Second Order Stochastic Optimization for Machine Learning in Linear Time [1]

Naman Agarwal, Brian Bullins, and Elad Hazan, 2016

Fan Gao, Huayu Zhang

Introduction

- Machine learning model

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}) \tag{1}$$

$$f(\boldsymbol{w}) = \frac{1}{m} \sum_{k=1}^{m} f_k(\boldsymbol{w}) + R(\boldsymbol{w}) \tag{2}$$

- Second-order optimization methods.

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \nabla^2 f(\boldsymbol{w}^{(t)})^{-1} \nabla f(\boldsymbol{w}^{(t)}) \tag{3}$$

  - faster convergence than first-order methods.
  - prohibitive computation cost $\Omega(md^2 + d^3)$.
- LiSSA (Linear Stochastic Second-Order Algorithm)
  Update Hessian in $O(md)$ time.

Preliminaries

- Denotations
  - $\beta_{max}(\boldsymbol{w}) = \max_k \lambda_{max}(\nabla^2 f_k(\boldsymbol{w})), \alpha_{min}(\boldsymbol{w}) = \min_k \lambda_{min}(\nabla^2 f_k(\boldsymbol{w}))$
  - condition number $\kappa = \frac{\max_{\boldsymbol{w}} \lambda_{max}(\nabla^2 f)}{\min_{\boldsymbol{w}} \lambda_{min}(\nabla^2 f)}$
  - condition number (SVRG) $\hat{\kappa} = \frac{\max_{\boldsymbol{w}} \beta_{max}(\boldsymbol{w})}{\min_{\boldsymbol{w}} \lambda_{min}(\nabla^2 f(\boldsymbol{w}))}$
  - local condition number $\hat{\kappa}_l = \max_{\boldsymbol{w}} \frac{\beta_{max}(\boldsymbol{w})}{\lambda_{min}(\nabla^2 f(\boldsymbol{w}))}, \hat{\kappa}_l^{max} = \max_{\boldsymbol{w}} \frac{\beta_{max}(\boldsymbol{w})}{\alpha_{min}(\boldsymbol{w})}$
- Assumptions
  - f is α-strongly convex and β-smooth.

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T(\boldsymbol{y} - \boldsymbol{x}) + \frac{\alpha}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2$$
$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T(\boldsymbol{y} - \boldsymbol{x}) + \frac{\beta}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2$$

  - ℓ_2 term divided equally and included in f_k.
  - $\frac{\boldsymbol{I}}{\hat{\kappa}_l} \preceq \nabla^2 f_k \preceq \boldsymbol{I} \quad \forall k$
  - $\nabla^2 f$ is M-Lipschitz.

Unbiased estimator

- Taylor expansion (first $j+1$ terms)

$$\nabla^{-2} f_j = \sum_{i=0}^{j} (\boldsymbol{I} - \nabla^2 f)^i \Leftrightarrow \nabla^{-2} f_j = \boldsymbol{I} + (\boldsymbol{I} - \nabla^2 f)\nabla^{-2} f_{j-1}$$

$$\lim_{j\to\infty} \nabla^{-2} f_j = \nabla^{-2} f$$

- Estimator of $\nabla^{-2} f_j$.

$$\tilde{\nabla}^{-2} f_i = \boldsymbol{I} + (\boldsymbol{I} - \nabla^2 f_{s_k})\tilde{\nabla}^{-2} f_{i-1}, i = 1, \ldots, j \; \tilde{\nabla} f_0 = \boldsymbol{I} \tag{4}$$

where s_k is uniformly sampled from $\{1, 2 \ldots, m\}$

- $\tilde{\nabla}^{-2} f_i$ is unbiased.

$$\mathbb{E}[\tilde{\nabla}^{-2} f_j] = \nabla^{-2} f_j$$

$$\lim_{j\to\infty} \mathbb{E}[\tilde{\nabla}^{-2} f_j] = \nabla^{-2} f$$

Proof: take expectation on both sides of Eq. 4, use recursion

$$\mathbb{E}[\tilde{\nabla}^{-2} f_j] = \sum_{i=0}^{j} (\boldsymbol{I} - \nabla^{-2} f)^i = \nabla^{-2} f_j$$

LiSSA

Input: T_1: number of iterations (first order methods). T: number of iterations, $f(\mathbf{w}) = \frac{1}{m}\sum_{k=1}^{m} f_k(\mathbf{w})$, S_1:number of biased estimators, S_2:Order of Taylor expansion

Output: $\mathbf{w}^{(T+1)}$

$\mathbf{w}^{(1)} \leftarrow FO(f(\mathbf{w}), T_1)$;

for $t = 1$ **to** T **do**

  for $i = 1$ **to** S_1 **do**

    $\mathbf{w}'_{i,0} \leftarrow \nabla f(\mathbf{w}^{(t)})$;

    for $j = 1$ **to** S_2 **do**

      Sample $\tilde{\nabla}^2 f_{i,j}(\mathbf{w}^{(t)})$ uniformly from $\{\nabla^2 f_k(\mathbf{w}^{(t)}) \mid k \in [m]\}$

