# Second-Order Optimization with Lazy Hessians

**Nikita Doikov**

Joint work with El Mahdi Chayti and Martin Jaggi

EPFL, Switzerland

SIAM Conference on Optimization, Seattle

June 3, 2023

## Plan

I. Introduction: second-order methods

II. Lazy Hessians

III. Conclusions and experiments

$$\min_{x \in \mathbb{R}^d} f(x),$$

where $f$ is twice differentiable, possibly non-convex.

**Gradient Method.** Iterate, $k \geq 0$:

$$x_{k+1} \;=\; x_k - \alpha_k \nabla f(x_k)$$

+ **Cheap iterations:** $\mathcal{O}(d)$
+ **Convergence from arbitrary** $x_0$
− **Slow rate**

Let the gradient be Lipschitz continuous:

$$\|\nabla f(x) - \nabla f(y)\| \;\leq\; L_1 \|x - y\|, \qquad \forall x, y \in \mathbb{R}^d.$$

Then, to find $\|\nabla f(\bar{x}_k)\| \leq \varepsilon$, the method needs

$$K \;=\; \mathcal{O}\Big(\tfrac{L_1(f(x_0) - f^\star)}{\varepsilon^2}\Big)$$

iterations.

# Newton's Method with Cubic Regularization

**New assumption.** Let the Hessian be Lipschitz continuous:

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \;\; \leq \;\; L_2 \|x - y\|, \qquad \forall x, y \in \mathbb{R}^d.$$

$\Rightarrow$ global upper model of the objective, for $H \geq L_2$:

$$f(y) \;\; \leq \;\; \Omega(x; y) + \tfrac{H}{6}\|y - x\|^3, \qquad \forall x, y \in \mathbb{R}^d,$$

where

$$\Omega(x; y) \;\; \overset{\text{def}}{=} \;\; f(x) + \langle \nabla f(x), y - x \rangle + \tfrac{1}{2}\langle \nabla^2 f(x)(y - x), y - x \rangle.$$

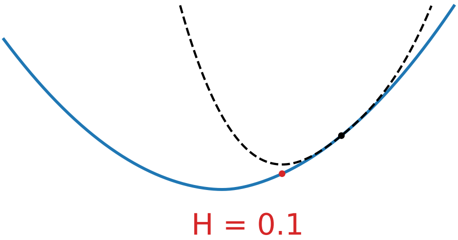
**Cubic Newton** [Nesterov-Polyak, 2006].
Iterate, $k \geq 0$:

$$x_{k+1} \;\; = \;\; \operatorname*{argmin}_{y \in \mathbb{R}^d} \Big\{ \, M_H(x; y) \;\; \equiv \;\; \Omega(x_k; y) + \tfrac{H}{6}\|y - x_k\|^3 \, \Big\}$$

# Cubic Model

Regularized quadratic model of $f(y)$ at point $x \in \mathbb{R}^d$:

$$M_H(x;y) \;\; \equiv \;\; \Omega(x;y) + \frac{H}{6}\|y-x\|^3$$




$\Rightarrow$ global progress of the method.

$$x_{k+1} = \operatorname*{argmin}_{y \in \mathbb{R}^d} \Big\{ M_H(x_k; y) \equiv \Omega(x_k; y) + \frac{H}{6}\|y - x_k\|^3 \Big\}$$

**Theorem.** Let $H := L_2$. Then, to find $\|\nabla f(\bar{x}_k)\| \leq \varepsilon$, the Cubic Newton needs

$$K = \mathcal{O}\Big( \frac{\sqrt{L_2}(f(x_0) - f^\star)}{\varepsilon^{3/2}} \Big)$$

iterations.

- For the Gradient Method, we had $\mathcal{O}(\frac{1}{\varepsilon^2})$
- We also can prove convergence to a second-order stationary point for the Cubic Newton: $\nabla^2 f(\bar{x}_k) \succeq -\sqrt{L_2 \varepsilon} I$.
- Adaptive strategy for $H$: ensure $f(x_{k+1}) \leq M_H(x_k; x_{k+1})$

[Nesterov-Polyak, 2006; Cartis-Gould-Toint, 2011; Grapiglia-Nesterov, 2017]

How to compute one step?

$$h^+ \;=\; \operatorname*{argmin}_{h \in \mathbb{R}^d} \Big\{ \langle g, h \rangle + \tfrac{1}{2}\langle Ah, h \rangle + \tfrac{H}{6}\|h\|^3 \Big\}$$

