Adaptive Gradient Methods

1 Introduction

We consider the following stochastic optimization problem:

$$\min_{w\in C} \phi(w), \quad \phi(w) = f(w) + g(w), \qquad f(w) = \mathbf{E}_{\xi\sim D} f(\xi, w), \tag{1}$$

where ξ is a random variable, drawn from a distribution D.

In this lecture, we consider setting different learning rates for different coordinates. The motivation comes from the sparse data scenario. If a coordinate appears infrequently, then we should set a larger learning rate. On the other hand, if a coordinate appears frequently, then we should set a smaller learning rate. Therefore it is useful to set different learning rates for different coordinates.

2 AdaGrad

The first work on coordinate dependent learning rate is the AdaGrad algorithm of [2]. The original motivation was using time varying proximal functions. We present the algorithm here with a slightly different motivation.

Given a minibatch B, and positive definite matrix Λ, we define

$$f_B(w) = \frac{1}{|B|} \sum_{\xi\in B} f(\xi, w),$$

and

$$\mathrm{prox}_{\Lambda,g}(w) = \arg\min_z \left[\frac{1}{2}(z-w)^\top \Lambda^{-1}(z-w) + g(z)\right].$$

Consider the following general proximal stochastic gradient method below with $B_t \sim D$:

$$w^{(t)} = \mathrm{prox}_{\Lambda_t,g}(w^{(t-1)} - \Lambda_t \nabla f_{B_t}(w^{(t-1)}),$$

where we replace the coordinate independent learning rate η_t by a diagonal matrix $\Lambda_t = \mathrm{diag}(\eta_{t,1}, \ldots, \eta_{t,d})$. Here $\eta_{t,j}$ is the learning rate for the j-th coordinate at time t.

If we define

$$Q_{\Lambda,B}(w; w') = f(w') + \nabla f_B(w')^\top (w - w') + \frac{1}{2}\|w - w'\|^2_{\Lambda^{-1}} + g(w),$$

where

$$\|u\|^2_\Lambda = u^\top \Lambda u.$$

Proposition 1 *Assume that $f(w)$ and $g(w)$ are convex. Then*

$$\phi(w) \leq Q_{\Lambda,B}(w;w') + \frac{1}{2}\|\nabla f_B(w') - \nabla f(w)\|_{\Lambda}^2,$$

and

$$f(w') + g(w) \leq Q_{\Lambda,B}(w;w') + \frac{1}{2}\|\nabla f_B(w')\|_{\Lambda}^2.$$

Proof The first inequality follows from the same argument of Lecture 19. For the second inequality, we note that

$$f(w') + g(w) \leq f(w') + g(w) + \frac{1}{2}\|w - w' + \Lambda \nabla f_B(w')\|_{\Lambda^{-1}}^2 = Q_{\Lambda,B}(w;w') + \frac{1}{2}\|\nabla f_B(w')\|_{\Lambda}^2.$$

This proves the result. ■

Similar to Lecture 19, we can obtain the following theorem.

Theorem 1 *Let $\Delta_j = \max_{w,w' \in C} |w - w'|_j$, and $\Delta = \mathrm{diag}(\Delta_1, \ldots, \Delta_d)$. If we take $\Lambda_1 \geq \Lambda_2 \geq \cdots$, then*

$$\mathbf{E} \sum_{t=1}^{T} \phi(w^{(t-1)}) \leq T\phi(w) + \mathbf{E}[g(w^{(0)}) - g(w^{(T)})] + \frac{1}{2}\mathbf{E}\,\mathrm{trace}(\Delta^2 \Lambda_T^{-1}) + \frac{1}{2}\sum_{t=1}^{T} \mathbf{E}\|\nabla f_{B_t}(w^{(t-1)})\|_{\Lambda_t}^2.$$

