Efficient Averaged Stochastic Gradient Descent for Large-Scale Learning: Theory and Optimization

Chris Junchi Li[◇]

Department of Electrical Engineering and Computer Sciences[◇]
University of California, Berkeley

October 4, 2024

Abstract

We propose a new analysis of Averaged Stochastic Gradient Descent (ASGD) for large-scale learning problems. ASGD, despite its promising asymptotic properties, has seen limited adoption in practice due to the large amount of data typically required to reach its optimal performance. In this work, we derive new non-asymptotic convergence guarantees for ASGD under specific learning rate schedules, demonstrating that it can outperform standard Stochastic Gradient Descent (SGD) when the dataset is sufficiently large. Our analysis also provides practical guidelines for selecting learning rate parameters based on problem characteristics, such as the curvature of the expected loss. Experimental results on multiple benchmark datasets show that ASGD achieves better convergence compared to state-of-the-art SGD algorithms.

Keywords: Averaged Stochastic Gradient Descent, Large-Scale Machine Learning, Non-Asymptotic Convergence, Learning Rate Scheduling.

1 Introduction

In recent years, Stochastic Gradient Descent (SGD) has become the dominant method for solving large-scale optimization problems in machine learning, due to its simplicity and scalability. However, the performance of SGD can be suboptimal when the dataset size grows large, primarily because it requires careful tuning of learning rates and multiple passes over the data to converge to an optimal solution.

Averaged Stochastic Gradient Descent (ASGD), first proposed by Polyak and Juditsky [**?**], offers a solution to some of these issues. ASGD averages the iterates of SGD, which has been shown to improve its convergence properties in an asymptotic sense. Despite its theoretical advantages, ASGD has been underutilized in practice. This is likely due to the fact that, without a proper learning rate schedule, ASGD may require an impractically large amount of data to reach its full potential.

In this paper, we propose a detailed analysis of ASGD's non-asymptotic behavior and derive practical guidelines for its implementation. By selecting an appropriate learning rate schedule based on problem characteristics, we show that ASGD can achieve superior performance to standard SGD, especially when applied to large-scale datasets. Furthermore, our experimental results demonstrate the practical benefits of ASGD across a range of tasks, including regression and classification problems.

Backgrounds For prediction problems, we want to find a function $f_\theta(x)$ with parameter θ to predict the value of the outcome variable y given an observed vector x. Typically, the problem is

formulated as an optimization problem:

$$\theta_t^* = \arg\min_\theta \frac{1}{t}\sum_{i=1}^{t}(L(f_\theta(x_i), y_i) + R(\theta)) \tag{1}$$

where t is the number of data points, θ_t^* is the parameter that minimize the empirical cost, (x_i, y_i) are the i^{th} training example, $L(s, y)$ is a loss function which gives small value if s is a good prediction for y, and $R(\theta)$ is a regularization function for θ which typically gives small value for small θ. Some commonly used L are: $\max(0, 1 - ys)$ for support vector machine (SVM), $\frac{1}{2}(\max(0, 1 - ys))^2$ for L2 SVM, and $\frac{1}{2}(y - s)^2$ for linear regression. Some commonly used regularization functions are: L2 regularization $\frac{\lambda}{2}\|\theta\|^2$, and L1 regularization $\lambda\|\theta\|_1$.

For large scale machine learning problems, we need to deal with optimization problems with millions or even billions of training samples. The classical optimization techniques such as interior point methods or conjugate gradient descent have to go through all data points to just evaluate the objective once. Not to say that they need to go through the whole data set many times in order to find the best θ.

On the other hand, stochastic gradient descent (SGD) has been shown to have great promise for large scale learning [Zhang(2004), Hazan et al.(2006), Shalev-Shwartz et al.(2007), Bottou and Bousquet(2008), Shalev-Shwartz and Tewari(2009), Langford et al.(2009)]. Let $d = (x, y)$ be one data sample, $l(\theta, d) = L(f_\theta(x), y) + R(\theta)$ be the cost of θ for d, $g(\theta, \xi) = \frac{\partial l(\theta, d)}{\partial \theta}$ be the gradient function, and $D_t = (d_1, \cdots, d_t)$ be all the training samples at t^{th} step. The SGD method updates θ according to its stochastic gradient:

$$\theta_t = \theta_{t-1} - \gamma_t g(\theta_{t-1}, d_t) \tag{2}$$

where γ_t is learning rate at the t^{th} step. γ_t can be either a scalar or a matrix. Let the expected loss of θ over test data be $\mathcal{E}(\theta) = E_d(l(\theta, d))$, the optimal parameter be $\theta^* = \arg\min_\theta \mathcal{E}(\theta)$, and the Hessian be $H = \frac{\partial^2 \mathcal{E}(\theta)}{\partial\theta\partial\theta^T}\Big|_{\theta=\theta^*}$. Note that θ_t and θ_t^* are random variables depending on D_t. Hence both $\mathcal{E}(\theta_t)$ and $\mathcal{E}(\theta_t^*)$ are random variables depending on D_t. If γ_t is a scalar, the best asymptotic convergence for the expected excess loss $E_{D_t}(\mathcal{E}(\theta_t)) - \mathcal{E}(\theta^*)$ is $O(t^{-1})$, which is obtained by using $\gamma_t = \gamma_0(1 + \gamma_0\lambda_0 t)^{-1}$, where λ_0 is the smallest eigenvalue of H and γ_0 is some constant. The asymptotic convergence rate of SGD can be potentially benefit from using second order information [Bottou and Bousquet(2008), Schraudolph et al.(2007), Amari et al.(2000)]. The optimal asymptotic convergence rate is achieved by using matrix valued learning rate $\gamma_t = \frac{1}{t}H^{-1}$. If this optimal matrix step size is used, then asymptotically second order SGD is as good as explicitly optimizing the empirical loss. More precisely, this means that both $tE_{D_t}(\mathcal{E}(\theta_t) - \mathcal{E}(\theta^*))$ and $tE_{D_t}(\mathcal{E}(\theta_t^*) - \mathcal{E}(\theta^*))$ converge to a same positive constant.

Since H is unknown in advance, methods for adaptively estimating H is proposed [Bottou and LeCun(2005), Amari et al.(2000)]. However, for high dimensional data sets, maintaining a full matrix H is too computationally expensive. Hence various methods for approximating H have been proposed [LeCun et al.(1998), Schraudolph et al.(2007), Roux et al.(2008), Bordes et al.(2009)]. However, with the approximated H, the optimal convergence cannot be guaranteed. It is worth to point out that most of the existing analysis for second order SGD is asymptotic, namely, that they do not tell how much data is needed for the algorithm to reach their asymptotic region.

In order to accelerate the convergence speed of SGD, averaged stochastic gradient (ASGD) was proposed in [Polyak and Juditsky(1992)]. For ASGD, the running average $\overline{\theta}_t = \frac{1}{t}\sum_{j=1}^{t}\theta_j$ of the parameters obtained by SGD is used as the estimator for θ^*. [Polyak and Juditsky(1992)] showed a

very nice result that $\overline{\theta}_t$ converges to θ^* as good as full second order SGD, which means that if there are enough training samples, ASGD can obtain the parameter as good as the empirical optimal parameter θ_t^* in just one pass of data. And another advantage of ASGD is that, unlike second order SGD, ASGD is extremely easy to implement. [Zhang(2004), Nemirovski et al.(2009)] gave some nice non-asymptotic analysis for ASGD with a fixed learning rate. However, the convergence bounds obtained by [Zhang(2004), Nemirovski et al.(2009)] are far less appealing than that of [Polyak and Juditsky(1992)].

Despite its nice properties, ASGD receives little attention in recent research for online large scale learning. The reason for the lack of interest in ASGD might be that its potential good convergence has not been realized by researchers in real applications. Our analysis shows the cause of this may due to the fact the ASGD needs a prohibitively large amount of data to reach asymptotics if learning rate is chosen arbitrarily.

A typical choice for the learning rate γ_t is to make it decease as fast as $\Theta(t^{-c})$ for some constant c. In this paper, we assume a particular form of learning rate schedule which satisfies this condition,

$$\gamma_t = \gamma_0(1 + a\gamma_0 t)^{-c} \tag{3}$$

where γ_0, a and c are some constants. Based on this form of learning rate schedule, we provide non-asymptotic analysis of ASGD. Our analysis shows that γ_0 and a should to be properly set according to the curvature of the expected cost function. c should be a problem independent constant. With our recipe for setting the learning rate, we show that ASGD outperforms SGD if the data size is large enough for SGD to reach its asymptotic region.

To demonstrate the effectiveness of ASGD with the proposed learning rate schedule, we apply ASGD for training linear classification and regression models. We compare ASGD with other prominent large scale SVM solvers on several benchmark tasks. Our experimental results show the clear advantage of ASGD.

In the rest of the paper, for matrices X and Y, $X \leq Y$ means $Y - X$ is positive semi-definite, $\|x\|_A$ is defined as $\sqrt{x^T A x}$. We will assume $\gamma_t = \gamma_0(1 + a\gamma_0 t)^{-c}$ for some constant $\gamma_0 > 0$, $a > 0$ and $0 \leq c \leq 1$ in all the theorems and lemmas. Through out this paper we denote $\Delta_t = \theta_t - \theta^*$ and $\overline{\Delta}_t = \overline{\theta}_t - \theta^*$. To help the reader focus on the main idea, we put most proofs to the Appendix.

