Comp6211e: Optimization for Machine Learning

Tong Zhang

Lecture 24: Applications of Adaptive Gradient Methods

Stochastic Optimization in Machine Learning

We can write this optimization problem as:

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

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

Coordinate-wise Learning Rate

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.

CTR Problem

Large scale logistic regression is useful for CTR (click through rate) estimation problem in computational advertising.
One of the standard approaches is to update models online in real time, to solve linear logistic regression:

$$\sum_{t=1}^{T} \log(1 + \exp(-w^\top x_t y_t)) + \frac{\lambda}{2}\|w\|_2^2 + \mu\|w\|_1,$$

where $y_t \in \{\pm 1\}$ indicates whether an ad is clicked or not. These are generally huge models with billions of parameters, and each feature x_t is sparse.
Linear methods are still widely used, although deep learning models have been applied in various situation.

FTRL (followed the regularized leader)

Algorithm 1: FTRL (follow the regularized leader)

```
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 $\tilde{\alpha}_0 \in \partial h(w^{(0)})$
$\tilde{g}_0^2 = [\epsilon, \ldots, \epsilon]$
$\bar{g}_0^2 = 0$
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 $\bar{g}_t = \bar{g}_{t-1} + g_t - (\Lambda_t^{-1} - \Lambda_{t-1}^{-1}) w^{(t-1)}$
    Let $w^{(t)} = \mathrm{prox}_{\Lambda_t g}(-\Lambda_t \bar{g}_t)$
Return: $w^{(T)}$
```

AdaGrad-RDA (from last lecture)

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: $w^{(T)}$
```

Original AdaGrad-RDA (from AdaGrad paper)

Algorithm 3: AdaGrad-RDA

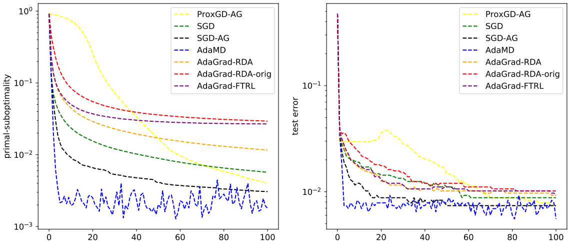
```
Input: $f(\cdot)$, $g(\cdot)$, $w^{(0)}$, $\eta_0, \eta_1, \eta_2, \ldots$
    $h(w)$ (default is $h(w) = 0.5\eta_0 \|w\|_2^2$)
Output: $w^{(T)}$
Let $\tilde{\alpha}_0 \in \partial h(w^{(0)})$
$\tilde{g}_0^2 = [\epsilon, \ldots, \epsilon]$
$\bar{g}_0^2 = 0$
for $t = 1, 2, \ldots, T$ do
    Randomly 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 $\bar{g}_t = \bar{g}_{t-1} + g_t$
    Let $w^{(t)} = \mathrm{prox}_{t\Lambda_t g}(-\Lambda_t \bar{g}_t)$
Return: $w^{(T)}$
```

FTRL

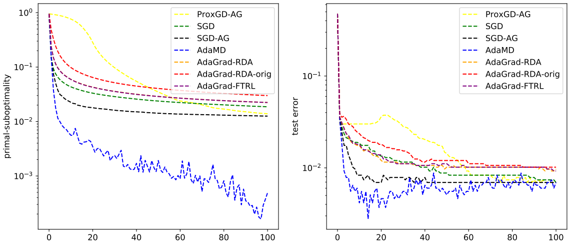
Algorithm 4: FTRL (follow the regularized leader)

```
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 $\tilde{\alpha}_0 \in \partial h(w^{(0)})$
$\tilde{g}_0^2 = [\epsilon, \ldots, \epsilon]$
$\bar{g}_0^2 = 0$
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 $\bar{g}_t = \bar{g}_{t-1} + g_t - (\Lambda_t^{-1} - \Lambda_{t-1}^{-1}) w^{(t-1)}$
    Let $w^{(t)} = \mathrm{prox}_{t\Lambda_t g}(-\Lambda_t \bar{g}_t)$
Return: $w^{(T)}$
```

Experiments



epochs ($\gamma = 0.1$ λ=1e-05 μ=0.001)



epochs ($\gamma = 0.1$ λ=1e-16 μ=1e-16)

AdaGrad with Acceleration

