Complexity analysis of second-order algorithms based on line search for smooth nonconvex optimization

Clément Royer - University of Wisconsin-Madison

Joint work with Stephen J. Wright

MOPTA, Bethlehem, Pennsylvania, USA - August 17, 2017

Smooth nonconvex optimization

We consider an unconstrained smooth problem:

$$\min_{x \in \mathbb{R}^n} f(x).$$

Assumptions on f

- f bounded from below.
- f twice continuously differentiable.
- f is not convex.

Optimality conditions

Second-order necessary point

x^* satisfies the second-order necessary conditions if

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

Basic paradigm

If x is not a second-order necessary point, $\exists\ d$ such that

1. $d^\top \nabla f(x) < 0$: **gradient-type direction**.
   and/or
2. $d^\top \nabla^2 f(x) d < 0$: negative curvature direction
   $\Rightarrow$ **specific to nonconvex problems**.

Motivation

Example: Nonconvex formulation of low-rank matrix problems

$$\min_{U \in \mathbb{R}^{n \times r}, V \in \mathbb{R}^{m \times r}} f(U V^\top).$$

For common classes of problems:

- Second-order necessary points are global minimizers (or close).
- Saddle points have negative curvature.

Motivation

Example: Nonconvex formulation of low-rank matrix problems

$$\min_{U \in \mathbb{R}^{n \times r}, V \in \mathbb{R}^{m \times r}} f(U V^\top).$$

For common classes of problems:

- Second-order necessary points are global minimizers (or close).
- Saddle points have negative curvature.

- Renewed interested: Second-order necessary points of **nonconvex problems**.
- Needed: **Efficient algorithms**.

Second-order complexity

Principle

For a given method, two tolerances $\epsilon_g, \epsilon_H \in (0, 1)$:

- **Obj:** bound the **worst-case cost** of reaching x_k such that

$$\|\nabla f(x_k)\| \leq \epsilon_g, \qquad \lambda_k = \lambda_{\min}(\nabla^2 f(x_k)) \geq -\epsilon_H.$$

- Focus: Bound dependencies on ϵ_g, ϵ_H.

Second-order complexity

Principle

For a given method, two tolerances $\epsilon_g, \epsilon_H \in (0, 1)$:

- **Obj:** bound the **worst-case cost** of reaching x_k such that

$$\|\nabla f(x_k)\| \leq \epsilon_g, \qquad \lambda_k = \lambda_{\min}(\nabla^2 f(x_k)) \geq -\epsilon_H.$$

- Focus: Bound dependencies on ϵ_g, ϵ_H.

- Definition of **cost** ?
- Best rates ?

Existing complexity results

Nonconvex optimization literature

- Classical cost: Number of (expensive) iterations.
- Best methods: Newton-type frameworks.

Existing complexity results

Nonconvex optimization literature

- Classical cost: Number of (expensive) iterations.
- Best methods: Newton-type frameworks.

Algorithms	Bounds
Classical trust region	$\mathcal{O}\left(\max\{\epsilon_g^{-2}\epsilon_H^{-1}, \epsilon_H^{-3}\}\right)$
Cubic regularization TRACE trust region	$\mathcal{O}\left(\max\{\epsilon_g^{-\frac{3}{2}}, \epsilon_H^{-3}\}\right)$

Existing complexity results (2)

Learning/Statistics community

- Specific setting $\epsilon_g = \epsilon, \epsilon_H = \mathcal{O}(\sqrt{\epsilon})$.

  Best Newton-type bound: $\mathcal{O}(\epsilon^{-\frac{3}{2}})$.
- Gradient-based $\Rightarrow$ cheaper iterations.
- Cost measure: *Hessian-vector products/gradient evaluations*.

Existing complexity results (2)

Learning/Statistics community

- Specific setting $\epsilon_g = \epsilon, \epsilon_H = \mathcal{O}(\sqrt{\epsilon})$.

  Best Newton-type bound: $\mathcal{O}(\epsilon^{-\frac{3}{2}})$.
- Gradient-based $\Rightarrow$ cheaper iterations.
- Cost measure: *Hessian-vector products/gradient evaluations*.

Algorithms	Bounds
Gradient descent methods with random noise	$\tilde{\mathcal{O}}\left(\epsilon^{-2}\right)$
Accelerated gradient methods for nonconvex problems	$\tilde{\mathcal{O}}(\epsilon^{-\frac{7}{4}})$

- $\tilde{\mathcal{O}}(\cdot)$: logarithmic factors.
- Results hold with **high probability**.

Our objective

Illustrate all the possible complexities...