Proof We have from Proposition 1

$$\begin{aligned}
f(w^{(t-1)}) + g(w^{(t)}) \leq & Q_{\Lambda_t,B_t}(w^{(t)}; w^{(t-1)}) + \frac{1}{2}\|\nabla f_{B_t}(w^{(t-1)})\|_{\Lambda_t}^2 \\
\leq & Q_{\Lambda_t,B_t}(w; w^{(t-1)}) - \frac{1}{2}(w - w^{(t)})\Lambda_t^{-1}(w - w^{(t)}) + \frac{1}{2}\|\nabla f_{B_t}(w^{(t-1)})\|_{\Lambda_t}^2 \\
\leq & f(w^{(t-1)}) + \nabla f_{B_t}(w^{(t-1)})^\top (w - w^{(t-1)}) + g(w) \\
& + \frac{1}{2}\|w - w^{(t-1)}\|_{\Lambda_t^{-1}}^2 - \frac{1}{2}\|w - w^{(t)}\|_{\Lambda_t^{-1}}^2 + \frac{1}{2}\|\nabla f_{B_t}(w^{(t-1)})\|_{\Lambda_t}^2.
\end{aligned}$$

Note that

$$\mathbf{E}_{B_t}[f(w^{(t-1)}) + \nabla f_{B_t}(w^{(t-1)})^\top (w - w^{(t-1)}) + g(w)] \leq f(w) + g(w) = \phi(w),$$

we obtain

$$\mathbf{E}_{B_t}[f(w^{(t-1)}) + g(w^{(t)})] \leq \phi(w) + \frac{1}{2}\mathbf{E}_{B_t}\left[\|w - w^{(t-1)}\|_{\Lambda_t^{-1}}^2 - \|w - w^{(t)}\|_{\Lambda_t^{-1}}^2\right] + \frac{1}{2}\mathbf{E}_{B_t}\|\nabla f_{B_t}(w^{(t-1)})\|_{\Lambda_t}^2.$$

By summing over t, and noticing that (we take $\Lambda_0^{-1} = 0$):

$$\begin{aligned}
& \sum_{t=1}^{T}\left[\|w - w^{(t-1)}\|_{\Lambda_t^{-1}}^2 - \|w - w^{(t)}\|_{\Lambda_t^{-1}}^2\right] \\
\leq & \sum_{t=1}^{T}\|w - w^{(t-1)}\|_{\Lambda_t^{-1} - \Lambda_{t-1}^{-1}}^2 \leq \sum_{t=1}^{T}\mathrm{trace}(\Delta^2(\Lambda_t^{-1} - \Lambda_{t-1}^{-1})) = \mathrm{trace}(\Delta^2 \Lambda_T^{-1}),
\end{aligned}$$

we obtain the bound. ■

In AdaGrad, we choose a specific Λ_t in Theorem 1, as described in Corollary 1. The algorithm, which employs constant Δ_j, is presented in Algorithm 1, where all vector operations are element-wise operations.

Corollary 1 *If for some $\eta > 0$, we take $\eta_{t,j}$ in Λ_t for $j = 1, \ldots, d$ as*

$$\eta_{t,j} = \frac{\eta \Delta_j}{\sqrt{\epsilon + \sum_{s=1}^{t} [\nabla f_{B_s}(w^{(s-1)})]_j^2}},$$

then

$$\mathbf{E} \sum_{t=1}^{T} \phi(w^{(t-1)}) \leq T\phi(w) + \mathbf{E}[g(w^{(0)}) - g(w^{(T)})] + (0.5\eta^{-1} + \eta)\mathbf{E} \sum_{j=1}^{d} \Delta_j \sqrt{\epsilon + \sum_{s=1}^{T} [\nabla f_{B_s}(w^{(s-1)})]_j^2}.$$

