Stochastic Adaptive Learning Rate and Acceleration 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 the finite sample case, one may consider ξ as i, and the distribution D is to randomly choosing $\xi=i$ from $1,\ldots,n$.

In this lecture, we consider the automatic tuning of learning rate for SGD, and acceleration methods for stochastic gradient algorithms.

2 Adaptive Learning Rate for Stochastic Gradient Descent

When the variance is relatively small with respect to the gradient, we and learning rate is small, each iteration of SGD behaves more like gradient descent. In this case, we can apply a variation of adaptive learning rate method to set learning rate of SGD. In order to derive the underlying algorithm, we will consider the case that $f(w)$ is L-smooth, and $g(w)=0$.

Consider a minibatch B, and let

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

and let

$$V=\mathbf{E}_\xi\|\nabla_w f(\xi,w)-\nabla_w f(w)\|_2^2.$$

Now consider the SGD update rule

$$w'=w-\eta\nabla f_B(w),$$

then

$$\begin{aligned}\mathbf{E}_B f(w') =& \mathbf{E}_B f(w-\eta\nabla f_B(w))\\ \leq& \mathbf{E}_B\left[f(w)-\eta\nabla f(w)^\top\nabla f_B(w)+\frac{\eta^2 L}{2}\mathbf{E}_B\|\nabla f_B(w)\|_2^2\right]\\ =& f(w)-(\eta-0.5\eta^2 L)\|\nabla f(w)\|_2^2+\frac{\eta^2 L}{2m}V.\end{aligned}$$

If we choose

$$\eta \leq \frac{\|\nabla f(w)\|_2^2}{L(\|\nabla f(w)\|_2^2 + V/m)},$$

then we have convergence rate of

$$\mathbf{E}_B f(w') = f(w) - 0.5\eta\|\nabla f(w)\|_2^2. \tag{2}$$

If

$$V \leq (n-m)\ \|\nabla f(w)\|_2^2, \tag{3}$$

then (2) holds as long as

$$\eta \leq m/(nL).$$

Since each minibatch SGD requires m gradient computations, per sample error reduction as in (2) becomes

$$\frac{\eta}{m}\|\nabla f(w)\|_2^2 \leq \frac{\|\nabla f(w)\|_2^2}{nL}.$$

which is the same per sample function value reduction as GD with step size $\eta = 1/L$:

$$f(w - \eta\nabla f(w)) \leq f(w) - \eta\nabla f(w)^\top \nabla f(w) + \frac{\eta^2 L}{2}\|\nabla f(w)\|_2^2 \leq f(w) - 0.5\eta\|\nabla f(w)\|_2^2.$$

Since each GD requires n gradient evaluations, the per sample error reduction is

$$\frac{\|\nabla f(w)\|_2^2}{nL}.$$

This implies that SGD is superior to GD when (3) holds. Since variance is constant, we know that SGD has a faster convergence than GD when

$$\|\nabla f(w)\|_2^2 \geq V/(n-m).$$

When $\|\nabla f(w)\| \ll V/(n-m)$, then per step function value reduction is larger with GD, and thus GD is superior.

In order to set the correct learning rate without known L, we can work with the following result, which generalizes the Armijo-Goldstein condition to the stochastic setting.

Proposition 1 (Stochastic AG Criterion) *Consider the SGD update rule:*

$$w' = w - \eta\nabla f_B(w).$$

Let

$$V_{B',B} = \left[f_{B'}(w - \eta\nabla f_B(w)) - \eta\nabla f_{B'}(w)^\top \nabla f_B(w) - f_{B'}(w)\right].$$