The paper is organized as follows: Section 2 establish some results on stochastic linear equation; Section 3 extends the result to ASGD for quadratic loss functions; Section 4 works on general non-quadratic loss functions; Section 5 discusses some implementation issues; Section 6 shows experimental results; Section 7 concludes the paper; and Appendix includes all the proofs.

2 Stochastic Linear Equation

To motivate the problem, we first take a close look at the SGD update (2). Let $\overline{g}(\theta) = E(g(\theta, d))$ and the first order Taylor expansion of $\overline{g}(\theta)$ around θ^* be $A\theta - b$, where $A = \frac{\partial \overline{g}(\theta)}{\partial \theta}\Big|_{\theta=\theta^*}$ and $b = A\theta^* - \overline{g}(\theta^*) = A\theta^*$. Then $g(\theta_{t-1}, d)$ can be decomposed as:

$$\begin{aligned} g(\theta_{t-1}, d) &= (A\theta_{t-1} - b) + g(\theta^*, d) + (g(\theta_{t-1}, d) - g(\theta^*, d) - \overline{g}(\theta_{t-1})) + (\overline{g}(\theta_{t-1}) - A\theta_{t-1} + b) \\ &= (A\theta_{t-1} - b) + \xi_t^{(1)} + \xi_t^{(2)} + \xi_t^{(3)} \end{aligned}$$

where $\xi_t^{(1)} = g(\theta^*, d_t)$, $\xi_t^{(2)} = g(\theta_{t-1}, d_t) - g(\theta^*, d_t) - \overline{g}(\theta_{t-1})$ and $\xi_t^{(3)} = \overline{g}(\theta_{t-1}) - A\theta_{t-1} + b$. So the SGD update (2) can be re-written as

$$\theta_t = \theta_{t-1} - \gamma_t(A\theta_{t-1} - b + \xi_t^{(1)} + \xi_t^{(2)} + \xi_t^{(3)}) \tag{4}$$

It is easy to see that $\xi_t^{(1)}$ is martingale with respect to d_t, i.e., $E(\xi_t^{(1)}|d_1, \cdots, d_{t-1}) = 0$, and has identical distribution for different t. $\xi_t^{(2)}$ is also martingale with respect to d_t. However, as we will see in later section, its magnitude depends on $\theta_{t-1} - \theta^*$. If $g(\theta, d)$ is smooth, we have $\xi_t^{(2)} = O(\|\theta_{t-1} - \theta^*\|)$. For smooth $\overline{g}(\theta)$, we have $\xi_t^{(3)} = o(\|\theta_{t-1} - \theta^*\|)$. Both $\xi_t^{(2)}$ and $\xi_t^{(3)}$ are asymptotically negligible if suitable conditions are met. We also note that $\xi_t^{(3)} = 0$ for quadratic $l(\theta, \xi)$.

By the above analysis, we first consider the following simple stochastic approximation procedure which ignores $\xi_t^{(2)}$ and $\xi_t^{(3)}$:

$$\theta_t = \theta_{t-1} - \gamma_t(A\theta_{t-1} - b + \xi_t) \tag{5}$$

$$\overline{\theta}_t = \frac{1}{t}\sum_{i=1}^{t}\theta_i \tag{6}$$

where A is a positive definite matrix with the smallest eigenvalue λ_0 and the largest eigenvalue λ_1, ξ_t is martingale difference process, i.e., $E(\xi_t|\xi_1, \cdots, \xi_{t-1}) = 0$, the variance of ξ_t is $E(\xi_t\xi_t^T) = S$. We will see that this algorithm can be used to find the root θ^* of equation $A\theta = b$

Theorem 1. *If $\gamma_0\lambda_1 \leq 1$ and $(2c-1)a < \lambda_0$, then the estimator $\overline{\theta}_t$ in (6) satisfies:*

$$tE(\|\overline{\theta}_t - \theta^*\|_A^2) \quad \leq \quad \mathrm{tr}(A^{-1}S) + \frac{(2c_0 + c_0^2)(1 + a\gamma_0 t)^{c-1}}{c}\mathrm{tr}(A^{-1}S) + \frac{(1+c_0)^2}{\gamma_0^2 t}\|\theta_0 - \theta^*\|_{A^{-1}}^2$$

where

$$c_0 = \frac{ac(1 + ac\gamma_0)}{(\lambda_0 - \max(0, 2c-1)a)}$$

The immediate conclusion from Theorem 1 is the asymptotic convergence bound of $\overline{\theta}_t$.

Corollary 1. *$\overline{\theta}_t$ in (6) satisfies*

$$tE(\|\overline{\theta}_t - \theta^*\|_A^2) \leq \mathrm{tr}(A^{-1}S) + O(t^{-(1-c)})$$

The above bound is consistent with Theorem 1 in [Polyak and Juditsky(1992)] and is the best possible asymptotic convergence rate that can be achieved by any algorithms [Fabian(1973)]. However, we are more interested in the non-asymptotic behavior of $\overline{\theta}_t$.

Corollary 2. *If we choose $a = \lambda_0$, it takes $t = O((\lambda_0\gamma_0)^{-1})$ samples for $\overline{\theta}_t$ in (6) to reach the asymptotic region. And at this point, $\overline{\theta}_t$ begins to become better than θ_t.*

Proof. Let $t = \frac{K}{\lambda_0\gamma_0}$, we have

$$E(\|\overline{\Delta}_t\|_A^2) \leq \frac{(1+c_0)^2}{K^2}\|\Delta_0\|_A^2 + \frac{\lambda_0\gamma_0}{K}\left(1 + \frac{(2c_0 + c_0^2)(1+K)^{c-1}}{c}\right)\mathrm{tr}(A^{-1}S) \tag{7}$$

On the other hand, the best possible convergence for θ_t is obtained with $a = \lambda_0$ and $c = 1$:

$$E\left(\|\Delta_t\|_A^2\right) \leq \frac{\|\Delta_0\|_A^2}{(1+K)^2} + \frac{\gamma_0\mathrm{tr}(S)}{1+K} \tag{8}$$

We omit the proof of (8), which is similar to that of Theorem 1. A related (but not exactly same) result can be found in section 2.1 of [Nemirovski et al.(2009)]. From (7) and (8) we can see that both θ_t and $\overline{\theta}_t$ need $t = O((\lambda_0\gamma_0)^{-1})$ to reach their asymptotic region. However, at this point, $\overline{\theta}_t$ begins to become better than θ_t because $\lambda_0\mathrm{tr}(A^{-1}S) \leq \mathrm{tr}(S)$. □

Corollary 3. *It takes* $t = \Omega\left(\left(\frac{a}{\lambda_0}\right)^{\frac{c}{1-c}}(\lambda_0\gamma_0)^{-1}\right)$ *samples for* $\overline{\theta}_t$ *in (6) to reach the asymptotic region.*

Proof. In order for $\overline{\theta}_t$ to reach its asymptotic region, we need at least the second term of the right hand side of the bound in Theorem 1 to be less than $\mathrm{tr}(A^{-1}S)$, which is to say

$$2\frac{c_0(1+a\gamma_0 t)^{c-1}}{c} \leq 1$$

Hence

$$t \geq \frac{1}{a\gamma_0}\left(\frac{2c_0}{c}\right)^{\frac{1}{1-c}} = \left(\frac{2a}{\lambda_0}\right)^{\frac{c}{1-c}}(\lambda_0\gamma_0)^{-1}$$

□

By Corollary 3, we should limit a in order to have fast convergence. For the linear problem (5), we should always use $a = 0$. If we use some arbitrary value such as 1 for a, although $\overline{\theta}_t$ still has asymptotic optimal convergence according to [Polyak and Juditsky(1992)], but it needs much more samples to reach the asymptotic region in situations where λ_0 is very small. For the general SGD update (4), we need to trade-off against the convergence of $\xi^{(2)}$ and $\xi^{(3)}$. Hence a should not be 0. In general, a should be a constant factor times of λ_0.

3 Regression Problem

In this section, we will analyze the convergence for regression problems. As we noted in section 2, the SGD update can be decomposed as (4), where $\xi_t^{(3)} = 0$ for quadratic loss of linear regression. As in the proof of Theorem 1, $\overline{\Delta}_t$ can be written as:

$$\overline{\Delta}_t = \frac{1}{\gamma_0 t}\overline{X}_0^t\Delta_0 + \frac{1}{t}\sum_{j=1}^{t}\overline{X}_j^t\xi_j^{(1)} + \frac{1}{t}\sum_{j=1}^{t}\overline{X}_j^t\xi_j^{(2)} = I^{(0)} + I^{(1)} + I^{(2)}$$

We already have a bound for $\|I^{(0)}\|_A$ and $\|I^{(1)}\|_A$ in Theorem 1. Now we work on $I^{(2)}$. We will make two assumptions:

$$E\left(\|\xi_j^{(2)}\|_{A^{-1}}^2 \middle| \theta_{j-1}\right) \leq c_1\|\Delta_{j-1}\|_A^2 \tag{9}$$

$$\sum_{i=j}^{t} E\left(\|\Delta_t\|_A^2 \middle| \theta_{j-1}\right) \leq c_2\|\Delta_{j-1}\|_A^2 + c_3\sum_{i=j}^{t}\gamma_t \tag{10}$$

(9) is related to the continuity of $g(\theta, d)$ and the distribution of y. (10) is related to the convergence of standard SGD. A bound similar to (10) can be found in section 3.1 of [Hazan et al.(2006)]. Using these assumptions, we can bound $E\|I^{(2)}\|_A^2$:

Lemma 1. *With Assumption (9) (10) , we have*

$$tE\|I^{(2)}\|_A^2 \leq (1+c_0)^2 c_1\left(\frac{1+c_2}{t}\|\Delta_0\|_A^2 + \frac{c_3\gamma_0}{1-c}(1+a\gamma_0 t)^{-c}\right) \tag{11}$$

With the above lemma, we can obtain the following asymptotic convergence result:

Corollary 4. *For quadratic loss, with assumption (9) (10), $\overline{\theta}_t$ satisfies*

$$tE\|\overline{\theta}_t - \theta^*\|_A^2 \leq \mathrm{tr}(A^{-1}S) + O\left(t^{-c/2}\right) + O\left(t^{-(1-c)}\right)$$

Proof. Note that

$$(E\|\overline{\Delta}_t\|_A^2)^{1/2} \leq (E\|I^{(0)}\|_A^2)^{1/2} + (E\|I^{(1)}\|_A^2)^{1/2} + (E\|I^{(2)}\|_A^2)^{1/2}$$

The corollary follows by applying (16), (17) and Lemma 1. □

The best convergence rate is obtained when $c = 2/3$. Now we take a close look at the constant factor c_1 in assumption (9) to have a better understanding of the non-asymptotic behavior of $tE\|I^{(2)}\|_A^2$.

