily evaluated terms:

$$\lambda_{\min}(\nabla^2 G(W)) \geq -2\|\nabla f(X)\|_F - \tfrac{1}{2}\|U^T U - V^T V\|_2,$$

   and use this to monitor convergence.

2. When we call MEO, we seek negative curvatures of $-\frac{1}{2}\gamma_k$, which is independent of ϵ_H! Thus the cost of MEO is bounded by $\log \gamma^{-1}$.

Conclusion: MEO does not mess up the operation complexity after all.

Complexity Results

- When a large gradient or large negative curvature step is taken, get

$$G(W^{k+1}) \leq G(W^k) - \mathcal{O}(\gamma_k^3).$$

- γ_k never gets reduced below $\frac{1}{2}\gamma$.

Theorem [O'Neill and Wright, 2020]

W.h.p. the algorithm terminates in at most

$$\mathcal{O}\left(\sigma_r(X^*)^{-3} + \frac{\log \max\{\epsilon_g^{-1}, \epsilon_H^{-1}\}}{\log(1/(1-\sigma_r(X^*)/L_g))}\right)$$

iterations at an approximate second-order point, where $\sigma_r(X^*)$ is the r-th singular value at the optimal solution X^* and L_g is the Lipschitz constant of $\nabla F(W)$. Operation complexity is

$$\mathcal{O}\left(\min\left\{(n+m)r\sigma_r(X^*)^{-3}, \sigma_r(X^*)^{-7/2}\right\} + \frac{\log \max\{\epsilon_g^{-1}, \epsilon_H^{-1}\}}{\log(1/(1-\sigma_r(X^*)/L_g))}\right).$$

2. Nonnegativity Bounds: Optimality Conditions

The most elementary inequality constrained problem.

$$\min_{x \geq 0} f(x)$$

First-order conditions: $0 \leq x \perp \nabla f(x) \geq 0$.

Less tersely: Can partition $\{1, 2, \ldots, n\} = \mathcal{A} \cup \mathcal{I} \cup \mathcal{D}$ such that

- $x_i = 0$, $\nabla_i f(x) > 0$ for $i \in \mathcal{A}$ (active);
- $x_i > 0$, $\nabla_i f(x) = 0$ for $i \in \mathcal{I}$ (inactive);
- $x_i = 0$, $\nabla_i f(x) = 0$ for $i \in \mathcal{D}$ (degenerate).

The strongest 2oN conditions are that $v^T \nabla^2 f(x^*) v \geq 0$ for v such that

$$\mathcal{S}_2 = \{v_i = 0,\ i \in \mathcal{A};\ \ v_i \geq 0,\ i \in \mathcal{D}\}.$$

But it can be NP-hard to check this condition. e.g. $f(x) := x^T Q x$ for symmetric Q. Satisfies first-order conditions with $\mathcal{D} = \{1, 2, \ldots, n\}$. But in this case 2oN conditions = copositivity of Q.

The standard "cop-out" is to aim for a weaker form of 2oN conditions:

$$[\nabla^2 f(x^*)]_{\mathcal{I}\mathcal{I}} \succeq 0.$$

2a. Log-Barrier Method: Approximate Optimality

Define $\bar{x} = \min(x, \mathbf{1})$ and $\bar{X} = \mathrm{diag}(\bar{x})$.

We work with the following approximate 2oN conditions (similar to [Haeser et al., 2018], except that they use X instead of $\bar{X}$).

Approx first-order: $x > 0, \quad \nabla f(x) > -\epsilon \mathbf{1}, \quad \|\bar{X} \nabla f(x)\|_\infty \leq \epsilon,$

Approx second-order: $\bar{X} \nabla^2 f(x) \bar{X} \succeq -\sqrt{\epsilon} I.$

Log-Barrier Approximation

We reduce the bound-constrained problem to unconstrained minimization of the log-barrier function:

$$\phi_\mu(x) := f(x) - \mu \sum_{i=1}^{n} \log(x_i)$$

for some $\mu > 0$. Only defined on the interior of the set $x \geq 0$.

We minimize this for a single (small) value of μ, chosen so that near-optimal second-order points for ϕ_μ satisfy the approximate second-order conditions for the bound-constrained problem.