If for some $c < 1$,

$$\mathbf{E}_{B',B}\ V_{B',B} \leq c\eta\|\nabla f(w)\|_2^2,$$

then

$$\mathbf{E}_B f(w') \leq f(w) - (1-c)\eta\|\nabla f(w)\|_2^2.$$

Proof

$$\begin{aligned}\mathbf{E}_B f(w') =& \mathbf{E}_B \left[f(w) - \eta \nabla f(w)^\top \nabla_B(w) + D_f(w', w) \right] \\ =& f(w) - \eta \|\nabla f(w)\|_2^2 + \mathbf{E}_{B,B'} \left[f_{B'}(w) - \eta \nabla f(w)^\top \nabla_B(w) + D_f(w', w) \right] \\ =& f(w) - \eta \|\nabla f(w)\|_2^2 + \mathbf{E}_{B,B'} V_{B',B},\end{aligned}$$

■

When $g(w) \neq 0$, we may easily generalize the Proposition 1 by using gradient mapping as follows.

$$\begin{aligned}\text{prox}_{\eta g}(w) =& \arg\min_z \left[\frac{1}{2\eta} \|z - w\|_2^2 + g(z) \right] \\ D_\eta \phi(w) =& \frac{1}{\eta}(w - \text{prox}_{\eta g}(w - \eta \nabla f(w))) \\ D_{\eta,B} \phi(w) =& \frac{1}{\eta}(w - \text{prox}_{\eta g}(w - \eta \nabla f_B(w))).\end{aligned}$$

With this modification, we obtain the adaptive learning rate method for stochastic gradient descent in Algorithm 1.

Algorithm 1: Stochastic Proximal Gradient Descent with Adaptive Learning Rate

Input: $f(\cdot)$, $g(\cdot)$, w_0, η_0, p (default is $\lceil n/m \rceil$)
Output: $w^{(T)}$

Let $\eta_1 = \eta_0$
Let $q = 0$
Let $V = 0$
Let Randomly select a minibatch B_0 independent samples from D
Let $\tilde{g} = D_{\eta_0, B_0} \phi(w_0)$
for $t = 1, 2, \ldots, T$ **do**
    Randomly select a minibatch B_t of m independent samples from D
    Let $w^{(t)} = \text{prox}_{\eta_t g}(w^{(t-1)} - \eta_t \nabla f_{B_t}(w^{(t-1)}))$
    Let $V = V + [f_{B_t}(w^{(t)} - \eta_t \tilde{g}) - f_{B_t}(w^{(t)}) - \eta_t \nabla f_{B_t}(w^{(t)})^\top \tilde{g}]$
    Let $\tilde{g} = (w^{(t-1)} - w^{(t)})/\eta_t$
    Let $\eta_{t+1} = \eta_t$
    Let $q = q + 1$
    if $q >= p$ **then**
        Let $\rho = \min(0.5, \max(2, qc\eta_t \|D_{\eta_t} \phi(w_t)\|_2^2 / V))$
        Let $\eta_{t+1} = \eta_t \rho$
        Let $q = 0$
        Let $V = 0$
        Let $\tilde{g} = D_{\eta_{t+1}, B_t} \phi(w_t)$

Return: $w^{(T)}$

3 Stochastic Accelerated Gradient

We can generalize Nesterov's accelerated gradient to the stochastic setting in Algorithm 2.

We may take $\beta_t = \beta < 1$ be a constant. Another choice is $\beta_t = 1 - 3/(t+2)$, which may work better in the beginning.

Algorithm 2: Stochastic Accelerated Proximal Gradient Descent

Input: $f(\cdot)$, $g(\cdot)$, $\{\eta_t, \beta_t\}$
Output: $w^{(T)}$

for $t = 1, 2, \ldots, T$ **do**
  Let $z^{(t)} = w^{(t-1)} + \beta_t(w^{(t-1)} - w^{(t-2)})$
  Randomly select a minibatch B_t of m independent samples from D
  Let $w^{(t)} = \text{prox}_{\eta_t g}(z^{(t)} - \eta_t \nabla f_{B_t}(z^{(t)}))$

Return: $w^{(T)}$

We can apply adaptive learning rate method to the accelerated SGD in Algorithm 3.

