Adaptive step size momentum method for deconvolution

Trung Vu and Raviv Raich
School of EECS, Oregon State University, Corvallis, OR, 97331-5501, USA
vutru@oregonstate.edu, raich@eecs.oregonstate.edu

1 Introduction

- Deconvolution is the process of reversing the effects of convolution.
- Common iterative algorithms: gradient descent, momentum method, and Newton-Raphson method [1].
- Momentum method: fast convergence, low computational cost, but requires the prior knowledge of the function curvature in choosing the step sizes.
- We propose an adaptive schedule that uses the gradient information to compute the step size for momentum method at each iteration accordingly.
- In a deconvolution setting, the special structure of the objective function allows us to implement the algorithm efficiently without heavy computations of the Hessian.
- Our method asymptotically recovers the optimal convergence rate while only requires **twice** the number of operations per iteration in gradient descent.

2 Problem Formulation (deconvolution)

- Settings: a training set $\{\boldsymbol{x}_m, \boldsymbol{y}_m\}_{m=1}^M$ and a convolution kernel $\boldsymbol{w}$ of size h

$$\boldsymbol{x}_m = [x_m(1), x_m(2), \ldots, x_m(n)]^T \qquad \boldsymbol{x}_{mt} = [x_m(t), x_m(t-1), \ldots, x_m(t-h+1)]^T$$
$$\boldsymbol{y}_m = [y_m(1), y_m(2), \ldots, y_m(n)]^T$$

- Goal: minimize the following objective function

$$f(\boldsymbol{w}) = \sum_{m=1}^{M} \sum_{t=1}^{n} c(\boldsymbol{w}^T \boldsymbol{x}_{mt}, \boldsymbol{y}_m(t)) + \Omega(\boldsymbol{w})$$

- Assumption:

$$0 \preceq \frac{\partial^2 c}{\partial^2 a} \preceq \mu I, \qquad \lambda I \preceq \frac{d^2\Omega}{d\boldsymbol{w} d\boldsymbol{w}^T} \preceq \gamma I, \qquad \forall \mu, \lambda, \gamma > 0$$

- Bounded Hessian:

$$\lambda I \preceq \nabla^2 f(\boldsymbol{w}) \preceq \mu \sum_{m=1}^{M} R_{\hat{x}_m} + \gamma I, \qquad \forall \boldsymbol{w}$$

3 Preliminary (momentum/heavy-ball method)

- Consider the following minimization

$$f(\boldsymbol{w}) : \mathbb{R}^d \to \mathbb{R}, \qquad lI \preceq \nabla^2 f(\boldsymbol{w}) \preceq LI, \qquad \forall \boldsymbol{w}$$

- Momentum updates

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} - \alpha^{(k)} \nabla f(\boldsymbol{w}^{(k)}) + \beta^{(k)} (\boldsymbol{w}^{(k)} - \boldsymbol{w}^{(k-1)}) \qquad (1)$$

- Polyak [2] showed that an optimal convergence rate of $\frac{\sqrt{L}-\sqrt{l}}{\sqrt{L}+\sqrt{l}}$ can be obtained on a quadratic by using **constant** step sizes

$$\alpha^{(k)} = \left(\frac{2}{\sqrt{L}+\sqrt{l}}\right)^2 \qquad \beta^{(k)} = \left(\frac{\sqrt{L}-\sqrt{l}}{\sqrt{L}+\sqrt{l}}\right)^2 \qquad (2)$$

- However, this is not optimal for non-quadratic objectives, where the asymptotic convergence is determined by the local Hessian at the solution [3].