Approach [O'Neill and Wright, 2019]: Use the Newton-CG approach for unconstrained minimization — modified to ensure positivity of all iterates x^k and well conditioned linear systems — to minimize this function.

Modifying Newton-CG for the Log-Barrier Function

Gradient and Hessian of the log-barrier function are:

$$\nabla \phi_\mu(x) = \nabla f(x) - \mu X^{-1} e \quad \text{and} \quad \nabla^2 \phi_\mu(x) = \nabla^2 f(x) + \mu X^{-2}.$$

Modify Newton-CG as follows:

- Fix $\mu = \frac{1}{4}\epsilon$.
- Precondition / scale the Newton equations with the diagonal $\bar{X}$.
- Keep iterates interior to the nonnegative orthant with a "fraction-to-the-boundary" rule:

$$x^k + d^k \geq (1-\beta)x^k, \quad \text{for fixed } \beta \in [\sqrt{\epsilon}, 1).$$

- Decrease in terms of optimality conditions: Add extra termination test to Modified CG ($\|r^j\|_\infty \leq \bar{\zeta}\mu$).

Log-Barrier Newton-CG

if Not first-order optimal **then**
  Call Modified CG with $H = \bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k$ and $g = \bar{X}_k \nabla \phi_\mu(x^k)$;
  if step type = "negative curvature" **then**
    Scale d to stay interior and flip sign to get d^k;
  else {step type is "damped Newton"}
    Scale d to stay interior to get d^k
  end if
else
  Call MEO with $H = \bar{X}_k \nabla^2 f(x^k) \bar{X}_k$ to output v;
  if MEO certifies that $\lambda_{\min}(\bar{X}_k \nabla^2 f(x^k) \bar{X}_k) \geq -\sqrt{\epsilon}$ **then**
    Terminate;
  else {direction of sufficient negative curvature found}
    Scale v to stay interior and flip sign to get d^k;
  end if
end if
Line Search: Require $\phi_\mu(x^k + \alpha_k \bar{X}_k d^k) < \phi_\mu(x^k) - \frac{\eta}{6} \alpha_k^3 \|d^k\|^3$;
$x^{k+1} \leftarrow x^k + \alpha_k \bar{X}_k d^k$;

Theorem [O'Neill and Wright, 2019]

Assume f smooth and bounded below, and we use Log-Barrier Newton-CG to seek an approximate second-order point.

- Iteration complexity is $\bar{K}_2 = \tilde{\mathcal{O}}\left(n\epsilon^{-1/2} + \epsilon^{-3/2}\right)$ with probability at least $(1-\delta)^{\bar{K}_2}$.
- Operation complexity is $\tilde{\mathcal{O}}(n\epsilon^{-3/4} + \epsilon^{-7/4})$ for large n and $\tilde{\mathcal{O}}(n\epsilon^{-3/2})$ for smaller n.

- The "n" term seems to be an unavoidable consequence of using the log-barrier function. Best previous result is $\tilde{\mathcal{O}}(n\epsilon^{-3/2})$ — we do better in the "large n" case.
- If we assume a priori that $\{x^k\}$ is bounded, get $\tilde{\mathcal{O}}(\epsilon^{-7/4})$ operation complexity.
- Complexities to get an approximate first-order point are the same, but without the possibility of failure in MEO.

2b. Projected Newton-CG

Bounds on a subset $\mathcal{I} \subset \{1, 2, \ldots, n\}$ of the components of x:

$$\min f(x) \text{ subject to } x_i \geq 0 \text{ for all } i \in \mathcal{I}.$$

For feasible x and small positive threshold ϵ, define *active set* $I^{\text{active}}(x)$ and *free set* $I^{\text{free}}(x)$:

$$I^{\text{active}}(x) := \{i \in \mathcal{I} \,|\, 0 \leq x_i \leq \sqrt{\epsilon}\}$$
$$I^{\text{free}}(x) := \{i \in \mathcal{I} \,|\, x_i > \sqrt{\epsilon}\} \cup \mathcal{I}^c$$

Define diagonal scaling matrix $S(x)$ to have ith diagonal 1 when $i \in I^{\text{free}}(x)$ and x_i when $i \in I^{\text{active}}(x)$.