Algorithm 3: Stochastic Accelerated Proximal Gradient with Adaptive Learning Rate

Input: $f(\cdot)$, $g(\cdot)$, w_0, η_0, β, p (default is $\lceil n/m \rceil$)
Output: $w^{(T)}$

Let $\eta_1 = \eta_0$
Let $q = 0$
Let $V = 0$
Let Randomly select a minibatch B_0 independent samples from D
Let $\tilde{g} = D_{\eta_0, B_0}\phi(w_0)$
for $t = 1, 2, \ldots, T$ **do**
  Let $z^{(t)} = w^{(t-1)} + \beta(w^{(t-1)} - w^{(t-2)})$
  Randomly select a minibatch B_t of m independent samples from D
  Let $w^{(t)} = \text{prox}_{\eta_t g}(z^{(t)} - \eta_t \nabla f_{B_t}(z^{(t)}))$
  Let $V = V + [f_{B_t}(z^{(t)} - \eta_t \tilde{g}) - f_{B_t}(z^{(t)}) - \eta_t \nabla f_{B_t}(z^{(t)})^\top \tilde{g}]$
  Let $\tilde{g} = (z^{(t)} - w^{(t)})/\eta_t$
  Let $\eta_{t+1} = \eta_t$
  Let $q = q + 1$
  if $q >= p$ **then**
    Let $\rho = \max(0.5, \min(2, qc(1-\beta)\eta_t \|D_{\eta_t}\phi(w_t)\|_2^2 / V))$
    Let $\eta_{t+1} = \eta_t \rho$
    Let $q = 0$
    Let $V = 0$
    Let $\tilde{g} = D_{\eta_{t+1}, B_t}\phi(w_t)$

Return: $w^{(T)}$

We may also generalize the heavy-ball method in Algorithm 4. If $g = 0$, then we can use a different implementation of the heavy-ball method:

$$\begin{aligned} p_t &= \beta_t p_{t-1} - \eta_t \nabla f_B(w_{t-1}) \\ w_t &= w_{t-1} + p_t, \end{aligned}$$

which is often referred to as the momentum method.

Algorithm 4: Stochastic Heavy-Ball Gradient Descent

Input: $f(\cdot)$, $g(\cdot)$, $\{\eta_t, \beta_t\}$
Output: $w^{(T)}$

for $t = 1, 2, \ldots, T$ **do**
  Let $z^{(t)} = w^{(t-1)} + \beta_t(w^{(t-1)} - w^{(t-2)})$
  Randomly select a minibatch B_t of m independent samples from D
  Let $w^{(t)} = \text{prox}_{\eta_t g}(z^{(t)} - \eta_t \nabla f_{B_t}(w^{(t-1)}))$

Return: $w^{(T)}$

4 Stochastic Accelerated RDA

One may accelerate regularized dual averaging (RDA). We have two set of tuning parameters: θ_t and $\tilde{\eta}$. In practice, we may take $\theta_t = \theta$ be a constant, or take the default choice of $\theta_t = 2/(t+2)$. The tuning of learning rate $\tilde{\eta}_t$ is more complex for this algorithm.

Algorithm 5: Stochastic Accelerated Regularized Dual Averaging

Input: $f(\cdot)$, $g(\cdot)$, $w^{(0)}$, $\tilde{\eta}_0 \le \tilde{\eta}_1, \ldots, \theta_t)$
  $h(w)$ (default is $h(w) = 0.5\|w\|_2^2$)
Output: $w^{(T)}$

Let $\tilde{\alpha}_0 \in \partial h(w^{(0)})$
Let $v^{(0)} = w^{(0)}$
for $t = 1, 2, \ldots, T$ **do**
  Let $u^{(t)} = (1-\theta_t)w^{(t-1)} + \theta_t v^{(t-1)}$
  Randomly select a minibatch B_t of m independent samples from D
  Let $\tilde{\alpha}_t = (1-\theta_t)\tilde{\alpha}_{t-1} - \theta_t \nabla f_B(u^{(t)})$
  Let $v^{(t)} = \arg\min_w \left[-\tilde{\alpha}_t^\top w + \tilde{\eta}_t^{-1} h(w) + g(w)\right]$
  Let $w^{(t)} = (1-\theta_t)w^{(t-1)} + \theta_t v^{(t)}$

Return: $w^{(T)}$

5 Stochastic Accelerated Linearized ADMM

We may consider the following stochastic dual decomposition problem:

$$\phi(w, z) = f(w) + g(z) \qquad \text{subject to } Aw + Bz = c, . \tag{4}$$

where

$$f(w) = \mathbf{E}_{\xi \sim D} f(\xi, w).$$

In this case, if $f(\cdot)$ is smooth, then we may consider accelerated version of stochastic linearized ADMM, as in Algorithm 6. We may take a fixed β and set a constant learning rate η.