4 Adaptive Step Size Schedule

- Idea: perform line search on the second-order Taylor expansion of the objective function
- Gradient descent

$$f(\boldsymbol{w} - \alpha \nabla f(\boldsymbol{w})) \approx f(\boldsymbol{w}) - \alpha \nabla f(\boldsymbol{w})^T \nabla f(\boldsymbol{w}) + \tfrac{1}{2}\alpha^2 \nabla f(\boldsymbol{w})^T \nabla^2 f(\boldsymbol{w}) \nabla f(\boldsymbol{w})$$
$$\Downarrow \min_{\alpha}(\cdot)$$
$$\alpha^* = \frac{\nabla f(\boldsymbol{w})^T \nabla f(\boldsymbol{w})}{\nabla f(\boldsymbol{w})^T \nabla^2 f(\boldsymbol{w}) \nabla f(\boldsymbol{w})} \qquad (3)$$

- Momentum

$$D = \begin{bmatrix} \nabla f(\boldsymbol{w}) & -\Delta \boldsymbol{w} \end{bmatrix} \qquad f(\boldsymbol{w} - D\boldsymbol{\eta}) \approx f(\boldsymbol{w}) - \nabla f(\boldsymbol{w})^T D\boldsymbol{\eta} + \tfrac{1}{2}\boldsymbol{\eta}^T D^T \nabla^2 f(\boldsymbol{w}) D\boldsymbol{\eta}$$
$$\boldsymbol{\eta} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \qquad \Downarrow \min_{\alpha}(\cdot)$$
$$\boldsymbol{\eta} = \left(D^T \nabla^2 f(\boldsymbol{w}) D\right)^{-1} D^T \nabla f(\boldsymbol{w})$$

$$\Rightarrow \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \nabla f(\boldsymbol{w})^T \nabla^2 f(\boldsymbol{w}) \nabla f(\boldsymbol{w}) & -\Delta \boldsymbol{w}^T \nabla^2 f(\boldsymbol{w}) \nabla f(\boldsymbol{w}) \\ -\Delta \boldsymbol{w}^T \nabla^2 f(\boldsymbol{w}) \nabla f(\boldsymbol{w}) & \Delta \boldsymbol{w}^T \nabla^2 f(\boldsymbol{w}) \Delta \boldsymbol{w} \end{bmatrix}^{-1} \begin{bmatrix} \nabla f(\boldsymbol{w})^T \nabla f(\boldsymbol{w}) \\ -\Delta \boldsymbol{w}^T \nabla f(\boldsymbol{w}) \end{bmatrix} \qquad (4)$$

5 Algorithm

Algorithm 1 Adaptive step size schedule for momentum.

Given initial guess $\boldsymbol{w}^{(0)}$ and $\boldsymbol{w}^{(1)}$.
repeat for $k = 1, 2, \ldots$
  $\Delta \boldsymbol{w} = \boldsymbol{w}^{(k)} - \boldsymbol{w}^{(k-1)}$ ▷ $O(h)$
  $\nabla f = \sum_{m,t} \frac{\partial c}{\partial (\boldsymbol{w}^T \boldsymbol{x}_{mt})} \boldsymbol{x}_{mt} + \lambda \boldsymbol{w}$ ▷ $O(Mnh)$
  for $m = 1, \ldots, M$, $t = 1, \ldots, n$ **do** ▷ $O(Mnh)$
    $p_{mt} = \boldsymbol{x}_{mt}^T \nabla f$, $q_{mt} = \boldsymbol{x}_{mt}^T \Delta \boldsymbol{w}$
    $c_{mt} = \partial^2 c / \partial (\boldsymbol{w}^T \boldsymbol{x}_{mt})^2$
  $u = \nabla f^T \nabla f$, $v = \Delta \boldsymbol{w}^T \nabla f$, $t = \Delta \boldsymbol{w}^T \Delta \boldsymbol{w}$ ▷ $O(h)$
  $a = \sum_{m=1}^{M} \sum_{t=1}^{n} c_{mt} p_{mt}^2 + \lambda u$ ▷ $O(Mn)$
  $b = \sum_{m=1}^{M} \sum_{t=1}^{n} c_{mt} p_{mt} q_{mt} + \lambda v$ ▷ $O(Mn)$
  $d = \sum_{m=1}^{M} \sum_{t=1}^{n} c_{mt} q_{mt}^2 + \lambda t$ ▷ $O(Mn)$
  $\alpha^{(k)} = \frac{du - bv}{ad - b^2}$, $\beta^{(k)} = \frac{bu - av}{ad - b^2}$ ▷ $O(1)$
  Update $\boldsymbol{w}^{(k+1)}$ using (1). ▷ $O(h)$
until convergence

6 Computational Complexity

Table 1: Computational complexity of fixed step size gradient (GD), adaptive step size gradient (AGD), fixed step size momentum (MO), adaptive step size momentum (AMO), and Newton's method. ϵ is the relative accuracy.