```
Algorithm 4: ADAM (without proximal term)
Input: $\phi(\cdot)$, learning rates $\eta$ , $\beta$ (default is 0.9), $\rho$ (default is 0.999),
       $\epsilon > 0$ (default is $10^{-8}$ , $w^{(0)}$
Output: $w^{(T)}$
Let $\tilde{g}_0^2 = 0$
Let $\bar{g}_0 = 0$
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 $\bar{g}_t = \beta \bar{g}_{t-1} + g_t$
    Let $\Lambda_t = \eta \mathrm{diag}((\tilde{g}_t/(1-\rho)+\epsilon)^{-1})$
    Let $w^{(t)} = w^{(t-1)} - \Lambda_t \bar{g}_t$
Return: $w^{(T)}$
```

Proximal Version

Algorithm 5: Prox-ADAM

```
Input: $\phi(\cdot)$, learning rates $\eta$ , $\beta$ (default is 0.9), $\rho$ (default is 0.999),
       $\epsilon > 0$ (default is $10^{-8}$ , $w^{(0)}$
Output: $w^{(T)}$
Let $\tilde{g}_0^2 = 0$
Let $\bar{g}_0 = 0$
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 $\bar{g}_t = \beta\bar{g}_{t-1} + g_t$
    Let $\Lambda_t = \eta\mathrm{diag}((\tilde{g}_t + \epsilon)^{-1})$
    Let $w^{(t)} = \mathrm{prox}_{\Lambda_t g}(w^{(t-1)} - \Lambda_t\bar{g}_t)$
    $\bar{g}_t = \Lambda_t^{-1}(w^{(t-1)} - w^{(t)})$
Return: $w^{(T)}$
```

Heavy-Ball Version

```
Algorithm 6: Prox-AdaHB
Input: $\phi(\cdot)$, learning rates $\eta$ , $\beta$ (default is 0.9), $\rho$ (default is 0.999),
       $\epsilon > 0$ (default is $10^{-8}$ , $w^{(0)}$
Output: $w^{(T)}$
Let $\tilde{g}_0^2 = 0$
Let $p_0 = 0$
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 $\Lambda_t = \eta \mathrm{diag}((\tilde{g}_t + \epsilon)^{-1})$
    Let $w^{(t)} = \mathrm{prox}_{\Lambda_t g}(w^{(t-1)} + \beta p_{t-1} - \Lambda_t g_t)$
    Let $p_t = w^{(t)} - w^{(t-1)}$
Return: $w^{(T)}$
```

Nesterov's Acceleration

```
Algorithm 7: Prox-AdaACCL
Input: $\phi(\cdot)$, learning rates $\eta$ , $\beta$ (default is 0.9), $\rho$ (default is 0.999),
       $\epsilon > 0$ (default is $10^{-8}$ , $w^{(0)}$
Output: $w^{(T)}$
$\tilde{g}_0^2 = 0$
$p_0 = 0$
for $t = 1, 2, \ldots, T$ do
    Randomly pick $B_t \sim D$
    Let $v^{(t)} = w^{(t-1)} + \beta(w^{(t-1)} - w^{(t-2)})$
    Let $g_t = \nabla_w f_{B_t}(v^{(t-1)})$
    Let $\tilde{g}_t^2 = \rho\tilde{g}_{t-1}^2 + (1-\rho)g_t^2$
    Let $\Lambda_t = \eta\mathrm{diag}((\tilde{g}_t + \epsilon)^{-1})$
    Let $w^{(t)} = \mathrm{prox}_{\Lambda_t g}(v^{(t)} - \Lambda_t g_t)$
Return: $w^{(T)}$
```

RDA version

Algorithm 8: Prox-AdaACCL-RDA

```
Input: $\phi(\cdot)$, learning rates $\eta$ , $\beta$ (default is 0.9), $\rho$ (default is 0.999), $\epsilon > 0$ (default is $10^{-8}$ , $w^{(0)}$
Output: $w^{(T)}$
Let $\tilde{g}_0^2 = 0$
Let $u^{(0)} = w^{(0)}$
Let $\tilde{\alpha}_0 = 0$
Let $\tilde{\Lambda}_0 = 0$
for $t = 1, 2, \ldots, T$ do
    Randomly pick $B_t \sim D$
    Let $v^{(t)} = \beta w^{(t-1)} + (1 - \beta) u^{(t-1)}$
    Let $g_t = \nabla_w f_{B_t}(v^{(t)})$
    Let $\tilde{g}_t^2 = \rho \tilde{g}_{t-1}^2 + (1 - \rho) g_t^2$
    Let $\Lambda_t = \frac{\eta}{1-\beta} \mathrm{diag}((\tilde{g}_t + \epsilon)^{-1})$
    Let $\tilde{\alpha}_t = \tilde{\alpha}_{t-1} - \Lambda_t g_t$
    Let $\tilde{\Lambda}_t = \tilde{\Lambda}_{t-1} + \Lambda_t$
    Let $u^{(t)} = \mathrm{prox}_{\tilde{\Lambda}_t g}(\tilde{\alpha}_t)$
    Let $w^{(t)} = \beta w^{(t-1)} + (1 - \beta) u^{(t)}$
Return: $w^{(T)}$
```

ADAM with Nesterov Acceleration (v3)

Algorithm 9: Prox-AdaACCL-v3