Algorithm 6: Stochastic Accelerated Linearized ADMM

Input: $\phi(\cdot)$, A, B, c, $\{\eta_t\}$, β, ρ, α_0, w_0, z_0

Output: w_T, z_T, α_T

Let $\bar{w}_0 = x_0$

Let $\bar{z}_0 = z_0$

for $t = 1, 2, \ldots, T$ **do**

  Let $\tilde{z}_t = \bar{z}_{t-1} - \eta_t B^\top [\alpha_{t-1} + \rho(A\bar{w}_{t-1} + Bz_{t-1} - c)]$

  Let $z_t = \arg\min_z [0.5\|z - \tilde{z}_t\|_2^2 + \eta_t g(z)]$

  Let $w_t = \bar{w}_{t-1} - \eta_t \nabla f_B(\bar{w}_{t-1}) - \eta_t A^\top [\alpha_{t-1} + \rho(A\bar{w}_{t-1} + Bz_t - c)]$

  Let $\alpha_t = \alpha_{t-1} + \rho(1-\beta)[Aw_t + Bz_t - c]$

  Let $\bar{z}_t = z_t + \beta(z_t - z_{t-1})$

  Let $\bar{w}_t = w_t + \beta(w_t - w_{t-1})$

Return: w_T, z_T, α_T

6 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 different versions of stochastic algorithms. For simplicity we use a fixed learning rate η.

We note that with different batch size, it is beneficial to increase learning rate proportionally with batch size. This is consistent with the theoretical analysis. This is true both with and without acceleration. The HB and Nesterov's method behave similarly.

Acceleration methods have a larger effective learning rate. Moreover, since we use a constant learning rate, oscillation eventually occur when the learning rate is too large. In such case, it is necessary to drop the learning rate. There is a more significant advantage when the objective is relatively nonsmooth.

Stochastic methods converge faster than deterministic proximal GD algorithms, but eventually stochastic methods converge slower when learning rate becomes too small.

Accelerated RDA is trickier to tune, and we include a particular implementation for comparison. The stochastic accelerated linearized ADMM works well, and easier to tune.