Proof We have

$$\mathrm{trace}(\Delta^2 \Lambda_T^{-1}) = \eta^{-1} \sum_{j=1}^{d} \Delta_j \sqrt{\epsilon + \sum_{s=1}^{T} [\nabla f_{B_s}(w^{(s-1)})]_j^2},$$

and

$$\begin{aligned}
&\sum_{t=1}^{T} \|\nabla f_{B_t}(w^{(t-1)})\|_{\Lambda_t}^2 = \eta^2 \sum_{t=1}^{T} \sum_{j=1}^{d} \Delta_j^2 \frac{\eta_{t,j}^{-2} - \eta_{t-1,j}^{-2}}{\eta_{t,j}^{-1}} \\
\leq & 2\eta^2 \sum_{t=1}^{T} \sum_{j=1}^{d} \Delta_j^2 \frac{\eta_{t,j}^{-2} - \eta_{t-1,j}^{-2}}{\eta_{t,j}^{-1} + \eta_{t-1,j}^{-1}} = 2\eta^2 \sum_{t=1}^{T} \sum_{j=1}^{d} \Delta_j^2 (\eta_{t,j}^{-1} - \eta_{t-1,j}^{-1}) \\
= & 2\eta \mathrm{trace}(\Delta^2 \Lambda_T^{-1}).
\end{aligned}$$

This implies the bound. ■

If we take $\Delta_j = \delta$, then $\delta = \sup\{\|w - w'\|_\infty : w, w' \in C\}$. In comparison, in the coordinate independent bound, we use $\|w - w_0\|_2^2$. Since $\|w - w_t\|_\infty \leq \|w - w_t\|_2$, and the different can be as large as $\sqrt{d}$, we know that Theorem 1 can be significantly better.

Algorithm 1: AdaGrad

Input: $\phi(\cdot)$, learning rates η , $\epsilon > 0$, $w^{(0)}$
Output: $w^{(T)}$
$\tilde{g}_0^2 = [\epsilon, \ldots, \epsilon]$
for $t = 1, 2, \ldots, T$ **do**
  Randomly pick $B_t \sim D$
  Let $g_t = \nabla_w f_{B_t}(w^{(t-1)})$
  Let $\tilde{g}_t^2 = \tilde{g}_{t-1}^2 + g_t^2$
  Let $\Lambda_t = \eta \mathrm{diag}(\tilde{g}_t^{-1})$
  Let $w^{(t)} = \mathrm{prox}_{\Lambda_t g}(w^{(t-1)} - \Lambda_t g_t)$
Return: $w^{(T)}$

AdaGrad can also be used with regularized dual averaging (RDA), as described in [2].

Algorithm 2: AdaGrad-RDA

Input: $f(\cdot)$, $g(\cdot)$, $w^{(0)}$, $\eta_0, \eta_1, \eta_2, \ldots$
$h(w)$ (default is $h(w) = 0.5\eta_0 \|w\|_2^2$)
Output: $w^{(T)}$
Let $\tilde{\alpha}_0 \in \partial h(w^{(0)})$
Let $\tilde{\Lambda}_0 = 0$
$\tilde{g}_0^2 = [\epsilon, \ldots, \epsilon]$
for $t = 1, 2, \ldots, T$ **do**
  Randomly select a minibatch B_t of m independent samples from D
  Let $g_t = \nabla_w f_{B_t}(w^{(t-1)})$
  Let $\tilde{g}_t^2 = \tilde{g}_{t-1}^2 + g_t^2$
  Let $\Lambda_t = \eta \mathrm{diag}(\tilde{g}_t^{-1})$
  Let $\tilde{\alpha}_t = \tilde{\alpha}_{t-1} - \Lambda_t g_t$
  Let $\tilde{\Lambda}_t = \tilde{\Lambda}_{t-1} + \Lambda_t$
  Let $w^{(t)} = \mathrm{prox}_{\tilde{\Lambda}_t g}(\tilde{\alpha}_t)$
Return: $w^{(T)}$

AdaGrad employs a learning rate that is $O(\sqrt{t})$ due to the accumulate of gradient. We may also consider constant learning rates, where we can set $\Lambda_t = \Lambda$ for all t, and the optimal learning rate to optimize the bound is

$$\eta_{t,j}^{-1} \propto \sqrt{\sum_{t=1}^{T} [\nabla f_{B_t}(w^{(t-1)})]_j^2}.$$

We may use a moving average to obtain the estimate, which leads to Algorithm 3.