Approx first-order conditions at feasible x:

$$\begin{aligned} \|S(x)\nabla f(x)\| &\leq 2\epsilon, \\ \nabla_i f(x) &\geq -\epsilon, \qquad i \in I^{\text{free}}(x), \\ \nabla_i f(x) &\geq -\epsilon^{3/4}, \quad i \in I^{\text{active}}(x). \end{aligned}$$

Approx second-order condition: $S(x)\nabla^2 f(x) S(x) \succeq -\sqrt{\epsilon} I$.

Projected Newton-CG: Details

Use $g^k = \nabla f(x^k)$, $H^k = \nabla^2 f(x^k)$, $S_k = S(x^k)$;
g^k_{active}, H^k_{active}, $S_{k,\text{active}}$ are the parts corresponding to $I^{\text{active}}(x^k)$.

1. If $\|g^k_{\text{free}}\| > \epsilon$, apply Modified CG to $(H^k_{\text{free}} + 2\sqrt{\epsilon} I)d = -g^k_{\text{free}}$ to obtain either an approximate reduced Newton direction or negative curvature direction d_{free}. Fill out with zeros to get $d^k \in \mathbb{R}^n$.
2. Else If $g^k_i < -\epsilon^{3/4}$ for some $i \in I^{\text{active}}(x^k)$ or $\|S_{k,\text{active}} g^k_{\text{active}}\| > \epsilon$, set $d^k = -g^k$ (negative gradient).
3. Else call MEO with $S_k H^k S_k$. If a direction d found with $d^T(S_k H^k S_k)d < -(\sqrt{\epsilon}/2)\|d\|^2$, search along $S^k d$ (scaled negative curvature).

For each direction d^k, obtain steplength α by backtracking along the projected path: $P(x^k + \alpha d^k)$, where $P(\cdot)$ is projection onto feasibility.

An approach like this investigated for convex QP in [Wright, 1990] and general nonconvex in [Lin and Moré, 1999].

Projected Newton-CG: Result

Theorem [Xie and Wright, 2020]

The algorithm will terminate at a point satisfying the approximate first-order conditions in $O(\epsilon^{-3/2})$ iterations. With high probability, the same iteration complexity holds for the approximate second-order conditions.

Operation complexity (gradients and Hessian-vector products) is a factor of $O(\epsilon^{-1/4})$ greater.

No explicit dependence on dimension n. (But there is dependence on Lipschitz constants.)

Similar results as for the unconstrained case!

3. Nonconvex Optimization with Equality Constraints

$$\min f(x) \quad \text{s.t. } c(x) = 0,$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is smooth, and $c : \mathbb{R}^n \to \mathbb{R}^m$ $(m \leq n)$ is a smooth vector function of equality constraints.

$\nabla c(x) \in \mathbb{R}^{n \times m}$ is the matrix of first partial derivatives of c.

Approx first-order (ϵ-1o):

$$\|\nabla f(x) + \nabla c(x)\lambda\| \leq \epsilon, \quad \|c(x)\| \leq \epsilon.$$

Approx second-order (ϵ-2o):

$$d^T \left(\nabla^2 f(x) + \sum_{i=1}^{m} \lambda_i \nabla^2 c_i(x) \right) d \geq -\epsilon \|d\|^2,$$

for any $d \in \mathbb{R}^n$ such that $\nabla c(x)^T d = 0$.

3a. Proximal Augmented Lagrangian (PAL) algorithm

The augmented Lagrangian is

$$\mathcal{L}_\rho(x,\lambda) \triangleq f(x) + \lambda^T c(x) + \frac{\rho}{2}\|c(x)\|^2,$$

where $\rho > 0$ and $\lambda \triangleq (\lambda_1, \ldots, \lambda_m)^T$.

PAL Algorithm:

0. Initialize x_0, λ_0 and fix $\rho > 0$, $\beta > 0$; Set $k := 0$;
1. Update x_k: Find approximate solution x_{k+1} to

$$\text{argmin } \mathcal{L}_\rho(x,\lambda_k) + \frac{\beta}{2}\|x - x_k\|^2;$$

2. Update λ_k: $\lambda_{k+1} := \lambda_k + \rho c(x_{k+1})$;
3. If termination criterion is satisfied, STOP; otherwise, $k := k+1$ and go to Step 1.