**Step 1:** compute factorization of $A = A^\top \in \mathbb{R}^{d \times d}$:

$$A \;=\; U \Lambda U^\top,$$

where $U \in \mathbb{R}^{d \times d}$ is orthonormal basis: $UU^\top = I$, and $\Lambda$ is **diagonal** or **tridiagonal** — $\mathcal{O}(d^3)$ arithmetic operations
**Step 2:** solve

$$P_\star \;=\; \min_{h \in \mathbb{R}^d} \Big\{ \langle \bar{g}, h \rangle + \tfrac{1}{2}\langle \Lambda h, h \rangle + \tfrac{H}{6}\|h\|^3 \Big\}$$

using duality:

$$P_\star \;=\; D^\star \;=\; \max_{\substack{\tau \in \mathbb{R} \text{ s.t.} \\ \tau > [-\lambda_{\min}]_+}} \Big\{ -\tfrac{1}{2}\langle (\Lambda + \tau I)^{-1}\bar{g}, \bar{g} \rangle - \tfrac{2^4}{3H^2}\tau^3 \Big\}$$

concave maximization of univariate function — $\tilde{\mathcal{O}}(d)$ operations

- **Cubic Newton step:**

$$x^+ \;=\; \operatorname*{argmin}_{y\in\mathbb{R}^d}\Big\{\; M_H(x;y) \;\Big\} \;=\; x \;-\; \Big(\nabla^2 f(x) + \tau I\Big)^{-1}\nabla f(x),$$

where $\tau$ is the solution of the dual. We have $\tau = \frac{H}{2}\|x^+ - x\|$.

- Let $f$ be convex. Then,

$$r \;\stackrel{\mathrm{def}}{=}\; \|x^+ - x\| \;=\; \|\big(\nabla^2 f(x) + \tfrac{Hr}{2} I\big)^{-1}\nabla f(x)\| \;\le\; \tfrac{2}{Hr}\|\nabla f(x)\|$$

Hence, we have an upper bound: $\boxed{\tau = \frac{Hr}{2} \le \sqrt{\frac{H\|\nabla f(x)\|}{2}}}$.

**Gradient Regularization.** [Ueda-Yamashita, 2014; Mishchenko, 2021; D-Nesterov, 2021]:

$$x^+ \;=\; x - \Big(\nabla^2 f(x) + \sqrt{\tfrac{H\|\nabla f(x)\|}{2}}\, I\Big)^{-1}\nabla f(x)$$

- One matrix inversion; fast global rates

# Newton's Method: conclusions

Classic Newton's step:

$$x_{k+1} \;=\; x_k - \left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$$

Two major issues:

- **No global convergence** $\Rightarrow$ **Cubic Regularization:**

$$x_{k+1} \;=\; x_k - \left[\nabla^2 f(x_k) + \tau_k I\right]^{-1} \nabla f(x_k)$$

where $\tau_k$ is computed at each step by univariate maximization.

For convex functions we can use **Gradient Regularization:**
$\tau_k = \sqrt{\frac{H\|\nabla f(x_k)\|}{2}}$.

- **High arithmetic cost:** $\mathcal{O}(d^3)$

$\Rightarrow$ **this work: Lazy Hessian updates**

It improves the total arithmetic cost of CN by a factor $\sqrt{d}$

## Plan

I. Introduction: second-order methods

II. Lazy Hessians

III. Conclusions and experiments

- Idea: use the same Hessian for $m \geq 1$ iterations.

**Lazy Hessian Updates:** compute new Hessian once per $m$ iterations.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Hessians: | $\nabla^2 f(\mathbf{x}_0)$ | reuse Hessian $\longrightarrow$ | | | $\nabla^2 f(\mathbf{x}_m)$ | reuse Hessian $\longrightarrow$ | |
| Gradients: | $\nabla f(\mathbf{x}_0)$ | $\nabla f(\mathbf{x}_1)$ | $\ldots$ | $\nabla f(\mathbf{x}_{m-1})$ | $\nabla f(\mathbf{x}_m)$ | $\nabla f(\mathbf{x}_{m+1})$ | $\ldots$ |

Appeared first in [Shamanskii, 1967]

# Cubic Newton with Lazy Hessians

Define step of the method with Hessian at some previous point $z$:

$$T_H(x, z) \;=\; \operatorname*{argmin}_{y \in \mathbb{R}^d} \Big\{ \langle \nabla f(x), y - x \rangle + \tfrac{1}{2} \langle \nabla^2 f(z)(y - x), y - x \rangle + \tfrac{H}{6} \|y - x\|^3 \Big\}$$

Define $\pi(k) \overset{\text{def}}{=} k - k \bmod m$.