```
Input: $\phi(\cdot)$, learning rates $\eta$ , $\beta$ (default is 0.9), $\rho$ (default is 0.999), $\epsilon > 0$ (default is $10^{-8}$ , $w^{(0)}$
Output: $w^{(T)}$
Let $\tilde{g}_0^2 = 0$
Let $u^{(0)} = w^{(0)}$
Let $\tilde{\alpha}_0 = 0$
Let $\tilde{\Lambda}_0 = 0$
for $t = 1, 2, \ldots, T$ do
    Randomly pick $B_t \sim D$
    Let $v^{(t)} = \beta w^{(t-1)} + (1-\beta) u^{(t-1)}$
    Let $g_t = \nabla_w f_{B_t}(v^{(t)})$
    Let $\tilde{g}_t^2 = \rho \tilde{g}_{t-1}^2 + (1-\rho) g_t^2$
    Let $\Lambda_t = \frac{\eta}{1-\beta} \mathrm{diag}((\tilde{g}_t + \epsilon)^{-1})$
    Let $u^{(t)} = \mathrm{prox}_{\Lambda_t g}(u^{(t-1)} - \Lambda_t g_t)$
    Let $w^{(t)} = \beta w^{(t-1)} + (1-\beta) u^{(t)}$
Return: $w^{(T)}$
```

Empirical Studies

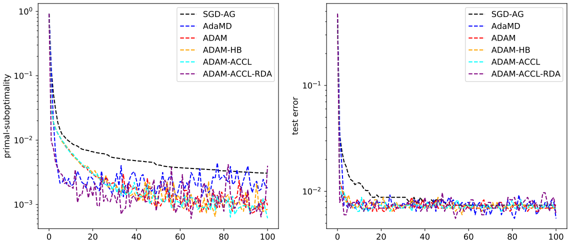
We study the smoothed hinge loss function $\phi_\gamma(z)$ with $\gamma = 1$, and solves the following $L_1 - L_2$ regularization problem:

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

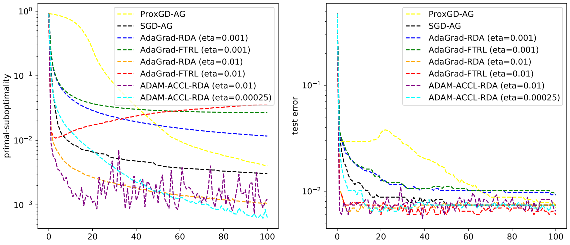
We compare different algorithms with constant learning rate.

Comparisons (proximal)

epochs ($\gamma = 0.1$ λ=1e-05 μ=0.001)

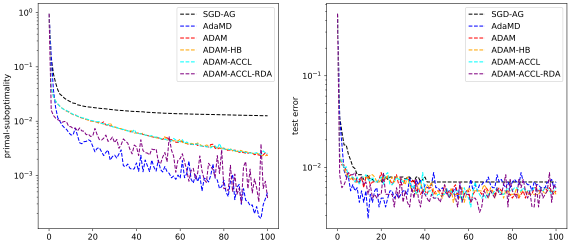


epochs ($\gamma = 0.1$ λ=1e-05 μ=0.001)

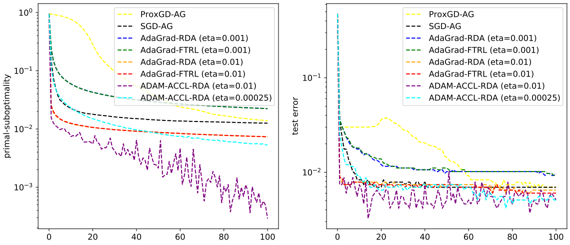


Comparisons (no-proximal)

epochs ($\gamma = 0.1$ λ=1e-16 μ=1e-16)



epochs ($\gamma = 0.1$ λ=1e-16 μ=1e-16)



Summary

This course introduced basic concepts used in optimization for machine learning

- First order gradient methods
- Acceleration
- Primal dual methods
- Coordinate descent
- Proximal methods
- Dual averaging
- Variance reduction
- Coordinate-wise Learning rate (AdaGrad)
- ...

By correctly combining these concepts, we can get very effective algorithms for machine learning.