- In terms of iterations, evaluations, etc.
- For arbitrary ϵ_g, ϵ_H.
- Deterministic and high probability results.

Our objective

Illustrate all the possible complexities...

- In terms of iterations, evaluations, etc.
- For arbitrary ϵ_g, ϵ_H.
- Deterministic and high probability results.

...in a single framework

- Based on line search.
- Matrix-free: only require Hessian-vector products.
- Good complexity guarantees.

Outline

Outline

Basic framework

Parameters: $x_0 \in \mathbb{R}^n,\ \theta \in (0,1),\ \eta > 0, \epsilon_g \in (0,1),\ \epsilon_H \in (0,1).$

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

1. Compute a search direction d_k.
2. Perform a backtracking line search to compute $\alpha_k = \theta^{j_k}$ such that

$$f(x_k + \alpha_k d_k) \ < \ f(x_k) - \frac{\eta}{6}\alpha_k^3 \|d_k\|^3.$$

3. Set $x_{k+1} = x_k + \alpha_k\, d_k$.

Selecting the search direction d_k

Step 1: Use gradient related information

- Compute

$$g_k = \nabla f(x_k), \; R_k = \frac{g_k^\top \nabla^2 f(x_k) g_k}{\|g_k\|^2}.$$

- If $R_k < -\epsilon_H$, set

$$d_k = \frac{R_k}{\|g_k\|} g_k.$$

- Elseif $R_k \in [-\epsilon_H, \epsilon_H]$ **and** $\|g_k\| > \epsilon_g$, set

$$d_k = -\frac{g_k}{\|g_k\|^{1/2}}.$$

- Otherwise perform Step 2.

Selecting the search direction d_k (2)

Step 2: Use eigenvalue information

- Compute an eigenpair (v_k, λ_k) such that $\lambda_k = \lambda_{\min}(\nabla^2 f(x_k))$ and

$$\nabla^2 f(x_k) v_k = \lambda_k v_k, \quad v_k^\top g_k \leq 0, \quad \|v_k\| = 1.$$

- Case $\lambda_k < -\epsilon_H$: $d_k = -\lambda_k v_k$;
- Case $\lambda_k > \epsilon_H$ - Newton step:

$$d_k = d_k^n, \quad \nabla^2 f(x_k) d_k^n = -g_k;$$

- Case $\lambda_k \in [-\epsilon_H, \epsilon_H]$ - regularized Newton step:

$$d_k = d_k^r, \quad \left(\nabla f^2(x_k) + 2\epsilon_H\right) d_k^r = -g_k.$$

Outline

Assumptions and notations

Assumptions

- $\mathcal{L}_f(x_0) = \{x | f(x) \leq f(x_0)\}$ compact.
- f twice continuously differentiable on a open set containing $\mathcal{L}_f(x_0)$, with Lipschitz continuous Hessian.

- L_H: Lipschitz constant for $\nabla^2 f$.
- f_{low}: lower bound on $\{f(x_k)\}$.
- U_H: upper bound on $\|\nabla^2 f(x_k)\|$.

Criterion

Approximate solution

x_k is an (ϵ_g, ϵ_H)-**point** if

$$\min\{\|g_k\|, \|g_{k+1}\|\} \leq \epsilon_g, \quad \lambda_k \geq -\epsilon_H.$$

Criterion

Approximate solution

x_k is an (ϵ_g, ϵ_H)-**point** if

$$\min\{\|g_k\|, \|g_{k+1}\|\} \leq \epsilon_g, \quad \lambda_k \geq -\epsilon_H.$$

Other possibilities:

- Remove gradient directions and use $\|g_{k+1}\|$
  No cheap gradient steps.
- Add a stopping criterion and use $\|g_k\|$.
  No global/local convergence.

Analysis of the method

Key principle

Bound the **decrease** produced at every step while an (ϵ_g, ϵ_H)-point has not been reached.

Analysis of the method

Key principle

Bound the **decrease** produced at every step while an (ϵ_g, ϵ_H)-point has not been reached.

- Five possible directions.
  - Two ways of scaling $-g_k$:
    - By its (negative) curvature;
    - By its norm;
  - Negative eigenvector;
  - Newton step;
  - Regularized Newton step.

Analysis of the method

Key principle

Bound the **decrease** produced at every step while an (ϵ_g, ϵ_H)-point has not been reached.

- Five possible directions.
  - Two ways of scaling $-g_k$:
    - By its (negative) curvature;
    - By its norm;
  - Negative eigenvector;
  - Newton step;
  - Regularized Newton step.
- **One proof technique**, typical of backtracking line search
  - If unit step is accepted, guaranteed decrease;
  - Otherwise, lower bound on accepted step size.