Cubic Newton with Lazy Hessians

**Iterate,** $k \geq 0$**:**

1. Set last snapshot point $z_k = x_{\pi(k)}$
2. Compute lazy cubic step $x_{k+1} = T_H(x_k, z_k)$

**Theorem.** Let $H := 6mL_2$. Then, to find $\|\nabla f(\bar{x})\| \leq \varepsilon$, the method needs

$$K \;=\; \mathcal{O}\Big(\frac{\sqrt{mL_2}(f(x_0)-f^\star)}{\varepsilon^{3/2}}\Big)$$

lazy steps.

- Worse than the full Cubic Newton by the factor $\sqrt{m}$.

Note: the total number of Hessian updates during these steps is

$$\frac{K}{m} \;=\; \mathcal{O}\Big(\frac{\sqrt{L_2}(f(x_0)-f^\star)}{\sqrt{m}\varepsilon^{3/2}}.\Big)$$

# Arithmetic Cost

» **Choice of $m$?** Optimize the total cost:

$$\texttt{Arithmetic complexity} = K \times \texttt{GradCost} + \frac{K}{m} \times \texttt{HessCost}$$

In many problems: $\boxed{\texttt{HessCost} = d \times \texttt{GradCost}}$

# Example

- Let $f(x) = \frac{1}{n}\sum_{i=1}^{n} \varphi(\langle a_i, x\rangle)$ (includes logistic regression)

Then,

$$\nabla f(x) = A^\top s(x), \quad \text{where} \quad \big[s(x)\big]_i = \tfrac{1}{n}\varphi'(\langle a_i, x\rangle),$$

$$\nabla^2 f(x) = A^\top D(x) A, \quad \text{where} \quad \big[D(x)\big]_{ii} = \tfrac{1}{n}\varphi''(\langle a_i, x\rangle).$$

Hence

$$\texttt{GradCost} = \mathrm{nz}(A) + d^2, \qquad \texttt{HessCost} = d \cdot \mathrm{nz}(A) + d^3.$$

- Neural Networks: computing $\nabla^2 f(x)h$ is the same cost as $\nabla f(x)$, for any $x, h$.

$$\nabla^2 f(x) = \Big[\nabla^2 f(x) e_1 \,\Big|\, \ldots \,\Big|\, \nabla^2 f(x) e_d\Big],$$

» **Choice of $m$?** Optimize the total cost:

$$\texttt{Arithmetic complexity} \;=\; K \times \texttt{GradCost} + \frac{K}{m} \times \texttt{HessCost}$$

In many problems: $\boxed{\texttt{HessCost} \;=\; d \times \texttt{GradCost}}$

Substituting, we get

$$\texttt{Arithmetic complexity}$$

$$= \mathcal{O}\left( \left(\sqrt{m} + \frac{d}{\sqrt{m}}\right) \cdot \frac{\sqrt{L_2}(f(x_0) - f^\star)}{\varepsilon^{3/2}} \right) \times \texttt{GradCost} \;\to\; \min_m$$

Optimal $\boxed{m := d}$ (update the Hessian once per $d$ steps).

## Gradient Regularization and Lazy Hessians

Let $f$ be **convex**. Then we can perform simpler iterations.

Regularized Newton with Lazy Hessians

**Iterate,** $k \geq 0$:

1. Set last snapshot point $z_k = x_{\pi(k)}$
2. Set regularization parameter $\tau_k = \sqrt{H \|\nabla f(x_k)\|}$
3. Compute lazy Newton step:
   $x_{k+1} = x_k - \left(\nabla^2 f(z_k) + \tau_k I\right)^{-1} \nabla f(x_k)$

**Theorem.** Let $H = 3mL_2$. The same global complexity as for the Cubic Newton, with an additive logarithmic term:

$$K = \mathcal{O}\left( \frac{\sqrt{mL_2}(f(x_0) - f^\star)}{\varepsilon^{3/2}} + \ln \frac{\|\nabla f(x_0)\|}{\varepsilon} \right).$$

