Adaptive Newton Sketch

Jonathan Lacotte, Yifei Wang and Mert Pilanci

Stanford University

Adaptive Newton Sketch

- A randomized algorithm with **quadratic convergence rate** for convex optimization problems:

$$\min_{x\in\mathbb{R}^d}\{f(x) := f_0(x) + g(x)\}.$$

  - f_0: self-concordant and convex
  - g: self-concordant and μ-strongly convex
- Perform a randomized Newton's step using a random projection of the Hessian:

$$\begin{aligned} H_S(x) =&(\nabla^2 f_0(x)^{\frac{1}{2}})^T S^T S \nabla^2 f_0(x)^{\frac{1}{2}} + \nabla^2 g(x), \\ x_+ =& x + sH_S(x)^{-1}\nabla f(x). \end{aligned}$$

  - $\nabla^2 f_0(x)^{\frac{1}{2}} \in \mathbb{R}^{n\times d}$: Hessian matrix square root
  - $S \in \mathbb{R}^{m\times n}$: sketching matrix with sketching dimension m

Example of loss function and matrix square root

- $f_0(x) = \sum_{i=1}^{n} \ell_i(a_i^\top x)$.
- In this case, a suitable Hessian matrix square root is given by the $n \times d$ matrix

$$\nabla^2 f_0(x)^{1/2} = \mathbf{diag}(\ell_i''(a_i^\top x)^{1/2})\, A.$$

- $g(x)$ can be ℓ_p-norms with $p > 1$ or approximations of ℓ_1-norm.

Our contribution

- Prior works on sketching require that $m \gtrsim d$ (the cost to solve the linear system is $O(d^3)$).
- Sketching dimension m can be **as small as the effective dimension** $\overline{d}_{\mathrm{e}}$ of the Hessian matrix, where

$$\overline{d}_{\mathrm{e}} = \max_x \operatorname{tr}(\nabla^2 f_0(x)(\nabla^2 f_0(x) + \mu I_d)^{-1}).$$

  The cost to solve the linear system is $O(dd_{\mathrm{e}}^2)$.
- Propose an adaptive sketch size algorithm with quadratic convergence rate **without** prior knowledge of the effective dimension.
- Achieve state-of-the-art computational complexity to achieve a δ-accurate solution

$$\mathcal{O}\left(nd\log(\overline{d}_{\mathrm{e}})\log\left(\frac{d}{\delta}\right)\log\left(\log\left(\frac{d}{\delta}\right)\right)\right).$$

Computational complexities comparison

Table: Complexity to achieve δ-accurate solution.

Algorithm	Time complexity	Sketch size	Proba.
Accelerated SVRG	$(nd + d\sqrt{\kappa n})\log(1/\delta)$	-	1
Newton method	$nd^2 \log(\log(1/\delta))$	-	1
Newton sketch	$nd \log(d)\log(1/\delta)$	d	$1 - \frac{1}{d}$
Adaptive Newton sketch	$nd \log(\overline{d}_{\mathrm{e}})\log(\frac{d}{\delta})\log(\log(\frac{d}{\delta}))$	$\frac{d}{\delta}\left(\overline{d}_{\mathrm{e}} + \log(\frac{d}{\delta})\log(\overline{d}_{\mathrm{e}})\right)$	$1 - \frac{1}{\overline{d}_{\mathrm{e}}}$

Adaptive Newton sketch

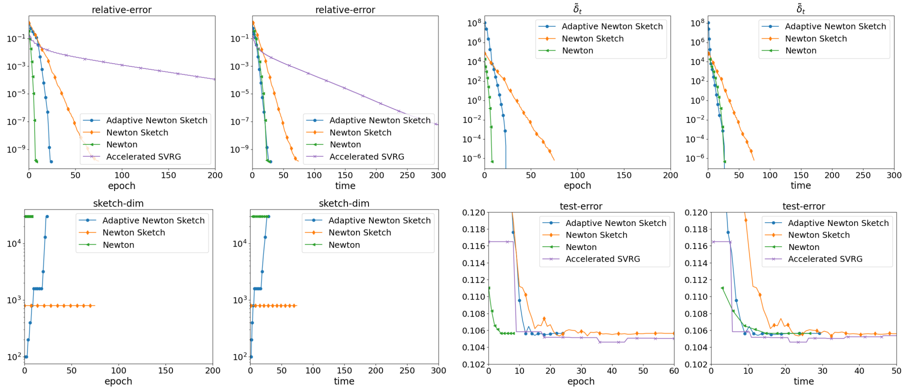
Same idea as for convex quadratic objectives. Start with $m_0 = 1$, $x_0 \in \mathbb{R}^d$ and $S_0 \in \mathbb{R}^{m_0 \times n}$. At each iteration:

- Compute $x_{t+1} = x_t - \mu_t H_{S_t}^{-1} \nabla f(x_t)$.
- Sample $S_{t+1} \in \mathbb{R}^{m_t \times n}$. Form and factorize $H_{S_{t+1}}$.
- Compute improvement ratio $\widetilde{r}_t = \widetilde{\delta}_{t+1} / \widetilde{\delta}_t$ where

$$\widetilde{\delta}_t = \nabla f(x_t)^\top H_{S_t}^{-1} \nabla f(x_t)\,.$$

- If $\widetilde{r}_t$ small enough, accept update x_{t+1}. Otherwise, set $x_{t+1} = x_t$, double sketch size $m_{t+1} = 2m_t$ and sample new $S_{t+1} \in \mathbb{R}^{m_{t+1} \times n}$.

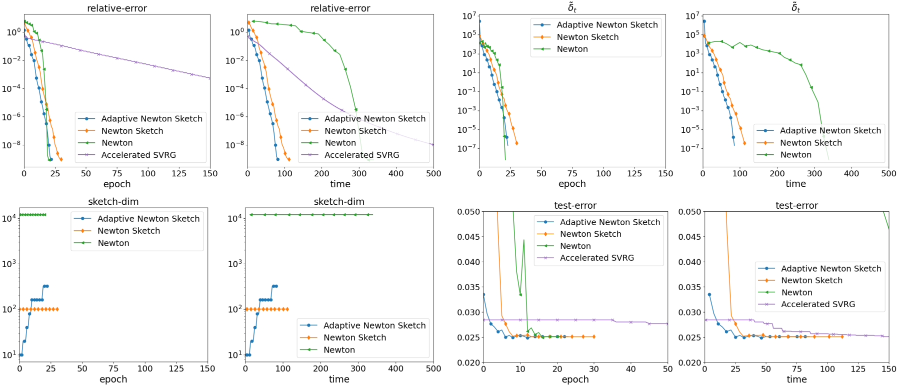
Numerical results



MNIST. $n = 30000, d = 780, \mu = 10^{-1}$.

 Adaptive Newton Sketch

Numerical results



w7a. kernel matrix. $n = 12000, d = 12000, \mu = 10.$