Example: When $d_k = -g_k/\|g_k\|^{1/2}$

- In that case: $\frac{g_k^\top \nabla^2 f(x_k) g_k}{\|g_k\|^2} \in [-\epsilon_H, \epsilon_H], \qquad \|g_k\| > \epsilon_g.$
- Unit step accepted:

$$f(x_k) - f(x_{k+1}) \;\geq\; \frac{\eta}{6}\|d_k\|^3 \;\geq\; \frac{\eta}{6}\epsilon_g^{\frac{3}{2}}.$$

- Unit step rejected: By Taylor expansion, there exists a step $\alpha_k = \theta^{j_k}$ that is accepted such that

$$\theta^{j_k} \;\geq\; \theta \min\left\{\frac{5}{3}, \sqrt{\frac{1}{L_H + \eta}}\right\} \epsilon_g^{\frac{1}{2}} \epsilon_H^{-1}.$$

So the line search terminates and

$$f(x_k) - f(x_{k+1}) \;\geq\; \frac{\eta}{6}\alpha_k^3\|d_k\|^3 \;\geq\; \mathcal{O}\left(\epsilon_g^3 \epsilon_H^{-3}\right).$$

Example: When $d_k = -g_k/\|g_k\|^{1/2}$

- In that case: $\frac{g_k^\top \nabla^2 f(x_k) g_k}{\|g_k\|^2} \in [-\epsilon_H, \epsilon_H], \qquad \|g_k\| > \epsilon_g.$
- Unit step accepted:

$$f(x_k) - f(x_{k+1}) \;\geq\; \frac{\eta}{6}\|d_k\|^3 \;\geq\; \frac{\eta}{6}\epsilon_g^{\frac{3}{2}}.$$

- Unit step rejected: By Taylor expansion, there exists a step $\alpha_k = \theta^{j_k}$ that is accepted such that

$$\theta^{j_k} \;\geq\; \theta \min\left\{\frac{5}{3}, \sqrt{\frac{1}{L_H + \eta}}\right\} \epsilon_g^{\frac{1}{2}} \epsilon_H^{-1}.$$

So the line search terminates and

$$f(x_k) - f(x_{k+1}) \;\geq\; \frac{\eta}{6}\alpha_k^3\|d_k\|^3 \;\geq\; \mathcal{O}\left(\epsilon_g^3 \epsilon_H^{-3}\right).$$

- Final decrease:

$$f(x_k) - f(x_{k+1}) \geq c_g \;\min\left\{\epsilon_g^3 \epsilon_H^{-3}, \epsilon_g^{\frac{3}{2}}\right\}.$$

Decrease bound

General decrease lemma

If at the k-th iteration, an (ϵ_g, ϵ_H)-point has not been reached, then

$$f(x_k) - f(x_{k+1}) \geq c \min\left\{\epsilon_g^{\frac{3}{2}}, \epsilon_H^3, \epsilon_g^3 \epsilon_H^{-3}, \varphi(\epsilon_g, \epsilon_H)^3\right\},$$

where

$$\varphi(\epsilon_g, \epsilon_H) = L_H^{-1} \epsilon_H \left(-2 + \sqrt{4 + 2L_H \epsilon_g / \epsilon_H^2}\right).$$

- c depends on L_H, η, θ.

Iteration complexity

Iteration complexity bound

The method reaches an (ϵ_g, ϵ_H)-point in at most

$$\frac{f_0 - f_{\text{low}}}{c} \max\left\{\epsilon_g^{-\frac{3}{2}}, \epsilon_H^{-3}, \epsilon_g^{-3}\epsilon_H^3, \varphi(\epsilon_g, \epsilon_H)^{-3}\right\}$$

iterations.

Specific rates:

- $\epsilon_g = \epsilon,\ \epsilon_H = \sqrt{\epsilon}$: $\mathcal{O}(\epsilon^{-\frac{3}{2}})$.
- $\epsilon_g = \epsilon_H = \epsilon$: $\mathcal{O}(\epsilon^{-3})$.
- Optimal bounds for Newton-type methods.

Function evaluation complexity

- #Iterations $=$ #Gradient/#Hessian evaluations.
- #Iterations $\leq$ **#Function evaluations**.

Function evaluation complexity

- #Iterations = #Gradient/#Hessian evaluations.
- #Iterations $\leq$ **#Function evaluations**.