[Xie and Wright, 2019]

Complexities and Assumptions

PAL involves two levels of iteration: the outer iteration, and the inner iterations to solve the nonconvex unconstrained subproblem.

Three types of complexity:

- Outer iteration complexity
- Total iteration complexity: total number of iterations of the inner loop. (a.k.a. "evaluation complexity")
- Operation complexity: bound on number of gradient evaluations / Hessian-vector products.

The assumptions vary between results, but include the following:

- f and c are twice Lipschitz continuously differentiable.
- $f(x) + (\rho_0/2)\|c(x)\|_2^2$ has compact level sets, for some $\rho_0 \geq 0$.
- $\nabla c(x)$ has uniformly full rank m for all x in a compact level set.

Note: We *don't* assume that β is large enough to make the subproblem convex i.e. swamp out the negative curvature in $\mathcal{L}_\rho$.

PAL Outer Iteration Complexity

For any $\epsilon > 0$ and $\eta \in [0, 2]$, choose

- Prox parameter $\beta = \epsilon^\eta/2$ (small)
- Penalty parameter $\rho = O(\epsilon^{-\eta})$ and above some threshold.

Outer iterations for ϵ-1o: Assume that an approx first-order point is found for each subproblem with gradient $\leq \frac{1}{2}\epsilon$, and $\|c(x_0)\| = O(\epsilon^{\eta/2})$. Then an ϵ-1o solution is found in $O(\epsilon^{\eta-2})$ outer iterations.

$\eta = 2$: need only $O(1)$ outer iterations! (But then the subproblems are extremely ill conditioned.)

$\eta = 0$: (settings of β and ρ are independent of ϵ), get $O(\epsilon^{-2})$ outer iterations.

Outer iterations for ϵ-2o: Require $\eta \in [1, 2]$, and additionally that subproblem solutions have Hessian $\succeq -\frac{1}{2}\epsilon I$. Then an ϵ-2o point is found in $O(\epsilon^{\eta-2})$ outer iterations.

PAL Total Iteration and Operation Complexity

Use line-search Newton-CG to solve the subproblems inexactly. Then get estimates of total iteration complexity and operation complexity. Require

$$\nabla_x \mathcal{L}_\rho(x_{k+1}, \lambda_k) + \beta(x_{k+1} - x_k) = \tilde{r}_{k+1},$$
$$\nabla^2_{xx} \mathcal{L}_\rho(x_{k+1}, \lambda_k) + \beta I \succeq -\tfrac{1}{2}\epsilon I,$$

with $\|\tilde{r}_k\| \leq \min(1/k, \frac{1}{2}\epsilon)$.

Total Iter Complexity: $\eta \in [1, 2]$, ϵ-2o point w.h.p.:

Constraints	Total Iter.	Optimal	
nonlinear	$\mathcal{O}(\epsilon^{-2\eta-5})$	$\mathcal{O}(\epsilon^{-7})$	$(\eta = 1)$
linear	$\mathcal{O}(\epsilon^{\eta-5})$	$\mathcal{O}(\epsilon^{-3})$	$(\eta = 2)$

Operation Complexity: $\eta \in [1, 2]$, ϵ-2o point w.h.p.:

Constraints	Operations	Optimal	
nonlinear	$\mathcal{O}(\epsilon^{-5\eta/2-11/2})$	$\mathcal{O}(\epsilon^{-8})$	$(\eta = 1)$
linear	$\mathcal{O}(\epsilon^{\eta/2-11/2})$	$\mathcal{O}(\epsilon^{-9/2})$	$(\eta = 2)$

PAL: Choosing ρ

Recall that $\rho = O(\epsilon^{-\eta})$, but it also has to be above a certain threshold (depending on many problem-dependent parameters).

Can wrap an outer loop around PAL in which ρ is increased by a constant factor on each loop.

For each ρ, run PAL for the # of outer iterations predicted by the theory. If an ϵ-2o point is not found by then, increase ρ and try again.

Increases total iteration complexity by a factor of only $\log \epsilon$.

Related Work

There are many related works on nonconvex constrained optimization, with complexity analyses of various types. See [Xie and Wright, 2020].

Often in more limited settings (e.g. linear constraints, approximate first-order optimality, iteration / evaluation complexity only, explicit second derivatives, use original feasible set in subproblems).