Lemma 2. *For ridge regression $l(\theta, d) = \frac{1}{2}(\theta^T x - y)^2$, if $\|x\| \leq M$, then*

$$E\left(\|\xi_j^{(2)}\|_{A^{-1}}^2 \middle| \theta_{j-1}\right) \leq \frac{M}{\lambda_0}\|\Delta_{j-1}\|_A^2$$

Assuming $\|x\| = M$, Lemma 6 in the Appendix shows that $\|\Delta_t\|^2$ will diverge if learning rate is greater than $\frac{2}{M}$. So $\gamma_0 \leq \frac{2}{M}$ and $c_1 \leq \frac{M}{\lambda_0}$. Plugging these bounds for c_1 and γ_0 into Lemma 1, we have the following for $t = \frac{K}{\lambda_0\gamma_0}$,

$$E\|I^{(2)}\|_A^2 \leq 2(1+c_0)^2\left(\frac{(1+c_2)\lambda_0\gamma_0\|\Delta_0\|_A^2}{K^2} + \frac{c_3\gamma_0}{(1-c)K(1+K)^c}\right)$$

Note that the best possible SGD error bound is $\frac{\|\Delta_0\|_A^2}{(1+K)^2} + \frac{c_3\gamma_0}{1+K}$ with $a = \lambda_0$ and $c = 1$. We see that $E\|I^{(2)}\|_A^2$ is negligible compared to the error of SGD if $t > O((\lambda_0\gamma_0)^{-1})$. Together with the analysis in Section 2, we conclude that ASGD begins to outperform SGD after $t > O((\lambda_0\gamma_0)^{-1})$. The conclusion we draw in this section applies not only to the case of y with constant norm. Similar conclusion can be drawn if y is normally distributed or if each dimension of y is independently distributed, and/or if L2 regularization is used.

Based on above analysis, for linear regression problems, we propose to use the following values for (3) to calculate the learning rate: $\gamma_0 = 1/M$, $a = \lambda_0$, $c = 2/3$. We will see that in the next section for general non-quadratic loss, optimal c is different since we need to further consider the convergence of $\xi_t^{(3)}$.

4 Non-quadratic loss

For non-quadratic loss, we need to analyze the contribution of $\xi^{(3)}$ to the error. We need the following two additional assumptions:

$$E\left(\|\xi_j^{(3)}\|_{A^{-1}} \middle| \theta_{j-1}\right) \leq c_4\|\theta_{j-1} - \theta^*\|_A^2 \tag{12}$$

$$\sum_{i=1}^{t} E(\|\Delta_t\|_A^4) \leq c_5\|\Delta_0\|_A^4 + c_6\sum_{i=1}^{t}\gamma_t \tag{13}$$

Similar to (9), (12) is related to the continuity of $g(\theta, d)$ and the distribution of x and y. Similar to (10), (13) is related to the convergence of standard SGD. We note that the asymptotic normality of θ_t [Fabian(1968)] suggests that assumption (13) is reasonable.

Lemma 3. *With Assumption (9) (10) (12) and (13) , we have*

$$tE\|I^{(3)}\|_A^2 \leq \frac{(1+c_0)^2 c_4^2}{t}\left((1+2c_2)c_5\|\Delta_0\|_A^4 + (2c_2c_3\|\Delta_0\|_A^2 + (1+2c_2)c_6)\gamma_1^t + c_3^2(\gamma_1^t)^2\right)$$

where $\gamma_1^t = \sum_{s=1}^t \gamma_s$.

Corollary 5. *For non-quadratic loss, with assumption (9) (10) (12) and (13), if* $c > \frac{1}{2}$*, then* $\overline{\theta}_t$ *satisfies*

$$tE\|\overline{\theta}_t - \theta^*\|_A^2 \leq \mathrm{tr}(A^{-1}S) + O\left(t^{-(c-1/2)}\right) + O\left(t^{-(1-c)}\right)$$

Proof. Note that

$$(E\|\overline{\Delta}_t\|_A^2)^{1/2} \leq (E\|I^{(0)}\|_A^2)^{1/2} + (E\|I^{(1)}\|_A^2)^{1/2} + (E\|I^{(2)}\|_A^2)^{1/2} + (E\|I^{(3)}\|_A^2)^{1/2}$$

The corollary follows by applying (16), (17), Lemma 1 and Lemma 3. □

The best convergence rate is obtained when $c = 3/4$, which is different from that for quadratic loss.

5 Implementation

In this section, we discuss how we implement ASGD for linear models $f_\theta(x) = \theta^T x$ with L2 regularization. The running average can be recursively updated by $\overline{\theta}_t = (1-\frac{1}{t})\overline{\theta}_{t-1} + \frac{1}{t}\theta_t$, which is very easy to implement. However, for sparse data sets, this can be very costly compared to SGD since θ_t is typically a dense vector. Consider the following average procedure:

$$\theta_t = (1-\lambda\gamma_t)\theta_{t-1} - \gamma_t g_t \quad , \quad \overline{\theta}_t = (1-\eta_t)\overline{\theta}_{t-1} + \eta_t\theta_t$$

where λ is the L2 regularization coefficient, $g_t = \frac{\partial L(\theta_{t-1}^T x_t, y_t)}{\partial \theta_{t_1}} = L_s(\theta_{t-1}^T x_t, y_t)x_t$, and η_t is the rate of averaging. Hence g_t is sparse when x_t is sparse. We want to take the advantage of the sparsity of x_t for updating θ_t and $\overline{\theta}_t$. Let

$$\alpha_t = \frac{1}{\prod_{i=1}^t(1-\lambda\gamma_i)} \quad , \quad \beta_t = \frac{1}{\prod_{i=1}^t(1-\eta_i)} \quad , \quad u_t = \alpha_t\theta_t \quad , \quad \overline{u}_t = \beta_t\overline{\theta}_t$$

After some manipulation, we get the following:

$$\begin{aligned}
u_t &= u_{t-1} - \alpha_t\gamma_t g_t \\
\overline{u}_t &= \overline{u}_{t-1} + \beta_t\eta_t\theta_t = \overline{u}_0 + \sum_{i=1}^t \frac{\beta_i\eta_i}{\alpha_i}u_i \\
&= \overline{u}_0 + \sum_{i=1}^t \frac{\beta_i\eta_i}{\alpha_i}\left(u_t + \sum_{j=i+1}^t \alpha_j\gamma_j g_j\right) \\
&= \overline{u}_0 + u_t\sum_{i=1}^t \frac{\eta_i\beta_i}{\alpha_i} + \sum_{j=1}^t\left(\sum_{i=1}^{j-1}\frac{\eta_i\beta_i}{\alpha_i}\right)\alpha_j\gamma_j g_j
\end{aligned}$$

Algorithm 1 Sparse ASGD

$\alpha_0 = 1 \quad , \quad \beta_0 = 1 \quad , \quad \tau_0 = 0 \quad , \quad u_0 = \overline{\theta}_0 \quad , \quad \widehat{u}_0 = \overline{\theta}_0$
while $t \leq T$ **do**
  $g_t = L_s(\frac{1}{\alpha_{t-1}} u_{t-1}^T x_t, y_t) x_t$
  $\alpha_t = \frac{\alpha_{t-1}}{1-\lambda\gamma_t}$
  $\beta_t = \frac{\beta_{t-1}}{1-\eta_t}$
  $u_t = u_{t-1} - \alpha_t \gamma_t g_t$
  $\widehat{u}_t = \widehat{u}_{t-1} + \tau_{t-1}\alpha_t\gamma_t g_t$
  $\tau_t = \tau_{t-1} + \frac{\eta_t \beta_t}{\alpha_t}$
end while

Now define $\tau_t = \sum_{i=1}^t \frac{\eta_i \beta_i}{\alpha_i}$ and $\widehat{u}_t = \widehat{u}_{t-1} + \tau_{t-1}\alpha_t\gamma_t g_t$ with $\widehat{u}_0 = \overline{u}_0$, we get

$$\overline{u}_t \quad = \quad \overline{u}_0 + \tau_t u_t + \sum_{j=1}^{t} \tau_{t-1}\alpha_j\gamma_j g_j = \tau_t u_t + \widehat{u}_t$$

Hence we obtain the following efficient algorithm for updating $\overline{\theta}_t$:

At any step of the algorithm, $\overline{\theta}_t$ can be obtained by $\overline{\theta}_t = \frac{\overline{u}_t}{\beta_t} = \frac{\tau_t u_t + \widehat{u}_t}{\beta_t}$. Note that in Algorithm 1, none of the operations involves two dense vectors. Thus the number of operations per sample is $O(Z)$, where Z is the number of non-zero elements in x.