Method	# Ops. / Iter.	Cvg. rate	# Iters. needed
GD	$O(Mnh)$	$\frac{r-1}{r+1}$	$\frac{r+1}{2} log(1/\epsilon)$
AGD	$O(Mnh)$	$\frac{\kappa-1}{\kappa+1}$	$\frac{\kappa+1}{2} log(1/\epsilon)$
MO	$O(Mnh)$	$\frac{\sqrt{r}-1}{\sqrt{r}+1}$	$\frac{\sqrt{r}+1}{2} log(1/\epsilon)$
AMO	$O(Mnh)$	$\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$	$\frac{\sqrt{\kappa}+1}{2} log(1/\epsilon)$
Newton	$O(Mnh^3)$	quadratic	$O(1)$

7 Numerical Example

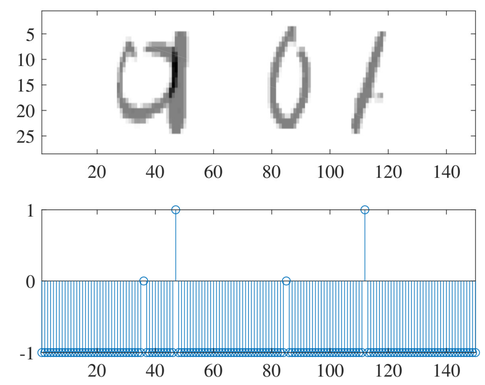

Fig. 1: Top - an image generated by randomly inserting a sequence of $0, 1, 0, 1$. Bottom - the corresponding label series.

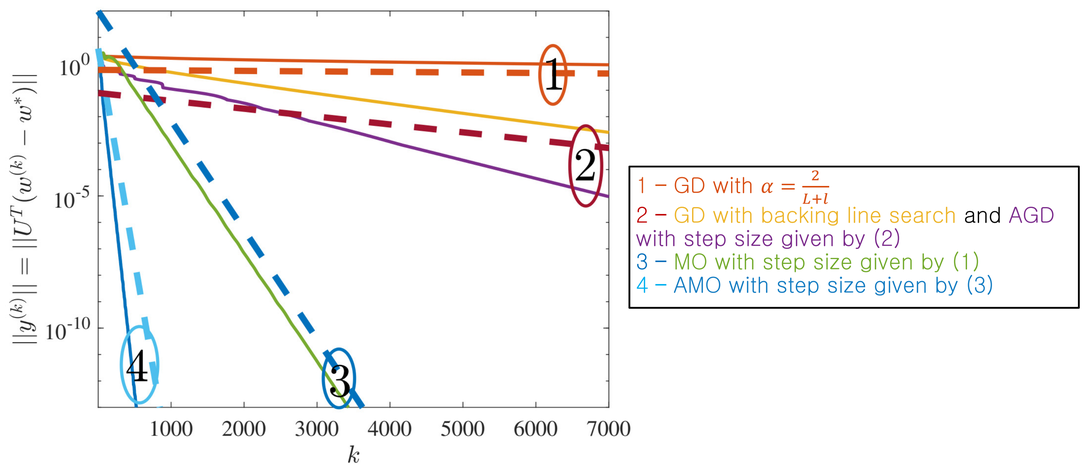

Fig. 2: The log-scale decrease in the distance to the solution on domain value side through iterations.

8 References

1. M. R. Banham and A. K. Katsaggelos, "Digital image restoration," IEEE Signal Processing Magazine, vol. 14, no. 2, pp. 24–41, 1997.
2. B. Polyak, "Some methods of speeding up the convergence of iteration methods," in Ussr Computational Mathematics and Mathematical Physics, vol. 4, no. 1, pp. 1–17, 1964.
3. B. O'Donoghue and E. Candès, "Adaptive restart for accelerated gradient schemes," Foundation of Computational Mathematics, vol. 15, no. 3, pp. 715–732, 2015.

Acknowledgements: This work is partially supported by the National Science Foundation grants CCF-1254218, DBI-1356792.

SSP 2018