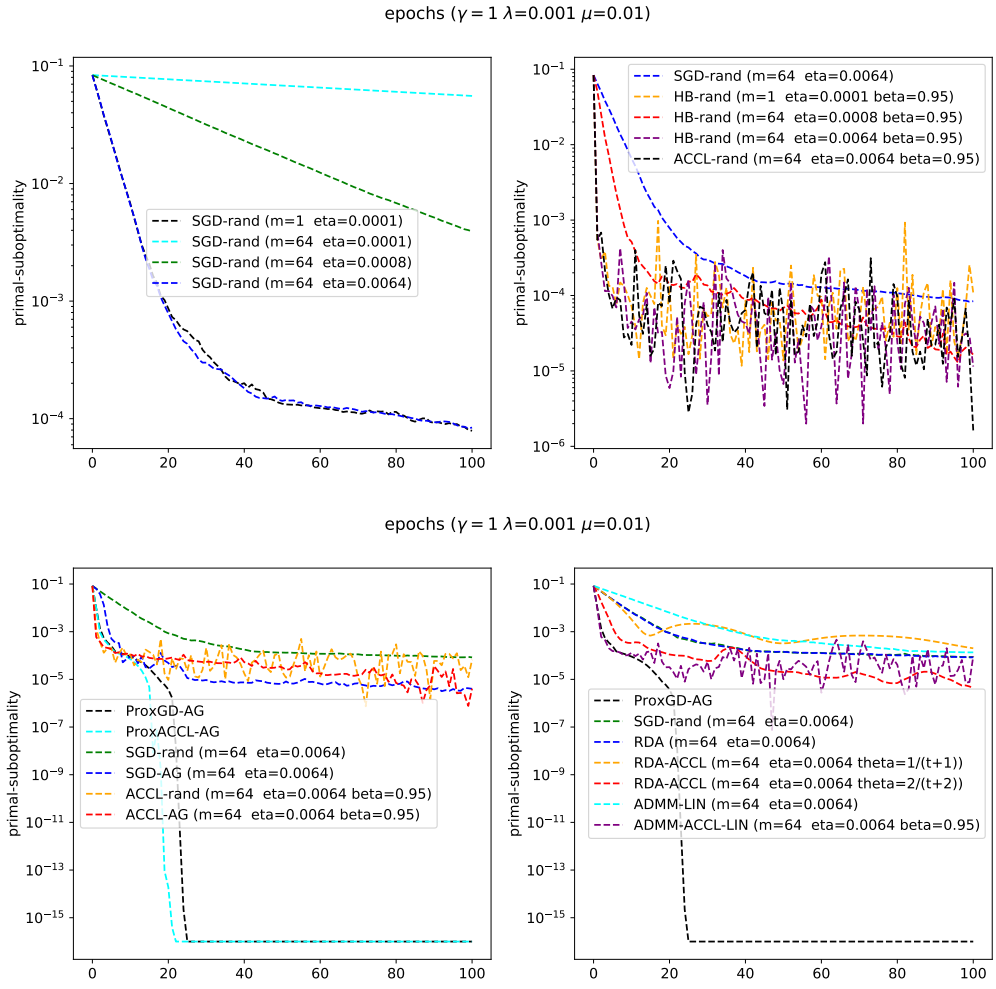

Figure 1: Comparisons of different stochastic algorithms (smooth and strongly convex)