From Theorem 1 we can see that if $\|\Delta_0\|^2_{A^{-1}}$ is large compared to $\mathrm{tr}(A^{-1}S)$, then the error is dominated by $I^{(0)}$ at the beginning. This can happen if noise is small compared to $\|\Delta_0\|$. It is possible to further improve the performance of ASGD by discarding θ_t from averaging during the initial period of training. We want to find a point t_0 whereafter averaging becomes beneficial. For this, we maintain an exponential moving average $\widehat{\theta}_t = 0.99\widehat{\theta}_{t-1} + 0.01\theta_t$ and compare the moving average of the empirical loss of $\widehat{\theta}_t$ and θ_t. Once $\widehat{\theta}_t$ is better than θ_t, we begin the ASGD procedure.

6 Experiments

In this section, we provide 3 sets of experiments. The first experiment illustrate the importance of learning rate scheduling for ASGD. The second experiment illustrates the asymptotic optimal convergence of ASGD. In the third set of experiments, we apply ASGD on many public benchmark data sets and compare it with several state of the art algorithms.

6.1 Effect of learning rate scheduling

Our first experiment is used to show how different learning rate schedule affects the convergence of ASGD using a synthetic problem. The exemplar optimization problem is $\min_\theta E_x((\theta - x)^T A(\theta - x))$, where A is a symmetric 100x100 matrix with eigenvalues $[1, 1, 1, 0.02 \cdots 0.02]$ and x follows normal distribution with zero mean and unit covariance. It can be shown that the optimal θ is $\theta^* = 0$. Figure 1 shows the excess risk $\mathcal{E}(\theta_t) - \mathcal{E}(\theta^*)$ of the solution vs. number of training samples t. We note that in this particular example the excess risk is simply $\theta_t^T A \theta_t$. For the good example of ASGD (ASGD in the figure), we use our proposed learning rate schedule $\gamma_t = (1 + 0.02t)^{-2/3}$ according to

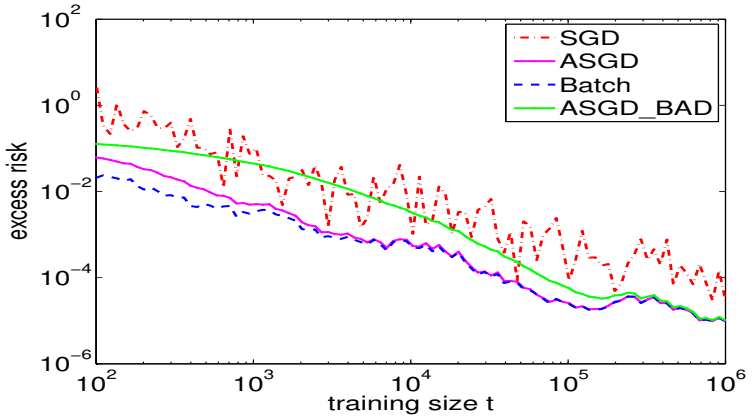

Figure 1. ASGD with proposed learning rate schedule (ASGD) and an arbitrarily chosen learning rate schedule (ASGD_BAD).

Section 3. For a bad example of ASGD (ASGD_BAD in the figure), we use $\gamma_t = (1+t)^{-1/2}$, which looks simple and also has optimal asymptotic convergence according to Corollary 1. Figure 1 also shows the performance of standard SGD using learning rate schedule $\gamma_t = (1+0.02t)^{-1}$ and batch method $\theta_t = \frac{1}{t}\sum_{j=1}^{t} x_t$. We see that both ASGD and ASGD_BAD eventually outperforms SGD and come close to the batch method. However, it takes only a few thousands example for ASGD to get to the asymptotic region, while it takes hundreds of thousands of examples for ASGD_BAD. This huge difference illustrates the significant role of learning rate scheduling for ASGD.

6.2 Asymptotic optimal convergence

Our second experiment is used to show the asymptotic optimality of ASGD for linear regression. For this purpose, we generate synthetic regression problem $y = x^T\theta^* + \epsilon$, where x is $N = 100$ dimensional vector following Gaussian distribution with zero mean and covariance A, the eigenvalues of A are evenly spread from 0.01 to 1, θ^* is a vector with all dimension equal to 1, ϵ follows Gaussian distribution with zero mean and unit variance. We compare ASGD with SGD and batch method. We use $\gamma_0 = 1/\mathrm{tr}(A)$ for both ASGD and SGD. For batch method, we simply calculate θ_t as $\theta_t = (\sum_{i=1}^{t} x_i x_i^T)^{-1} \sum_{i=1}^{t} x_i y_i$. Figure 2 shows the excess risk $\mathcal{E}(\theta_t) - \mathcal{E}(\theta^*)$ of the solution vs. number of training samples t. As the figure shows, after about 10^4 examples, the accuracy of ASGD starts to be close to batch solution while the solution of SGD remains more than 10 times worse than ASGD. Note that although ASGD and batch solution has similar accuracy, ASGD is considerably fast than batch method since ASGD only need $O(N)$ computation per sample while batch method need $O(N^2)$ computation per sample.

6.3 Experiments on benchmark data sets

In the third set of experiments, we compare ASGD with several other algorithms for training large scale linear models: online limited-memory BFGS (oLBFGS) of [Schraudolph et al.(2007)], stochastic gradient descent (SGD2) of [Bottou(2007)], dual coordinate descent (LIBLINEAR) of [Fan et al.(2008)], Pegasos of [Shalev-Shwartz et al.(2007)] and SGDQN of [Bordes et al.(2009)]. We performed extensive evaluation of ASGD on many data sets. Due to space limit, we only show detailed results on four tasks in this paper. COVTYPE is the detection of class 2 among 7 forest cover types (Blackard et al). All dimensions are normalized between 0 and 1. DELTA is

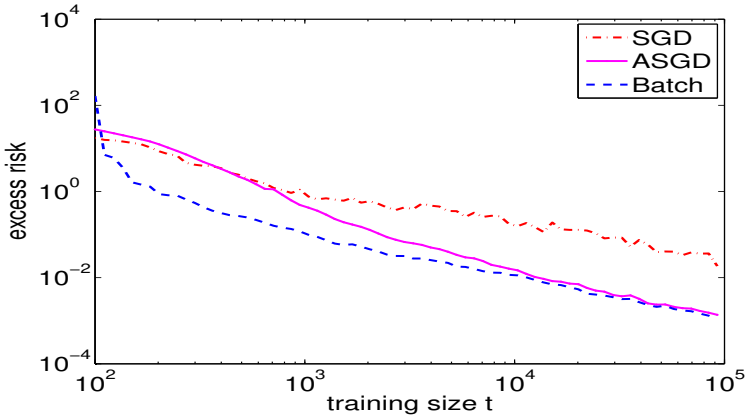

Figure 2: Compare ASGD with batch method.

a synthetic data set from the PASCAL Large Scale Challenge [Sonnenburg et al.(2008)]. We use the default data preprocessing provided by the challenge organizers. RCV1 is the classification of documents belonging to class CCAT in RCV1 text data set [Lewis et al.(2004)]. We use the same preprocessing as provided in [Bottou(2007)]. MNIST9 is the classification of digit 9 against all other digits in MNIST digit image data set [LeCun et al.(1998)]. For this task, we generate our own image feature vectors for recognition. The experiments for these four tasks use squared hinge loss $L(s, y) = \frac{1}{2}(\max(0, 1 - ys))^2$ with $L2$ regularization $R(\theta) = \frac{\lambda}{2}\|\theta\|_2^2$. Since λ_0 is unknown, we use the regularization coefficient λ as λ_0, which is a lower bound for true λ_0. Table 1 summarizes the data sets, where M is the $\max \|x\|^2$ calculated from 1000 samples, t_0 is the point where average begins (See Section 5). Figure 3 shows the test error rate (left), elapsed time (middle) and test cost (right) at different points within first two passes of training data.

We also include more experimental results on data sets from Pascal Large Scale Challenge. However, to save space, we only show figures for test error rate. All experiments use the default data preprocessing provided by the challenge organizers. Table 2 summarize the data sets. Figure 4 and Figure 5 shows result for L2 SVM, logistic regression and SVM. LIBLINEAR is not included in the figures for logistic regression because the dual coordinate descent method used by LIBLINEAR cannot solve logistic regression. Although the theory of ASGD only applies to smooth cost functions, we also include the results of SVM to satisfy the possible curiosity of some readers.

As we can see from the figures, ASGD clearly outperforms all other 5 algorithms in terms accuracy in most of the data sets. In fact, for most of the data sets, ASGD reaches good performance with only one pass of data, while many other algorithms still perform poorly at that point. The only exception is the *beta* data set, where all methods performs equally bad because the two classes in this data set are not linearly separable. Moreover, the performance of the other 5 methods tend to be more volatile, while performance of ASGD is more robust due to average. In terms of time spent on one pass of data, ASGD is similar to the other methods except oLBFGS, which means that ASGD needs less time to reach similar test performance compared to the other methods. Another interesting point is that although the current theory of ASGD is based on the assumption that cost function is smooth, as shown in the figures, ASGD also works pretty well with non-smooth loss such as hinge loss.