- Augmented Lagrangian type: [Grapiglia and Yuan, 2019], [Birgin and Martínez, 2019]
- (Modified) Proximal AL (linear constraints): [Zhang and Luo, 2020], [Hajinezhad and Hong, 2019]
- Other algorithms: Exact penalty [Cartis et al., 2011], two-phase target-following [Cartis et al., 2014, Cartis et al., 2019b] and others.
- Methods that use the original feasible set in the subproblems: SQP [Nouiehed et al., 2018], cubic regularization [Cartis et al., 2015, Cartis et al., 2020a], active set [Birgin and Martínez, 2018], interior-point [Haeser et al., 2018].

Conclusions

- Second-order necessary points are of interest in some new applications
- Complexity analysis of (nearly) practical methods for finding such points is interesting too
- Significant open questions, particularly surrounding nonconvex nonlinear constrained:
  - primal-dual interior-point methods;
  - exact penalty functions;
  - can some "impractial" methods with good complexity inspire "practical" methods?

Thanks to OWOS Organizers!

References I

Agarwal, N., Allen-Zhu, Z., Bullins, B., Hazan, E., and Ma, T. (2017).
Finding approximate local minima faster than gradient descent.
arXiv:1611.01146v4.

Birgin, E. G. and Martínez, J. M. (2017).
The use of quadratic regularization with a cubic descent condition for unconstrained optimization.
SIAM J. Optim., 27:1049–1074.

Birgin, E. G. and Martínez, J. M. (2018).
On regularization and active-set methods with complexity for constrained optimization.
SIAM Journal on Optimization, 28(2):1367–1395.

Birgin, E. G. and Martínez, J. M. (2019).
Complexity and performance of an augmented Lagrangian algorithm.
arXiv preprint arXiv:1907.02401.

Carmon, Y., Duchi, J. C., Hinder, O., and Sidford, A. (2017a).
Accelerated methods for non-convex optimization.
arXiv:1611.00756v2.

References II

Carmon, Y., Duchi, J. C., Hinder, O., and Sidford, A. (2017b).
"Convex until proven guilty": Dimension-free acceleration of gradient descent on non-convex functions.
In *Volume 70: International Conference on Machine Learning, 6-11 August 2017, International Convention Centre, Sydney, Australia*, pages 654–663. PMLR.

Cartis, C., Gould, N. I. M., and Toint, P. L. (2011).
On the evaluation complexity of composite function min- imization with applications to nonconvex nonlinear programming.
SIAM Journal on Optimization, 21(4):1721–1739.

Cartis, C., Gould, N. I. M., and Toint, P. L. (2012).
An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function-evaluation complexity.
IMA Journal of Numerical Analysis, 32(4):1662–1695.

Cartis, C., Gould, N. I. M., and Toint, P. L. (2014).
On the complexity of finding first-order critical points in constrained nonlinear optimization.
Mathematical Programming, Series A, 144:93–106.

References III

Cartis, C., Gould, N. I. M., and Toint, P. L. (2015).
On the evaluation complexity of constrained nonlinear least-squares and general constrained nonlinear optimization using second-order methods.
SIAM Journal on Numerical Analysis, 53(2):836–851.

Cartis, C., Gould, N. I. M., and Toint, P. L. (2019a).
A concise second-order evaluation complexity for unconstrained nonlinear optimization using high-order regularized models.
Optimization Methods and Software.

Cartis, C., Gould, N. I. M., and Toint, P. L. (2019b).
Optimization of orders one to three and beyond: Characterization and evaluation complexity in constrained nonconvex optimization.
Journal of Complexity, 53:68–94.

Cartis, C., Gould, N. I. M., and Toint, P. L. (2020a).
Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints.
SIAM Journal on Optimization, 30(1):513–541.

Cartis, C., Gould, N. I. M., and Toint, P. L. (2020b).
Strong evaluation complexity bounds for arbitrary-order optimization of nonconvex nonsmooth composite functions.
arXiv preprint arXiv:2001.10802.

References IV

Curtis, F. E., Robinson, D. P., Royer, C. W., and Wright, S. J. (2019).
Trust-region Newton-CG with strong second-order complexity guarantees for nonconvex optimization.
arXiv preprint arXiv:1912.04365, (arXiv:1912.04365).
Revised July 2020.