# Adaptive Scheme

**Algorithm 1** Adaptive Cubic Newton with Lazy Hessians

**Initialization:** $x_0 \in \mathbb{R}^d$, $m \geq 1$. Fix some $H_0 > 0$.

**for** $t = 0, 1, \dots$ **do**
  Compute snapshot Hessian $\nabla^2 f(x_{tm})$
  **do**
    Update $H_t = 2 \cdot H_t$
    **for** $i = 1, \dots, m$ **do**
      Compute lazy cubic step $x_{tm+i} = T_{H_t}(x_{tm+i-1}, x_{tm})$
  **until** $f(x_{tm}) - f(x_{tm+m}) \;\geq\; \frac{1}{\sqrt{H_t}} \sum_{i=1}^{m} \|\nabla f(x_{tm+i})\|_*^{3/2}$
  Set $H_{t+1} = \frac{1}{4} \cdot H_t$

- No need to know any parameters
- Makes the methods universal (working properly on problem classes with Hölder continuous derivatives and uniformly convex objectives)

[Grapiglia-Nesterov, 2017; D-Nesterov, 2019; D-Mishchenko-Nesterov, 2022]

# Local Superlinear Convergence

- Let $f$ be strongly convex: $\nabla^2 f(x) \succeq \mu I$
- Let initial gradient be small enough: $\|\nabla f(x_0)\| \leq \frac{\mu^2}{2^4(3L_2+4H)}$

**Theorem.** Local superlinear convergence for the lazy Hessian updates:

$$\|\nabla f(x_k)\| \quad \leq \quad \frac{\mu^2}{2^2(3L_2+4H)} \cdot \left(\tfrac{1}{2}\right)^{2(1+m/2)^{\pi(k)}(1+(k \bmod m)/2)},$$

where $\pi(k) \stackrel{\mathrm{def}}{=} k - k \bmod m$.

- $m = 1 \;\Rightarrow\;$ local quadratic rate of the classic Newton
- $m \geq 1$ [Shamanskii, 1967]

**Plan**

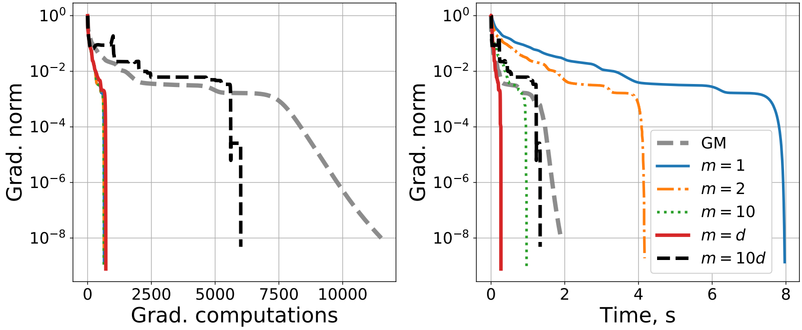
I. Introduction: second-order methods

II. Lazy Hessians

III. Conclusions and experiments

$$\min_{x \in \mathbb{R}^d} f(x) \quad := \quad \mu \ln\Big( \sum_{i=1}^{n} \exp\big(\tfrac{\langle a_i, x\rangle - b_i}{\mu}\big) \Big) \quad \approx \quad \max_{1 \le i \le n} \big[\langle a_i, x\rangle - b_i\big].$$

Log-sum-exp, $d = 100$, $n = 100$, $\mu = 0.5$



**Total arithmetic complexity**

- Gradient Method:

$$\mathcal{O}\left(\frac{L_1(f(x_0)-f^\star)}{\varepsilon^2}\right) \times \texttt{GradCost}$$

- Full Cubic Newton:

$$\mathcal{O}\left(\frac{\sqrt{L_2}(f(x_0)-f^\star)}{\varepsilon^{3/2}}\right) \times \texttt{GradCost} \times d$$

- Lazy Cubic Newton ($m = d$):

$$\mathcal{O}\left(\frac{\sqrt{L_2}(f(x_0)-f^\star)}{\varepsilon^{3/2}}\right) \times \texttt{GradCost} \times \sqrt{d}$$

# Conclusions

- Using cubic regularization or gradient regularization for Newton's method we can establish global convergence
- With lazy Hessian updates we improve the total arithmetic complexity

**Research directions:**

- Convex optimization
- Stochastic methods (we have a follow-up work)
- Sparse problems (different schedules of updating the Hessian)

Thank you very much for your attention!