Table 1: Data Set Summary

	description	type	dim	train size	test size	λ	M	t_0
covtype	forest cover type	sparse	54	500k	81k	10^{-6}	6.8	100
delta	synthetic data	dense	500	400k	50k	10^{-2}	3.8×10^3	100
rcv1	text data	sparse	47153	781k	23k	10^{-5}	1	781
mnist9	digit image features	dense	2304	50k	10k	10^{-3}	2.1×10^4	128

Table 2: Data Set Summary

	description	type	dim	train size	test size	λ	M
alpha	synthetic data	dense	500	400k	50k	10^{-5}	1
beta	synthetic data	dense	500	400k	50k	10^{-4}	1
gamma	synthetic data	dense	500	400k	50k	10^{-3}	2.5×10^3
epsilon	synthetic data	dense	2000	400k	50k	10^{-5}	1
zeta	synthetic data	dense	2000	400k	50k	10^{-5}	1
fd	character image	dense	900	1000k	470k	10^{-5}	1
ocr	character image	dense	1156	1000k	500k	10^{-5}	1
dna	DNA sequence	sparse	800	1000k	1000k	10^{-3}	200

7 Conclusion

Averaged Stochastic Gradient Descent (ASGD) is relatively easy to implement compared to more complex algorithms, and as demonstrated on both synthetic and real datasets, it performs better in large-scale learning problems when using our proposed learning rate schedule. While ASGD has traditionally been underutilized due to the large number of samples required to reach its asymptotic region, our finite-sample analysis shows that with a properly chosen learning rate, ASGD can achieve optimal performance much sooner. This makes ASGD a strong contender for large-scale linear models with convex loss functions.

In future work, it will be interesting to explore the application of ASGD to more complex models, such as conditional random fields (CRF) or models with multiple local optima, like neural networks, to further extend its effectiveness in diverse machine learning tasks.

References

[Amari et al.(2000)] Shun-ichi Amari, Hyeyoung Park, and Kenji Fukumizu. Adaptive method of realizing natural gradient learning for multilayer perceptrons. *Neural Computation*, 12:1399–1409, 2000.

[Bordes et al.(2009)] Antoine Bordes, Léon Bottou, and Patrick Gallinari. SGD-QN: Careful quasi-Newton stochastic gradient descent. *Journal of Machine Learning Research*, 10:1737–1754, 2009.

[Bottou(2007)] Léon Bottou. Stochastic gradient descent on toy problems. http://leon.bottou.org/projects/sgd, 2007.

[Bottou and Bousquet(2008)] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 161–168. MIT Press, Cambridge, MA, 2008.

[Bottou and LeCun(2005)] Léon Bottou and Yann LeCun. On-line learning for very large datasets. *Apllied Stochastic Models in Business and Industry*, 21(21):137–151, 2005.

[Fabian(1968)] Václav Fabian. On asymptotic normality in stochastic approximation. *The Annals of Mathematical Statistics*, 39(4):1327–1332, 1968.

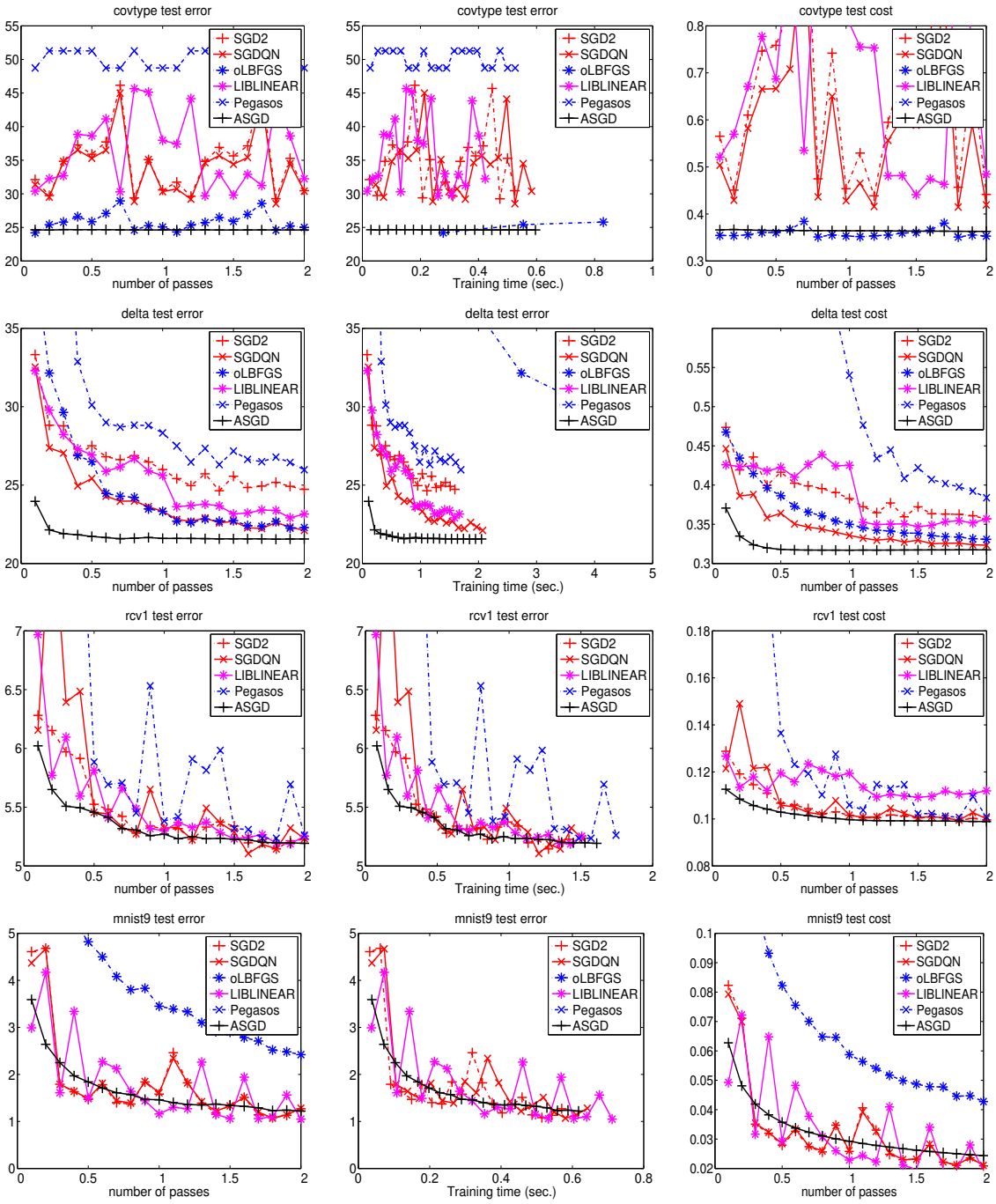

Figure 3. Left: Test error (%) vs. number of passes. Middle: Test error vs. training time. Right: Test cost vs. number of passes.

[Fabian(1973)] VÁclav Fabian. Asymptotically efficient stochastic approximation; the RM case. *The Annals of Statistics*, 1(3):486–495, 1973.

[Fan et al.(2008)] Rong-En Fan, Kai-Wei Change, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