      $\mathbf{w}'_{i,j} \leftarrow \nabla f(\mathbf{w}^{(t)}) + (I - \tilde{\nabla}^2 f_{i,j}(\mathbf{w}^{(t)}))\mathbf{w}'_{i,j-1}$;

    end

  end

  $\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \frac{1}{S_1}\sum_{i=1}^{S_i} \mathbf{w}'_{i,S_2}$;

end

return $\mathbf{w}^{(T+1)}$;

Convergence rate

THEOREM

Set $T_1 = FO(M, \hat{\kappa}_l), S_1 = O((\hat{\kappa}_l^{max})^2 \ln(\frac{d}{\delta}), S_2 \geq 2\hat{\kappa}_l \ln(4\hat{\kappa}_l)$. For every $t \geq T_1$, with probability $1 - \delta$,

$$\|\mathbf{w}^{(t+1)} - \mathbf{w}^\star\| \leq \frac{\|\mathbf{w}^{(t)} - \mathbf{w}^\star\|}{2} \tag{5}$$

where $FO(M, \hat{\kappa}_l)$ is the number of iterations for the first-order algorithm to reaches

$$\|\mathbf{w}^{(1)} - \mathbf{w}^\star\| \leq \frac{1}{4M\hat{\kappa}_l}$$

Computational complexity

$\boldsymbol{w}^{(1)} \leftarrow FO(f(\boldsymbol{w}), T_1)$;
for $t = 1$ **to** T **do**
  for $i = 1$ **to** S_1 **do**
    $\boldsymbol{w}'_{i,0} \leftarrow \nabla f(\boldsymbol{w}^{(t)})$ // $O(md)$ `compute only once`
    for $j = 1$ **to** S_2 **do**
      Sample $\tilde{\nabla}^2 f_{i,j}(\boldsymbol{w}^{(t)})$ uniformly from $\{\nabla^2 f_k(\boldsymbol{w}^{(t)}) \mid k \in [m]\}$
        // $O(d^2)$ `GLM:` $O(d)$ $(\nabla^2 h(\boldsymbol{wx}) \propto \alpha \boldsymbol{xx}^T)$
      $\boldsymbol{w}'_{i,j} \leftarrow \nabla f(\boldsymbol{w}^{(t)}) + (I - \tilde{\nabla}^2 f_{i,j}(\boldsymbol{w}^{(t)}))\boldsymbol{w}'_{i,j-1}$ // $O(d^2)$ `GLM:` $O(d)$
    end
  end
  $\boldsymbol{w}^{(t+1)} \leftarrow \boldsymbol{w}^{(t)} - \frac{1}{S_1}\sum_{i=1}^{S_i} \boldsymbol{w}'_{i,S_2}$ // $O(S_1 d)$
end
return $\boldsymbol{w}^{(T+1)}$;

Each iteration: $O(md + S_1 S_2 d^2)$. For GLM: $O(md + S_1 S_2 d)$.

Total computational time

THEOREM

For a GLM function $f(\boldsymbol{w})$, LiSSA outputs $\boldsymbol{w}^{(t)}$ s.t. with probability at least $1-\delta$,

$$f(\boldsymbol{w}^{(t)}) \leq \min_{\boldsymbol{w}^\star} f(\boldsymbol{w}^\star) + \varepsilon \tag{6}$$

in total time $O((m + (\hat{\kappa}_l^{max})^2 \hat{\kappa}_l) d \ln(\frac{1}{\varepsilon})$. The log factors of $\kappa, d, \frac{1}{\delta}$ are hidden.

Experiments

- Datasets: MNIST (11791x784), CoverType (8214x112), Mushroom(100000x54).
- Loss function: Logistic regression
- Metrics: log-error vs time/epoches
- Parameter settings: $\lambda = 1/m$ or $10/m$, $S_1 = 1$, $S_2 \sim \kappa \ln(\kappa)$
- Comparison: SVRG, SAGA, AdaGrad, BFGS, Gradient Descent, SGD.