Algorithm 3: RMSprop

Input: $\phi(\cdot)$, learning rates η , ρ (default is 0.9),$\epsilon > 0$, $w^{(0)}$
Output: $w^{(T)}$
$\tilde{g}_0^2 = [\epsilon, \ldots, \epsilon]$
for $t = 1, 2, \ldots, T$ **do**
  Randomly pick $B_t \sim D$
  Let $g_t = \nabla_w f_{B_t}(w^{(t-1)})$
  Let $\tilde{g}_t^2 = \rho \tilde{g}_{t-1}^2 + (1 - \rho) g_t^2$
  Let $\Lambda_t = \eta \mathrm{diag}(\tilde{g}_t^{-1})$
  Let $w^{(t)} = \mathrm{prox}_{\Lambda_t g}(w^{(t-1)} - \Lambda_t g_t)$
Return: $w^{(T)}$

3 Automatically Tuning of Global Learning Rate

AdaDelta, proposed in [7], can be regarded as a method of coordinate-wise tuning of learning rates for RMSprop. It can be stated in Algorithm 4.

Algorithm 4: AdaDelta

Input: $\phi(\cdot)$, learning rates η, ρ, $\epsilon > 0$, $w^{(0)}$
Output: $w^{(T)}$
Let $\tilde{g}_0^2 = 0$
Let $\eta_0^2 = 0$
for $t = 1, 2, \ldots, T$ **do**
  Randomly pick $B_t \sim D$
  Let $g_t = \nabla_w f_{B_t}(w^{(t-1)})$
  Let $\tilde{g}_t^2 = \rho \tilde{g}_{t-1}^2 + (1-\rho) g_t^2$
  Let $\Lambda_t = \operatorname{diag}\left(\sqrt{\epsilon + \eta_{t-1}^2} / \sqrt{\epsilon + \tilde{g}_t^2}\right)$
  Let $w^{(t)} = \operatorname{prox}_{\Lambda_t g}(w^{(t-1)} - \Lambda_t g_t)$
  Let $\eta_t^2 = \rho \eta_{t-1}^2 + (1-\rho)(w^{(t)} - w^{(t-1)})^2$
Return: $w^{(T)}$

One problem of AdaDelta is that it does not have a solid theoretical justification. It is closely related to Corollary 1, because it uses $|w(t) - w^{(t-1)}|$ to approximate Δ. We may also employ Corollary 1 directly, and compute Δ every epoch. This leads to an algorithm we call AdaMD.

Algorithm 5: AdaMD

Input: $f(\cdot)$, $g(\cdot)$, $w^{(0)}$, η_0, c, p (default is $\lceil n/m \rceil$)
Output: $w^{(T)}$
Let $\Lambda_0 = \eta_0 \Lambda$
Let $\tilde{g}^2 = 0$
Let $w_{\min} = w^{(0)}$
Let $w_{\max} = w^{(0)}$
Let $q = 0$
for $t = 1, 2, \ldots, T$ **do**
  Randomly pick $B_t \sim D$
  Let $g_t = \nabla_w f_{B_t}(w^{(t-1)})$
  Let $\tilde{g}^2 = \tilde{g}^2 + g_t^2$
  Let $\Lambda_t = \Lambda_{t-1}$
  Let $w^{(t)} = \operatorname{prox}_{\Lambda_t g}(w^{(t-1)} - \Lambda_t g_t)$
  Let $w_{\min} = \min(w_{\min}, w^{(t)})$
  Let $w_{\max} = \max(w_{\max}, w^{(t)})$
  Let $q = q + 1$
  if $q >= p$ **then**
    Let $\Lambda_t = \operatorname{diag}((c(w_{\max} - w_{\min}) + \sqrt{\epsilon}) / \sqrt{\epsilon + \tilde{g}^2})$
    Let $\tilde{g}^2 = 0$
    Let $w_{\min} = w^{(t)}$
    Let $w_{\max} = w^{(t)}$
    Let $q = 0$
Return: $w^{(T)}$