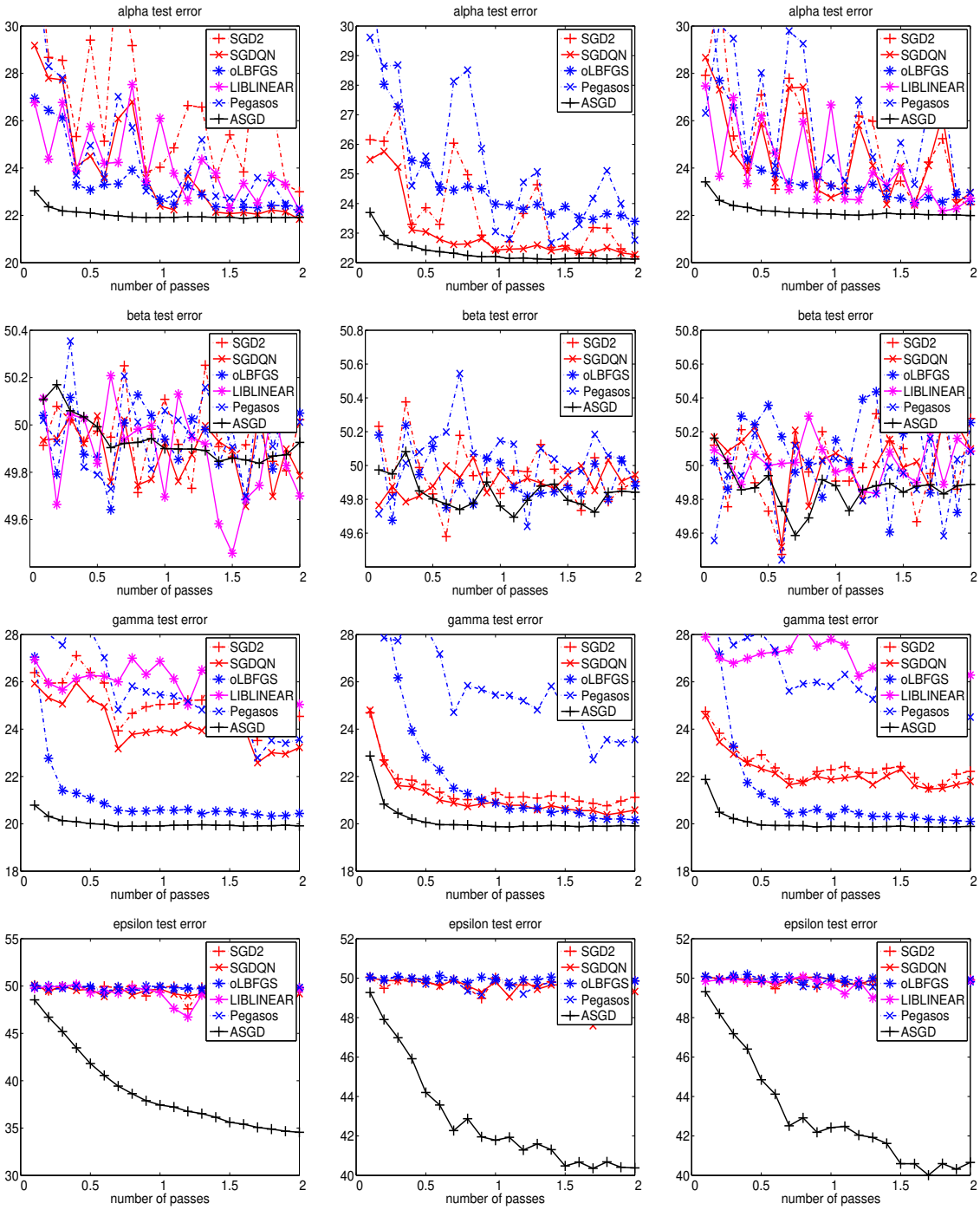

Figure 4. Test error (%) vs. number of passes. Left: L2SVM; Middle: logistic regression; Right: SVM.

[Hazan et al.(2006)] Elad Hazan, Adam Kalai, and Satyen Kale Amit Agarwal. Logarithmic regret algorithms for online convex optimization. In *Proceedings of the 19th Annual Conference on Learning Theory*, Pittsburgh, Pennsylvania, 2006.

[Langford et al.(2009)] John Langford, Lihong Li, and Tong Zhang. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10:777–801, 2009.

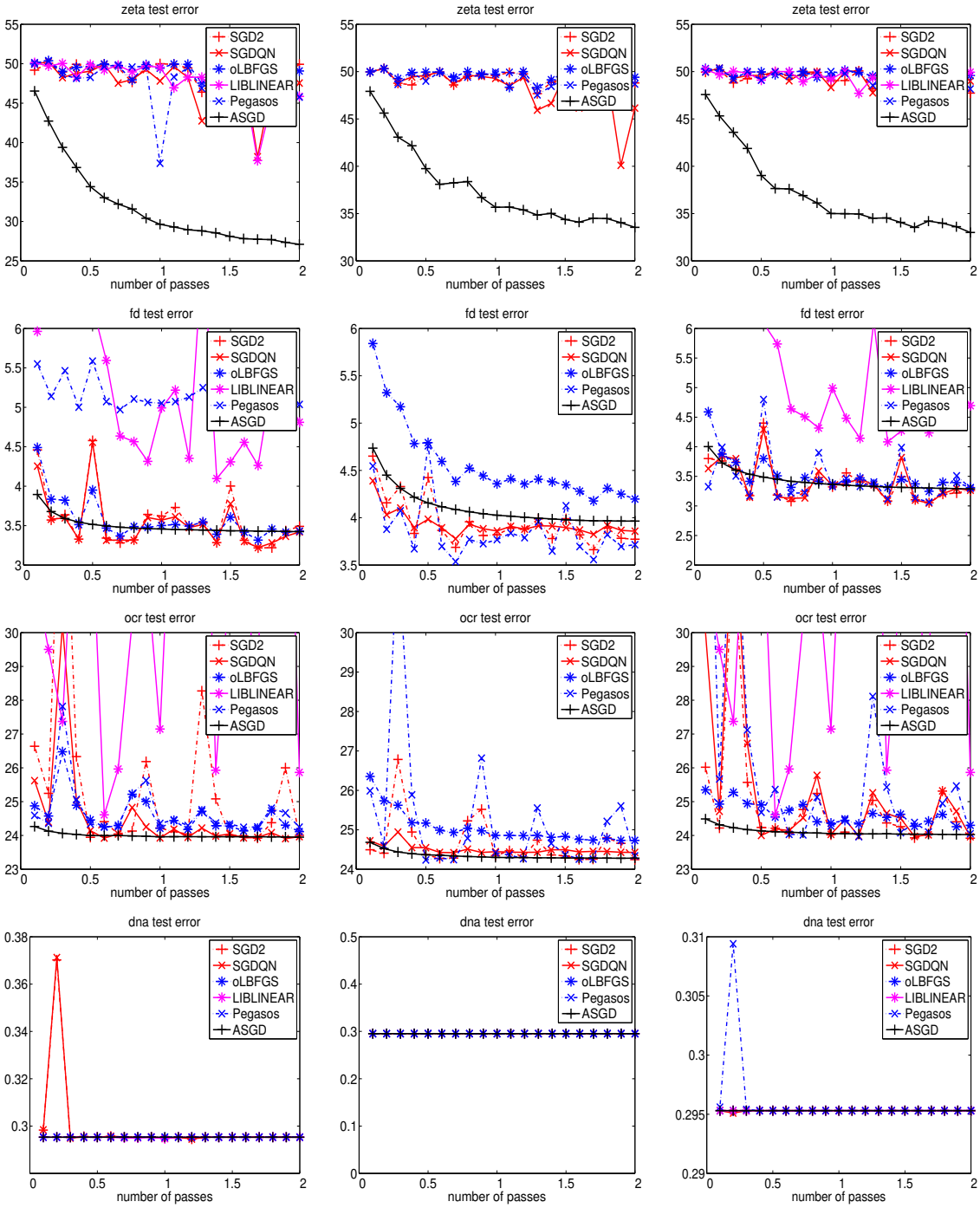

Figure 5. Test error (%) vs. number of passes. Left: L2SVM; Middle: logistic regression; Right: SVM.

[LeCun et al.(1998)] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Lewis et al.(2004)] David D. Lewis, Yiming Yang, Tony G. Rose, G., and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[Nemirovski et al.(2009)] Arkadi Nemirovski, Anatoli Juditski, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Control and Optimization*, 19(4):1574–1609, 2009.

[Polyak and Juditsky(1992)] Boris T. Polyak and Anatoli. B. Juditsky. Acceleration of stochastic approximation by averaging. *Automation and Remote Control*, 30(4):838–855, 1992.

[Roux et al.(2008)] Nicolas Le Roux, Pierre-Antoine Manzagol, and Yoshua Bengio. Topmoumoute online natural gradient algorithm. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 849–856. MIT Press, Cambridge, MA, 2008.

[Schraudolph et al.(2007)] Nicol N. Schraudolph, Jin Yu, and Simon Günter. A stochastic quasi-newton method for online convex optimization. In *Proceedings of the 9^{th} International Conference on Artificial Intelligence and Statistics (AISTAT)*, pages 433–440, 2007.

[Shalev-Shwartz and Tewari(2009)] Shai Shalev-Shwartz and Ambuj Tewari. Stochastic methods for ℓ_1 regularized loss minimization. In *Proceedings of the 26^{st} International Conference on Machine Learning (ICML)*, 2009.

[Shalev-Shwartz et al.(2007)] Shai Shalev-Shwartz, Yoram Shinger, and Nathan Srebro. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *Proceedings of the 24^{th} Fourth International Conference on Machine Learning (ICML)*, Corvallis, OR, 2007.

[Sonnenburg et al.(2008)] Soeren Sonnenburg, Vojtech Franc, Elad Yom-Tov, and Michele Sebag. Pascal large scale learning challenge. http://largescale.first.fraunhofer.de, 2008.

[Zhang(2004)] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the 21^{st} International Conference on Machine Learning (ICML)*, 2004.

A Proofs

Lemma 4. *Let* $\kappa = 1 - \max(0, 2c-1)\frac{a}{\lambda_0}$. *If* $\gamma_0\lambda_1 \le 1$, *then*

$$\left(\frac{1}{\gamma_{k+1}} - \frac{1}{\gamma_k}\right)\frac{1}{\gamma_{k+1}} \le \left(\frac{1}{\gamma_k} - \frac{1}{\gamma_{k-1}}\right)\frac{1}{\gamma_k}(1-\lambda_0\gamma_k)^{\kappa-1}$$

Proof. For $0 < c \le 0.5$, let $f(x) = (x^c - (x-1)^c)x^c$, where $x = k + \frac{1}{a\gamma_0}$. We only need to show $f'(x) \le 0$

$$\begin{aligned} f'(x) &= 2cx^{2c-1} - c(x-1)^{c-1}x^c - c(x-1)^c x^{c-1} \\ &= 2cx^{c-1}(x^c - (x-1)^c - \frac{1}{2}(x-1)^{c-1}) \\ &\le 2cx^{c-1}((x-1)^c + c(x-1)^{c-1} - (x-1)^c - \frac{1}{2}(x-1)^{c-1}) \\ &= c(2c-1)x^{c-1}(x-1)^{c-1} \le 0 \end{aligned}$$

where we used the fact $x^c \le (x-1)^c + c(x-1)^{c-1}$ for $0 \le c \le 1$.