Curtis, F. E., Robinson, D. P., and Samadi, M. (2017a).
An inexact regularized Newton framework with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization.
Technical Report 17T-011, COR@L Laboratory, Department of ISE, Lehigh University.

Curtis, F. E., Robinson, D. P., and Samadi, M. (2017b).
A trust region algorithm with a worst-case iteration complexity of $O\left(\epsilon^{-3/2}\right)$ for nonconvex optimization.
Math. Program., 162:1–32.

Grapiglia, G. N. and Yuan, Y. (2019).
On the complexity of an augmented lagrangian method for nonconvex optimization.
arXiv preprint arXiv:1906.05622.

Griewank, A. (1981).
The modification of Newton's method for unconstrained optimization by bounding cubic terms.
Technical Report NA/12, DAMTP, Cambridge University.

References V

Haeser, G., Liu, H., and Ye, Y. (2018).
Optimality condition and complexity analysis for linearly-constrained optimization without differentiability on the boundary.
Mathematical Programming, 178:263–299.

Hajinezhad, D. and Hong, M. (2019).
Perturbed proximal primal–dual algorithm for nonconvex nonsmooth optimization.
Mathematical Programming, 176:207–245.

Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. (2017a).
How to escape saddle points efficiently.
arXiv:1703.00887v1.

Jin, C., Netrapalli, P., and Jordan, M. I. (2017b).
Accelerated gradient descent escapes saddle points faster than gradient descent.
arXiv:1711.10456.

Kuczyński, J. and Woźniakowski, H. (1992).
Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start.
SIAM J. on Matrix Analysis and Applications, 13(4):1094–1122.

References VI

Kuczyński, J. and Woźniakowski, H. (1994).
Probabilistic bounds on the extremal eigenvalues and condition number by the Lanczos algorithm.
SIAM J. on Matrix Analysis and Applications, 15(2):672–691.

Lin, C. and Moré, J. J. (1999).
Newton's method for large bound-constrained optimization problems.
SIAM Journal on Optimization, 9(4):1100–1127.

Martínez, J. M. and Raydan, M. (2017).
Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization.
J. Global Optim., 68:367–385.

Nemirovski, A. S. and Yudin, D. B. (1983).
Problem Complexity and Method Efficiency in Optimization.
John Wiley.

Nesterov, Y. and Polyak, B. T. (2006).
Cubic regularization of Newton method and its global performance.
Mathematical Programming, Series A, 108:177–205.

Nouiehed, M., Lee, J. D., and Razaviyayn, M. (2018).
Convergence to second-order stationarity for constrained non-convex optimization.
arXiv preprint arXiv:1810.02024.

References VII

O'Neill, M. and Wright, S. J. (2019).
A log-barrier Newton-CG method for bound constrained optimization with complexity guarantees.
(arXiv:1904.03563).
To appear in *IMA Journal on Numerical Analysis*.

O'Neill, M. and Wright, S. J. (2020).
A line-search descent algorithm for strict saddle functions with complexity guarantees.
arXiv preprint arXiv:2006.07925.

Royer, C. W., O'Neill, M., and Wright, S. J. (2020).
A Newton-CG algorithm with complexity guarantees for smooth unconstrained optimization.
Mathematical Programming, Series A, 180(arXiv:1803.02924):451–488.

Wright, S. J. (1990).
Implementing proximal point methods for linear programming.
Journal of Optimization Theory and Applications, 65:531–554.

Xie, Y. and Wright, S. J. (2019).
Complexity of proximal augmented Lagrangian for nonconvex optimization with nonlinear equality constraints.
Technical Report arrXiv:1908.00131, University of Wisconsin-Madison.

References VIII

Xie, Y. and Wright, S. J. (2020).
Complexity of projected Newton methods in bound-constrained optimization.
In preparation.

Zhang, J. and Luo, Z. Q. (2020).
A proximal alternating direction method of multiplier for linearly constrained nonconvex minimization.
SIAM Journal on Optimization, 30(3):2272–2302.

Zhu, Z., Li, Q., Tang, G., and Wakin, M. B. (2017).
The global optimization geometry of low-rank matrix optimization.
arXiv preprint arXiv:1703.01256.