4 Empirical Studies

We study the smoothed hinge loss function $\phi_\gamma(z)$ with $\gamma = 1$, and solves the following $L_1 - L_2$ regularization problem:

$$\min_w \left[\underbrace{\frac{1}{n}\sum_{i=1}^{n} \phi_\gamma(w^\top x_i y_i)}_{f(w)} + \underbrace{\frac{\lambda}{2}\|w\|_2^2 + \mu\|w\|_1}_{g(w)} \right].$$

We compare different adaptive gradient algorithms.

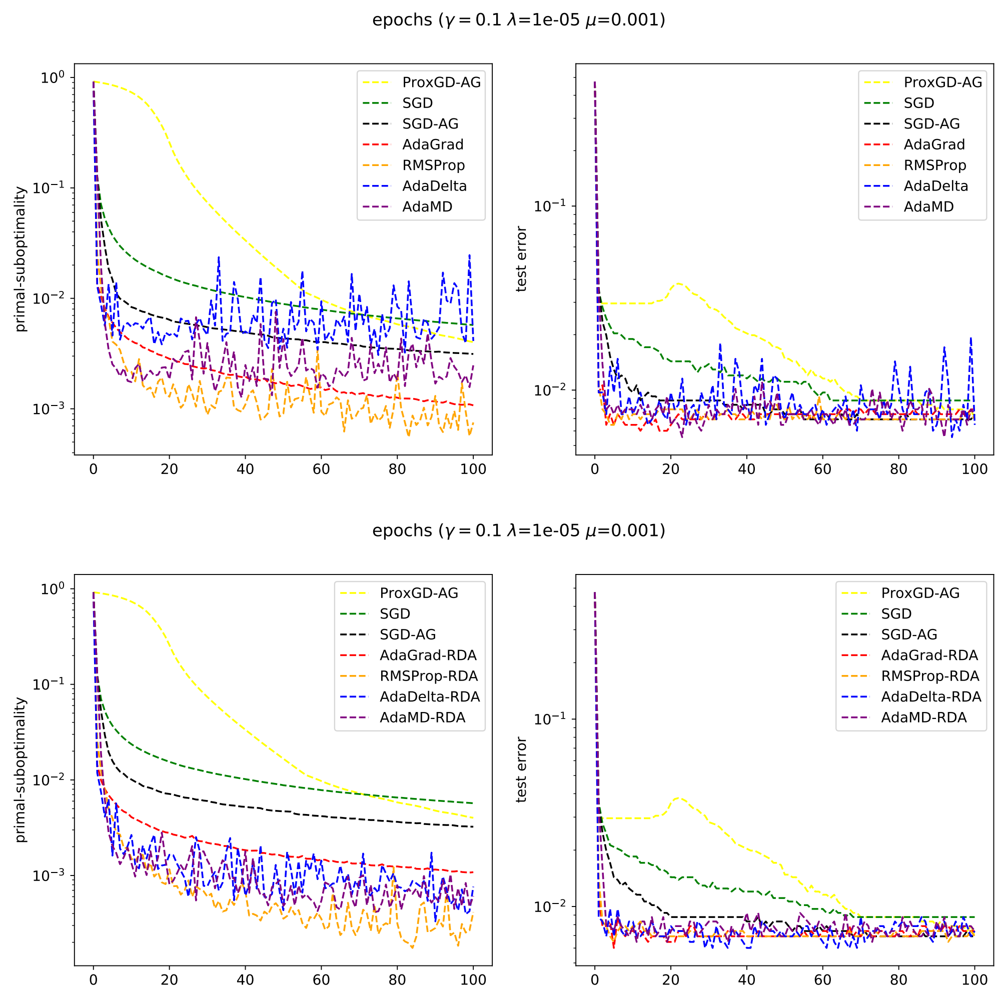

Figure 1: Comparisons of different stochastic algorithms with proximal terms

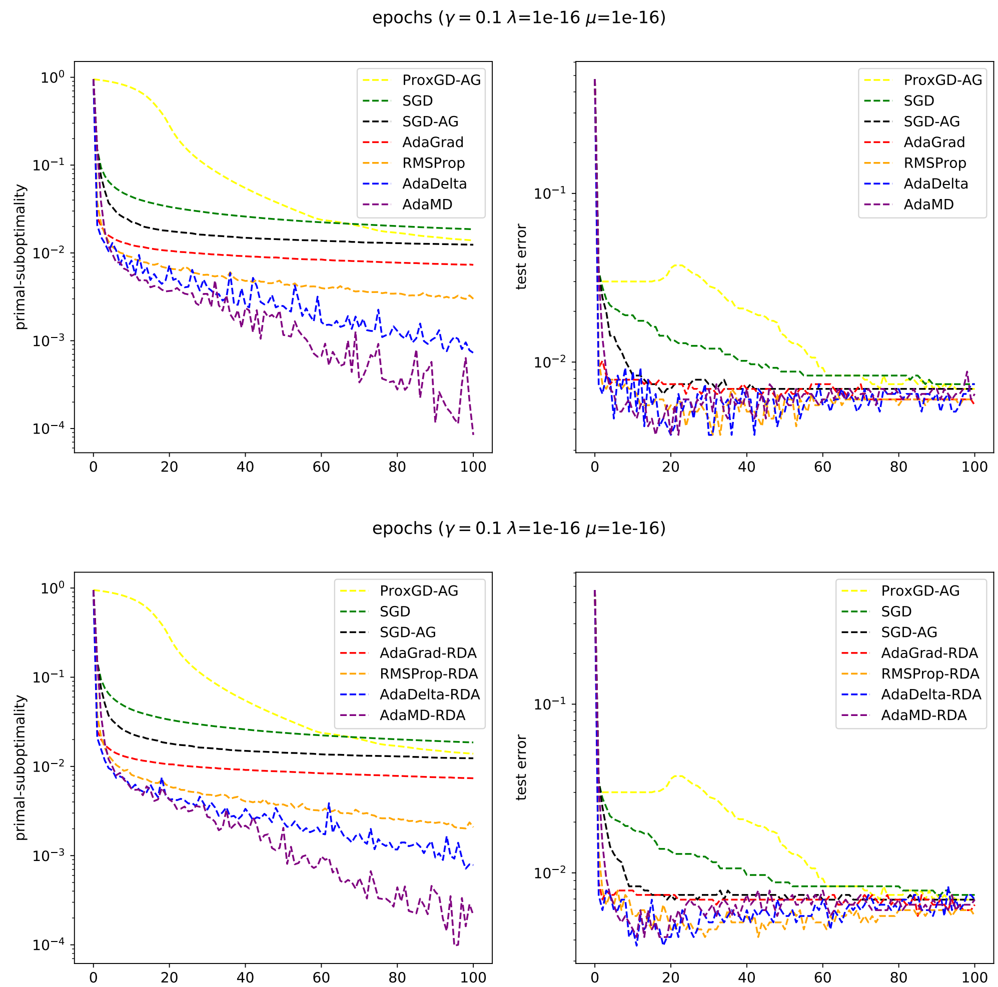

Figure 2: Comparisons of different stochastic algorithms (without proximal terms)

References

[1] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS' 14*, 2014.

[2] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12:2121–2159, 2011.

[3] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS' 13*, 2013.

[4] Atsushi Nitanda. Stochastic proximal gradient descent with acceleration techniques. In *NIPS' 14*. 2014.

[5] Shai Shalev-Shwartz and Tong Zhang. Accelerated mini-batch stochastic dual coordinate ascent. In *NIPS' 13*. 2013.

[6] Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24:2057–2075, 2014.

[7] Matthew Zeiler. Adadelta: an adaptive learning rate method. Technical Report arXiv:1212.5701, arXiv preprint, 2012.