For $c > 0.5$, let $f(x) = \log((x^c - (x-1)^c)x^c)$, where $x = k + \frac{1}{a\gamma_0}$. We only need to show

$$f(x+1) - f(x) + \frac{a(2c-1)}{\lambda_0}\log(1 - \lambda_0\gamma_0(a\gamma_0 x)^{-c}) \le 0$$

By mean value theorem, there exists some $y : x \le y \le x+1$ s.t. $f(x+1) - f(x) = f'(y)$. Hence

$$\begin{aligned} & f(x+1) - f(x) + \frac{a(2c-1)}{\lambda_0}\log((1 - \lambda_0\gamma_0(a\gamma_0 x)^{-c}) \\ \le\ & f'(y) - a(2c-1)\gamma_0(a\gamma_0 x)^{-c} \le f'(y) - (2c-1)(a\gamma_0)^{1-c}y^{-c} \\ =\ & \frac{2c(y^c - (y-1)^c - \frac{1}{2}(y-1)^{c-1})}{y(y^c - (y-1)^c)} - \frac{(2c-1)(a\gamma_0 y)^{1-c}}{y} \\ \le\ & \frac{2c(y^c - (y-1)^c - \frac{1}{2}(y-1)^{c-1})}{y(y^c - (y-1)^c)} - \frac{2c-1}{y} \\ =\ & \frac{y^c - (y-1)^c - c(y-1)^{c-1}}{y(y^c - (y-1)^c)} \le 0 \end{aligned}$$

□

The following is a key lemma which is used several times in this paper.

Lemma 5. *Let* X_j^t *and* $\overline{X}_j^t$ *be*

$$X_j^t = \prod_{i=j}^{t}(I - \gamma_i A) \quad , \quad X_j^t = I \text{ for } j > t \quad , \quad \overline{X}_j^t = \sum_{i=j}^{t}\gamma_j X_{j+1}^i$$

If $\gamma_0\lambda_1 \le 1$ *and* $(2c-1)a < \lambda_0$, *then we have the following bound for* $\overline{X}_j^t$.

$$(I - X_j^t)A^{-1} \le \overline{X}_j^t \le (1 + c_0(1 + a\gamma_0 j)^{c-1})A^{-1} \le (1+c_0)A^{-1}$$

where c_0 *is the same as in Theorem 1.*

Proof. It is easy to verify the following relation by induction on t,

$$\sum_{i=j}^{t} \gamma_i X_j^{i-1} = (I - X_j^t)A^{-1} \tag{14}$$

Now we calculate the difference between $\overline{X}_j^t$ and $\sum_{i=j}^t \gamma_i X_j^{i-1}$.

$$\begin{aligned}
\overline{X}_j^t - \sum_{i=j}^{t} \gamma_i X_j^{i-1} &= \sum_{i=j}^{t} (\gamma_j - \gamma_i) X_{j+1}^{i-1} = \sum_{i=j}^{t} \frac{\gamma_j - \gamma_i}{\gamma_i} \gamma_i X_{j+1}^{i-1} \\
&= \sum_{i=j}^{t} \sum_{k=j+1}^{i} \left(\frac{\gamma_j}{\gamma_k} - \frac{\gamma_j}{\gamma_{k-1}} \right) \gamma_i X_{j+1}^{i-1} = \sum_{k=j+1}^{t} \left(\frac{\gamma_j}{\gamma_k} - \frac{\gamma_j}{\gamma_{k-1}} \right) \sum_{i=k}^{t} \gamma_i X_{j+1}^{i-1} \\
&= \sum_{k=j+1}^{t} \left(\frac{\gamma_j}{\gamma_k} - \frac{\gamma_j}{\gamma_{k-1}} \right) \left(\sum_{i=j+1}^{t} \gamma_i X_{j+1}^{i-1} - \sum_{i=j+1}^{k-1} \gamma_i X_{j+1}^{i-1} \right) \\
&= \sum_{k=j+1}^{t} \left(\frac{\gamma_j}{\gamma_k} - \frac{\gamma_j}{\gamma_{k-1}} \right) A^{-1} (I - X_{j+1}^t - I + X_{j+1}^{k-1}) \\
&= -\left(\frac{\gamma_j}{\gamma_t} - 1 \right) A^{-1} X_{j+1}^t + \gamma_j A^{-1} \sum_{k=j+1}^{t} \left(\frac{1}{\gamma_k} - \frac{1}{\gamma_{k-1}} \right) X_{j+1}^{k-1}
\end{aligned}$$

It is clear that from the first line of above equation that $\overline{X}_j^t - \sum_{i=j}^t \gamma_i X_j^{i-1} > 0$. Hence we obtain the first inequality of the lemma. We have

$$(1 - \lambda_0 \gamma_k)^{-1} I \leq (I - \gamma_k A)^{-1}$$

By Lemma 4, we have

$$\left(\frac{1}{\gamma_{k+1}} - \frac{1}{\gamma_k} \right) \frac{1}{\gamma_{k+1}} I \leq \left(\frac{1}{\gamma_k} - \frac{1}{\gamma_{k-1}} \right) \frac{1}{\gamma_k} (I - \gamma_k A)^{\kappa - 1}$$

Hence

$$\left(\frac{1}{\gamma_k} - \frac{1}{\gamma_{k-1}} \right) \frac{1}{\gamma_k} X_{j+1}^{k-1} \leq \left(\frac{1}{\gamma_{j+1}} - \frac{1}{\gamma_j} \right) \frac{1}{\gamma_{j+1}} (X_{j+1}^{k-1})^{\kappa}$$

Define Y_j^k as $Y_j^k = \prod_{i=j}^k (I - \kappa \gamma_i A)$. Since $0 < \kappa \leq 1$, we have $(X_j^k)^\kappa \leq Y_j^k$. Hence

$$\begin{aligned}
\overline{X}_j^t - \sum_{i=j}^{t} \gamma_i X_j^{i-1} &\leq -\left(\frac{\gamma_j}{\gamma_t} - 1 \right) A^{-1} X_{j+1}^t + \gamma_j \left(\frac{1}{\gamma_{j+1}} - \frac{1}{\gamma_j} \right) \frac{1}{\gamma_{j+1}} A^{-1} \sum_{k=j+1}^{t} \gamma_k (X_{j+1}^{k-1})^{\kappa} \\
&\leq -\left(\frac{\gamma_j}{\gamma_t} - 1 \right) A^{-1} X_{j+1}^t + \frac{\gamma_j - \gamma_{j+1}}{\gamma_{j+1}^2} A^{-1} \sum_{k=j+1}^{t} \gamma_k Y_{j+1}^{k-1} \\
&= -\left(\frac{\gamma_j}{\gamma_t} - 1 \right) A^{-1} X_{j+1}^t + \frac{\gamma_j - \gamma_{j+1}}{\kappa \gamma_{j+1}^2} A^{-2} (I - Y_{j+1}^t) \\
&\leq \frac{\gamma_0}{\kappa \gamma_1} \frac{\gamma_j - \gamma_{j+1}}{\gamma_j \gamma_{j+1}} A^{-2} = \frac{1}{\kappa \gamma_1} ((1 + a\gamma_0(j+1))^c - (1 + a\gamma_0 j))^c) A^{-2}
\end{aligned}$$

$$\begin{aligned} &\leq \frac{ac\gamma_0(1+a\gamma_0 j)^{c-1}}{\kappa\gamma_1}A^{-2} \leq \frac{ac\gamma_0(1+a\gamma_0 j)^{c-1}}{\kappa\gamma_1}\frac{A^{-1}}{\lambda_0} \\ &= c_0(1+a\gamma_0 j)^{c-1}A^{-1} \end{aligned}$$

Now plugging (14) into above inequality, we obtain the claim of the lemma. □

With Lemma 5, we can now prove Theorem 1.

Proof. (Theorem 1) From (5), we get

$$\Delta_t = \Delta_{t-1} - \gamma_t(A\Delta_{t-1} + \xi_t) \quad , \quad \overline{\Delta}_t = \frac{1}{t}\sum_{i=1}^{t}\Delta_i \tag{15}$$

From (15), we have

$$\Delta_t = \prod_{j=1}^{t}(I-\gamma_j A)\Delta_0 + \sum_{j=1}^{t}\prod_{i=j+1}^{t}(I-\gamma_i A)\gamma_j\xi_j$$

then

$$\begin{aligned} \overline{\Delta}_t &= \frac{1}{t}\sum_{j=1}^{t}\Delta_j = \frac{1}{t}\sum_{j=1}^{t}\prod_{i=1}^{j}(I-\gamma_i A)\Delta_0 + \frac{1}{t}\sum_{j=1}^{t}\left(\sum_{k=j}^{t}\prod_{i=j+1}^{k}(I-\gamma_i A)\right)\gamma_j\xi_j \\ &= \frac{1}{\gamma_0 t}(\overline{X}_0^t - \gamma_0 I)\Delta_0 + \frac{1}{t}\sum_{j=1}^{t}\overline{X}_j^t\xi_j = I^{(0)} + I^{(1)} \end{aligned}$$

where $\overline{X}_j^t$ is defined in Lemma 5. Hence

$$tE(\|I^{(0)}\|_A^2) = \frac{1}{\gamma_0^2 t}\Delta_0^T A(\overline{X}_0^t - \gamma_0 I)^2\Delta_0 \leq \frac{(1+c_0)^2}{\gamma_0^2 t}\Delta_0^T A^{-1}\Delta_0 \tag{16}$$

$$\begin{aligned} &tE(\|I^{(1)}\|_A^2) = \frac{1}{t}\sum_{j=1}^{t}E(\xi_j^T A(\overline{X}_j^t)^2\xi_j) \leq \frac{1}{t}\sum_{j=1}^{t}(1+c_0(1+a\gamma_0 j)^{c-1})^2 E(\xi_t^T A^{-1}\xi_t) \\ \leq &\left(1 + \frac{2c_0+c_0^2}{t}\sum_{j=1}^{t}(1+a\gamma_0 j)^{c-1}\right)\mathrm{tr}(A^{-1}S) \leq \left(1 + \frac{(2c_0+c_0^2)((1+a\gamma_0 t)^c - 1)}{ac\gamma_0 t}\right)\mathrm{tr}(A^{-1}S) \\ \leq &\left(1 + \frac{(2c_0+c_0^2)(1+a\gamma_0 t)^{c-1}}{c}\right)\mathrm{tr}(A^{-1}S) \end{aligned} \tag{17}$$