Algorithm	Runtime
SVRG,SAGA,SDCA	$(md + O(\hat{\kappa} d)) \log(\frac{1}{\varepsilon})$
LiSSA	$(md + O(\hat{\kappa}_l) S_1) \log(\frac{1}{\varepsilon})$

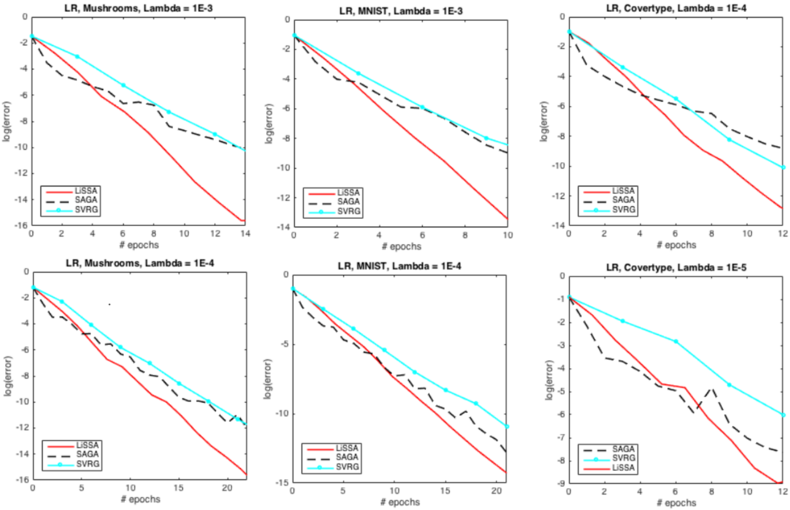

Figure: Performance of LiSSA as compared to a variety of related optimization methods.

Comparison with Newton's method

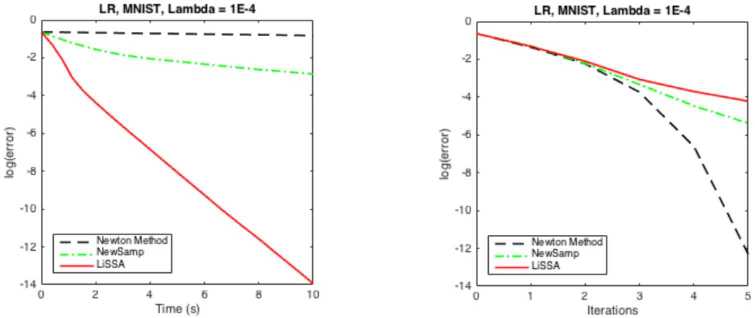

Figure: Convergence of LiSSA over time/iterations for LR with MNIST, as compared to NewSamp and Newtons method.

Running time

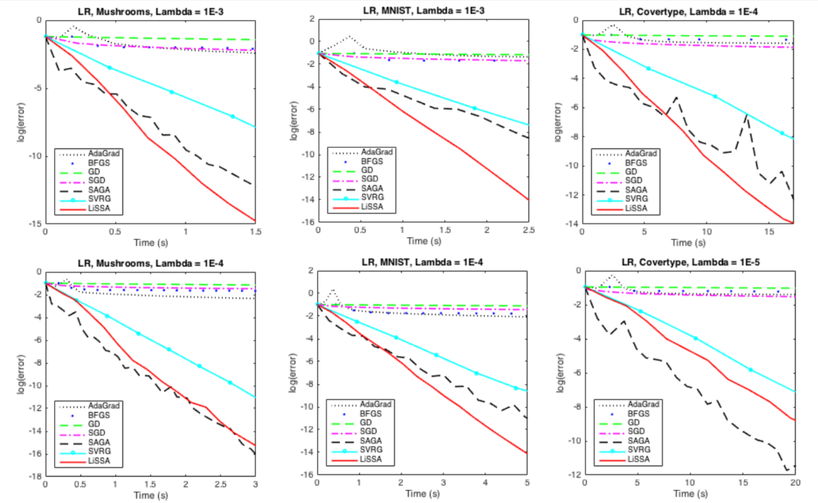

Figure: Performance (running time) of LiSSA as compared to a variety of related optimization methods.

Fine tune S_2

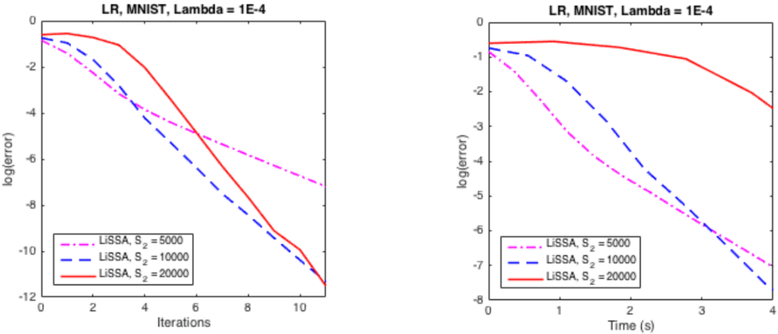

Figure: Differing convergence rates for LiSSA based on different choices of the S_2 parameter.

Main References

Brian Bullins Naman Agarwal and Elad Hazan. “Second Order Stochastic Optimization for Machine Learning in Linear Time”. In: (). arXiv: 1602.03943 [quant-ph].