Line-search iterations

If x_k is not a (ϵ_g, ϵ_H)-point, the line search takes at most

$$\mathcal{O}\left(\log_\theta\left(\min\{\epsilon_g^{\frac{1}{2}}\epsilon_H^{-1}, \epsilon_H^2\}\right)\right) \text{ iterations.}$$

Evaluation complexity bound

The method reaches an (ϵ_g, ϵ_H)-point in at most

$$\tilde{\mathcal{O}}\left(\max\left\{\epsilon_g^{-\frac{3}{2}}, \epsilon_H^{-3}, \epsilon_g^{-3}\epsilon_H^3, \varphi(\epsilon_g, \epsilon_H)^{-3}\right\}\right)$$

function evaluations.

Outline

Motivation

Algorithmic cost

- The method should be matrix-free.
- We use matrix-related operations:
  - Linear system solve;
  - Eigenvalue/Eigenvector computation.

Inexactness

- Perform the matrix operations inexactly.
- Main cost unit: **matrix-vector product**/gradient evaluation.

Conjugate gradient for linear systems

- We solve systems of the form $Hd = -g$, with $H \succeq \epsilon_H I$.

Conjugate gradient for linear systems

- We solve systems of the form $Hd = -g$, with $H \succeq \epsilon_H I$.

Conjugate Gradient (CG)

- We apply the conjugate gradient algorithm with stopping criterion:

$$\|Hd + g\| \leq \frac{\xi}{2} \min\{\|g\|, \epsilon_H \|d\|\}, \qquad \xi \in (0, 1).$$

- If $\kappa = \lambda_{\max}(H)/\lambda_{\min}(H)$, the CG method will find such a vector in at most

$$\min\left\{n, \tfrac{1}{2}\sqrt{\kappa}\log\left(4\kappa^{\frac{5}{2}}/\xi\right)\right\} = \min\left\{n, \mathcal{O}\left(\sqrt{\kappa}\log(\kappa/\xi)\right)\right\}$$

matrix-vector products.

Lanczos for eigenvalue computation

- Lanczos method to compute a minimum eigenvector.
- Can fail if deterministic $\Rightarrow$ **Random start**.
- Results for matrices $A \succeq 0 \Rightarrow$ Change the Hessian.

Lanczos for eigenvalue computation

- Lanczos method to compute a minimum eigenvector.
- Can fail if deterministic $\Rightarrow$ **Random start**.
- Results for matrices $A \succeq 0 \Rightarrow$ Change the Hessian.

Lanczos iterations

Let $H \in \mathbb{R}^{n \times n}$ symmetric with $\|H\| \leq U_H,\ \epsilon > 0,\ \delta \in (0,1)$.
With probability at least $1 - \delta$, the Lanczos procedure applied to $U_H I - H$ outputs a vector v such that

$$v^\top H v \ \leq\ \lambda_{\min}(H) + \epsilon.$$

in at most

$$\min\left\{n, \frac{\ln(n/\delta^2)}{2\sqrt{2}}\sqrt{\frac{U_H}{\epsilon}}\right\}$$

iterations/matrix-vector products.

Selecting the search direction d_k - Inexact version

Step 1: Use gradient related information

- Compute

$$g_k = \nabla f(x_k), \; R_k = \frac{g_k^\top \nabla^2 f(x_k) g_k}{\|g_k\|^2}.$$

- If $R_k < -\epsilon_H$, set

$$d_k = \frac{R_k}{\|g_k\|} g_k.$$

- Elseif $R_k \in [-\epsilon_H, \epsilon_H]$ **and** $\|g_k\| \geq \epsilon_g$, set

$$d_k = -\frac{g_k}{\|g_k\|^{\frac{1}{2}}}.$$

- Otherwise perform the Inexact Step 2.

Selecting the direction d_k - Inexact version (2)

Inexact Step 2: Use (inexact) eigenvalue information

- Compute an eigenpair (v_k^i, λ_k^i) such that **with probability** $1-\delta$,

$$\lambda_k^i = [v_k^i]^\top \nabla^2 f(x_k) v_k^i \leq \lambda_k + \frac{\epsilon_H}{2}, \quad [v_k^i]^\top g_k \leq 0, \quad \|v_k^i\| = 1.$$

- Case $\lambda_k^i < -\frac{1}{2}\epsilon_H$: $d_k = v_k^i$;
- Case $\lambda_k^i > \frac{3}{2}\epsilon_H$:
  - Inexact Newton: Use CG to obtain

$$d_k = d_k^{in}, \quad \left\|\nabla^2 f(x_k) d_k^{in} + g_k\right\| \leq \frac{\xi}{2} \min\left\{\|g_k\|, \epsilon_H \|d_k^{in}\|\right\};$$