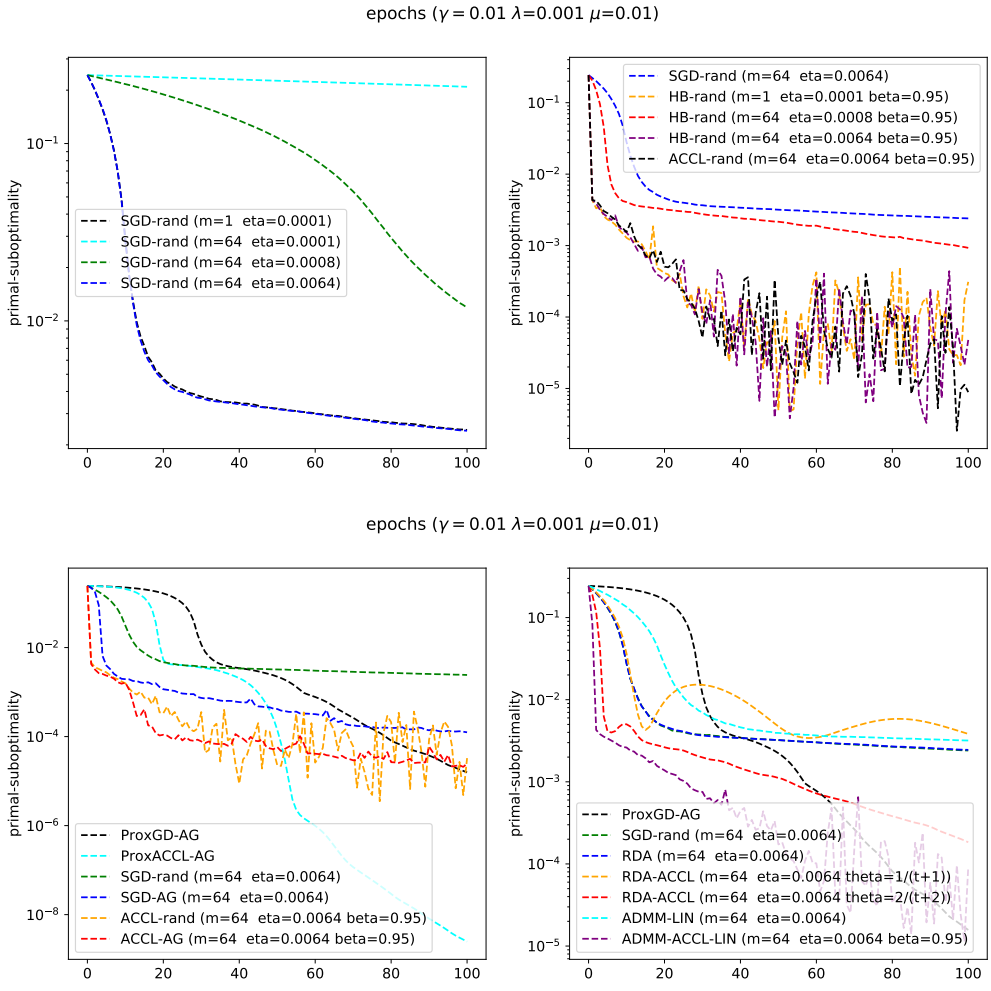

Figure 2: Comparisons of different stochastic algorithms (near non-smooth and strongly convex)