- Case $\lambda_k^i \in [-\frac{1}{2}\epsilon_H, \frac{3}{2}\epsilon_H]$:
  - Inexact regularized Newton: Use CG to obtain

$$d_k = d_k^{ir}, \quad \left\|\left[\nabla^2 f(x_k) + 2\epsilon_H\right] d_k^{ir} + g_k\right\| \leq \frac{\xi}{2} \min\left\{\|g_k\|, \epsilon_H \|d_k^{ir}\|\right\}.$$

Complexity analysis of the inexact method

- Identical reasoning: 5 steps, 1 proof.
- Using Lanczos with a random start, the negative curvature decrease only holds with probability $1 - \delta$.
- With CG, the inexact Newton and regularized Newton give slightly different formulas.

Decrease lemma

For any iteration k, if x_k is not an (ϵ_g, ϵ_H)-point,

$$f(x_k) - f(x_{k+1}) \geq \hat{c} \min \left\{ \epsilon_g^3 \epsilon_H^{-3}, \epsilon_g^{\frac{3}{2}}, \epsilon_H^3, \varphi\left(\epsilon_g, \tfrac{\xi}{2}\epsilon_H\right)^3, \varphi\left(\epsilon_g, \tfrac{4+\xi}{2}\epsilon_H\right)^3 \right\},$$

with probability at least $1 - \delta$, and $\hat{c}$ only depends on L_H, η, θ.

Complexity results

Iteration complexity

An (ϵ_g, ϵ_H)-point is reached in at most

$$\hat{K} := \frac{f_0 - f_{\text{low}}}{\hat{c}} \max\left\{\epsilon_g^{-3}\epsilon_H^3, \epsilon_g^{-\frac{3}{2}}, \epsilon_H^{-3}, \varphi\left(\epsilon_g, \tfrac{\xi}{2}\epsilon_H\right)^{-3}, \varphi\left(\epsilon_g, \tfrac{4+\xi}{2}\epsilon_H\right)^{-3}\right\},$$

iterations, **with probability at least** $1 - \hat{K}\delta$.

Cost complexity

The number of **Hessian-vector products or gradient evaluations** needed to reach an (ϵ_g, ϵ_H)-point is at most

$$\min\left\{n, \mathcal{O}\left(U_H^{1/2}\epsilon_H^{-\frac{1}{2}}\log(\epsilon_H^{-1}/\xi)\right), \mathcal{O}\left(U_H^{1/2}\epsilon_H^{-\frac{1}{2}}\log(n/\delta^2)\right)\right\}\hat{K},$$

with probability at least $1 - \hat{K}\delta$.

Complexity results (simplified)

- Setting: $\epsilon_g = \epsilon, \epsilon_H = \sqrt{\epsilon}$.

An $(\epsilon, \sqrt{\epsilon})$-point is reached in at most

- $\mathcal{O}(\epsilon^{-\frac{3}{2}})$ iterations,
- $\tilde{\mathcal{O}}\left(\epsilon^{-\frac{7}{4}}\right)$ Hessian-vector products/gradient evaluations,

with probability $1 - \mathcal{O}(\epsilon^{-\frac{3}{2}}\delta)$.

Complexity results (simplified)

- Setting: $\epsilon_g = \epsilon, \epsilon_H = \sqrt{\epsilon}$.

An $(\epsilon, \sqrt{\epsilon})$-point is reached in at most

- $\mathcal{O}(\epsilon^{-\frac{3}{2}})$ iterations,
- $\tilde{\mathcal{O}}\left(\epsilon^{-\frac{7}{4}}\right)$ Hessian-vector products/gradient evaluations,

with probability $1 - \mathcal{O}(\epsilon^{-\frac{3}{2}}\delta)$.

Setting $\delta = 0$ gives results *in probability 1*:

- Iterations: $\mathcal{O}(\epsilon^{-\frac{3}{2}})$.
- Hessian-vector/gradients: $\mathcal{O}\left(n\epsilon^{-\frac{3}{2}}\right)$.

Summary

Our proposal

- A class of **second-order line-search** methods.
- **Best known complexity guarantees**.
- Features gradient steps and inexactness.
- Can be implemented matrix-free.

For more details...

- **Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization**, C. W. Royer and S. J. Wright, arXiv:1706.03131.
- Also contains local convergence results.

Follow-up

Perspectives

- **Numerical testing** of our class of methods.
- Extension to constrained problems.

Follow-up

Perspectives

- **Numerical testing** of our class of methods.
- Extension to constrained problems.

Thank you for your attention!

croyer2@wisc.edu