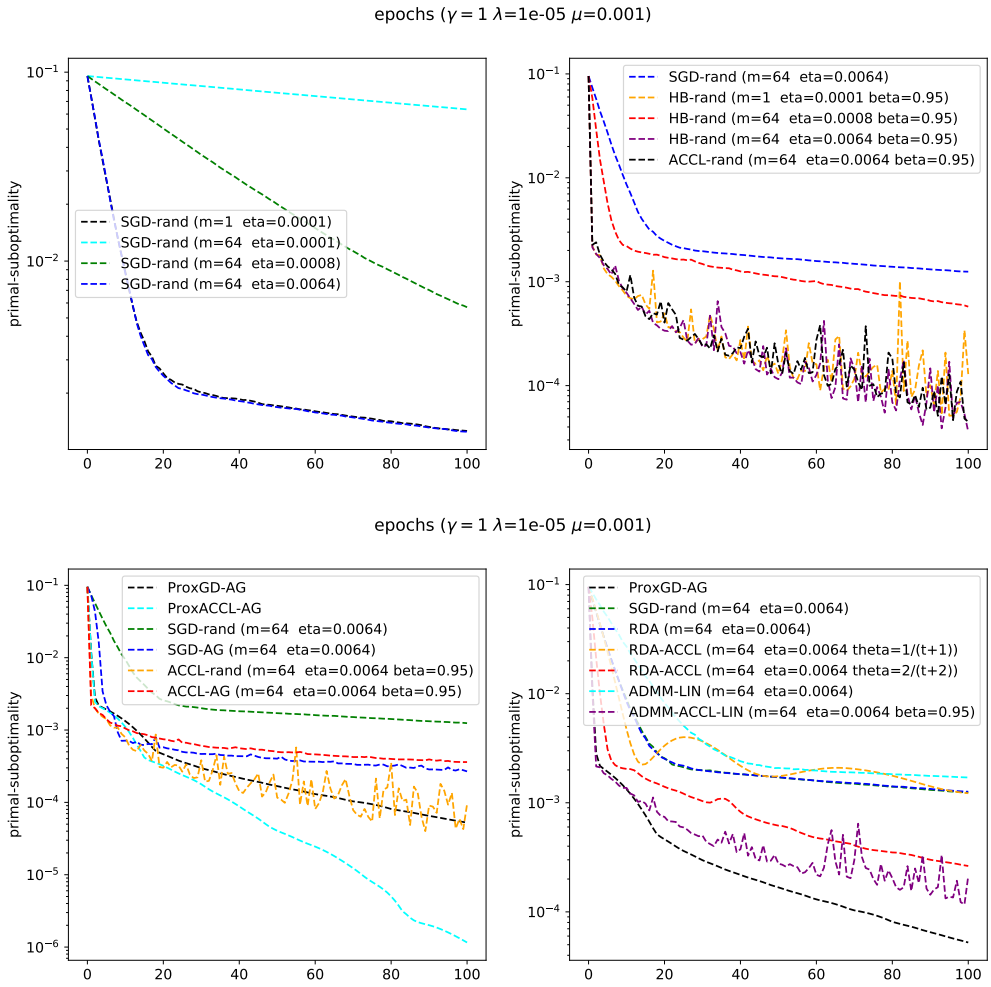

Figure 3: Comparisons of different stochastic algorithms (near non-smooth and near non-strongly convex)

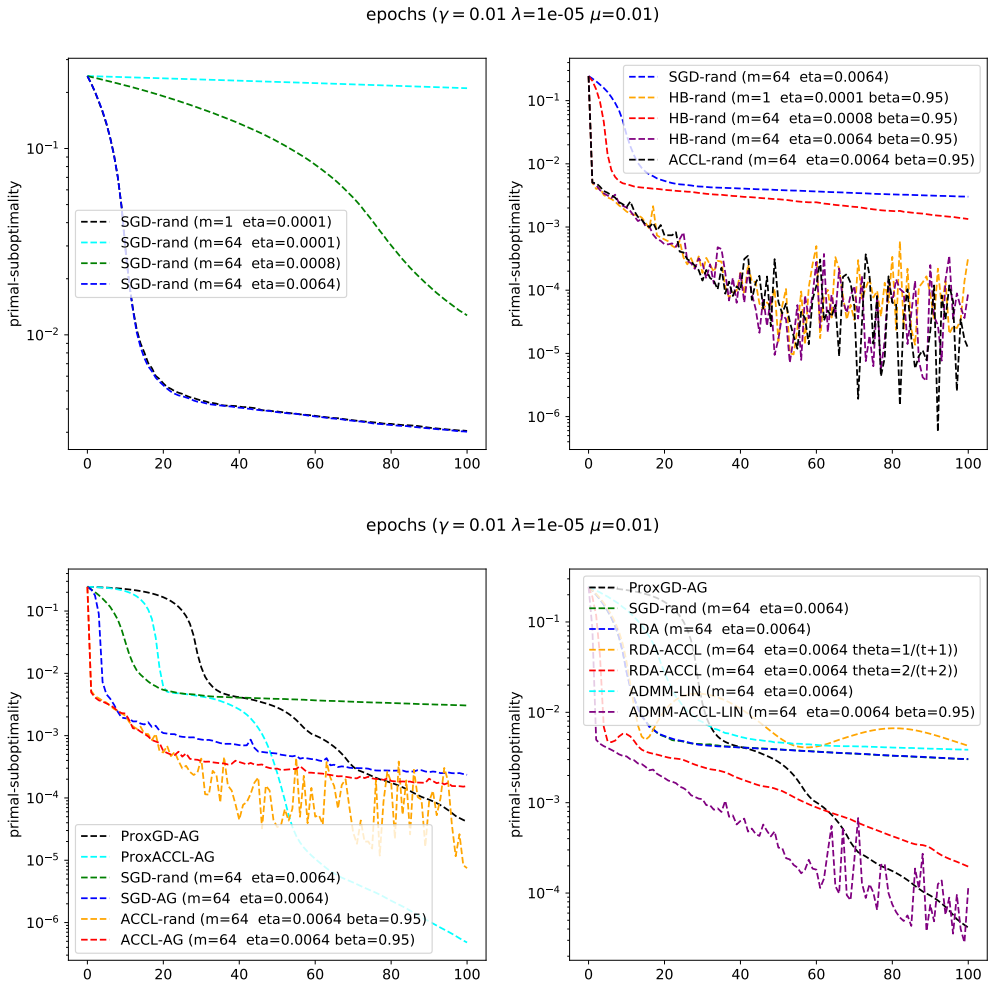

Figure 4: Comparisons of different stochastic algorithms (near non-smooth and near non-strongly convex)