And we have $E((I^{(0)})^T A I^{(1)}) = 0$ since $E(\xi_j) = 0$. □

Proof. (Lemma 1)

$$tE\|I^{(2)}\|_A^2 = tE\left\|\frac{1}{t}\sum_{j=1}^{t}\overline{X}_j^t\xi_j^{(2)}\right\|_A^2 = \frac{1}{t}\sum_{j=1}^{t}E\|\overline{X}_j^t\xi_j^{(2)}\|_A^2$$

$$
\begin{aligned}
&= \frac{1}{t}\sum_{j=1}^{t} E(\xi_j^{(2)T} A(\overline{X}_j^t)^2 \xi_j^{(2)}) \leq \frac{1}{t}\sum_{j=1}^{t}(1+c_0)^2 E(\xi_j^{(2)T} A^{-1} \xi_j^{(2)}) \\
&\leq \frac{1}{t}\sum_{j=1}^{t}(1+c_0)^2 c_1 E(\|\Delta_{j-1}\|_A^2) \leq \frac{(1+c_0)^2 c_1}{t}\left((1+c_2)\|\Delta_0\|_A^2 + c_3\sum_{j=1}^{t-1}\gamma_j\right) \\
&\leq \frac{(1+c_0)^2 c_1}{t}\left((1+c_2)\|\Delta_0\|_A^2 + \frac{c_3((1+a\gamma_0 t)^{1-c}-1)}{a(1-c)}\right) \\
&\leq (1+c_0)^2 c_1\left(\frac{1+c_2}{t}\|\Delta_0\|_A^2 + \frac{c_3\gamma_0}{1-c}(1+a\gamma_0 t)^{-c}\right)
\end{aligned}
$$

□

Proof. (Lemma 2) Let $\Sigma_x = E(xx^T)$. We have the following:

$$
\begin{aligned}
&g(\theta, d) = \frac{\partial l(\theta, d)}{\partial \theta} = xx^T\theta - xy \\
&\overline{g}(\theta) = E(g(\theta, d)) = \Sigma_x\theta - E(xy) \\
&A = \Sigma_x \quad , \quad b = E(xy) \quad , \quad \theta^* = A^{-1}b \\
&\xi^{(2)} = g(\theta, d) - g(\theta^*, d) - \overline{g}(\theta) = (xx^T - \Sigma_x)(\theta - \theta^*) \\
&E\left(\|\xi^{(2)}\|_{A^{-1}}^2 \middle| \theta\right) = (\theta-\theta^*)^T E(xx^T A^{-1} xx^T - \Sigma_x A^{-1}\Sigma_x)(\theta-\theta^*) \qquad (18)
\end{aligned}
$$

By the assumption of this lemma, we get

$$
E(xx^T A^{-1} xx^T) \leq \frac{1}{\lambda_0}E(xx^T xx^T) \leq \frac{M}{\lambda_0}A \qquad (19)
$$

From (18) and (19), we get

$$
E\left(\|\xi^{(2)}\|_{A^{-1}}^2 \middle| \theta\right) \leq \frac{M}{\lambda_0}\|\theta - \theta^*\|_A^2
$$

□

Lemma 6. *For linear regression problem $l(\theta, x, y) = \frac{1}{2}(\theta^T x - y)^2$, assuming all $\|x\|^2$ are M, then (2) will diverge if learning rate is greater than $\frac{2}{M}$.*

Proof. Let X_i^t be defined as in Lemma 5. We obtain the following from (2),

$$
\Delta_t = (I - \gamma_t x_t x_t^T)\Delta_{t-1} - \gamma_t(x_t x_t^T\theta^* - x_t y_t)
$$

Let $A_t = x_t x_t^T$, $b_t = x_t y_t$, $A = E(A_t)$, $b = E(b_t)$. Taking expectation with respect to x_t, y_t, noticing that $A\theta^* = b$, we get

$$
\begin{aligned}
E(\Delta_t|\theta_{t-1}) &= (I - \gamma_t A)\Delta_{t-1} \\
E(\|\Delta_t\|^2|\Delta_{t-1}) &= \Delta_{t-1}^T E(I - 2\gamma_t A + \gamma_t^2 A_t A_t)\Delta_{t-1} \\
&\quad + \gamma_t^2 E(\|A_t\theta^* - b_t\|^2) + 2\gamma_t^2 E(\theta^{*T} A_t A_t - b_t^T A_t)\Delta_{t-1} \\
&= \|\Delta_{t-1}\|^2 - (2\gamma_t - M\gamma_t^2)\|\Delta_{t-1}\|_A^2 + \gamma_t^2 \mathrm{tr}(S) + 2\gamma_t^2 u^T \Delta_{t-1}
\end{aligned}
$$

where $S = E((A_t\theta^* - b_t)(A_t\theta^* - b_t)^T)$, $u = E(A_t A_t\theta^* - A_t b_t)$. Hence

$$
E(\|\Delta_t\|^2) = E(\|\Delta_{t-1}\|^2) - (2\gamma_t - M\gamma_t^2)E(\|\Delta_{t-1}\|_A^2) + \gamma_t^2 \mathrm{tr}(S) + 2\gamma_t^2 u^T X_1^{t-1}\Delta_0
$$

If $\gamma_t >= \frac{2}{M} + \delta > \frac{2}{M}$, then

$$\begin{aligned} E(\|\Delta_t\|^2) &\geq E(\|\Delta_{t-1}\|^2) + \delta(2+\delta M)E(\|\Delta_{t-1}\|_A^2) + \gamma_t^2 \mathrm{tr}(S) + 2\gamma_t^2 u^T X_1^{t-1}\Delta_0 \\ &\geq (1+\lambda_0\delta(2+\delta M))E(\|\Delta_{t-1}\|^2) + \gamma_t^2 \mathrm{tr}(S) + 2\gamma_t^2 u^T X_1^{t-1}\Delta_0 \end{aligned}$$

Noticing that $X_1^{t-1} \to 0$ as $t \to \infty$, we conclude that $E(\|\Delta_t\|^2)$ is diverging if $\gamma_t \geq \frac{2}{M}$. □

Proof. (Lemma 3) Let $\gamma_i^t = \sum_{j=i}^t \gamma_j$,

$$\begin{aligned} tE\|I^{(3)}\|_A^2 &\leq \frac{1}{t}\sum_{j=1}^t E\|\overline{X}_j^t \xi_j^{(3)}\|_A^2 + \frac{2}{t}\sum_{j=1}^t \sum_{k=j+1}^t E(\xi_j^{(3)T}\overline{X}_j^t A \overline{X}_k^t \xi_k^{(3)}) \\ &\leq \frac{1}{t}\sum_{j=1}^t (1+c_0)^2 E\|\xi_j^{(3)}\|_{A^{-1}}^2 + \frac{2}{t}\sum_{j=1}^t \sum_{k=j+1}^t (1+c_0)^2 E(\|\xi_j^{(3)}\|_{A^{-1}}\|\xi_k^{(3)}\|_{A^{-1}}) \\ &\leq \frac{(1+c_0)^2 c_4^2}{t}\left(\sum_{j=1}^t E\|\Delta_j\|_A^4 + 2\sum_{j=1}^t \sum_{k=j+1}^t E(\|\Delta_j\|_A^2\|\Delta_k\|_A^2)\right) \\ &\leq \frac{(1+c_0)^2 c_4^2}{t}\left(\sum_{j=1}^t E\|\Delta_j\|_A^4 + 2\sum_{j=1}^t E\left(\|\Delta_j\|_A^2 \sum_{k=j+1}^t E(\|\Delta_k\|_A^2|\theta_j)\right)\right) \\ &\leq \frac{(1+c_0)^2 c_4^2}{t}\left(\sum_{j=1}^t E\|\Delta_j\|_A^4 + 2\sum_{j=1}^t E\left(\|\Delta_j\|_A^2\left(c_2\|\Delta_j\|_A^2 + c_3\sum_{k=j+1}^t \gamma_k\right)\right)\right) \\ &\leq \frac{(1+c_0)^2 c_4^2}{t}\left((1+2c_2)\sum_{j=1}^t E\|\Delta_j\|_A^4 + c_6\gamma_1^t) + 2c_3\sum_{j=1}^t E(\|\Delta_j\|_A^2)\sum_{k=j+1}^t \gamma_k\right) \\ &= \frac{(1+c_0)^2 c_4^2}{t}\left((1+2c_2)(c_5\|\Delta_0\|_A^4 + c_6\gamma_1^t) + 2c_3\sum_{k=2}^t \gamma_k \sum_{j=1}^{k-1} E(\|\Delta_j\|_A^2)\right) \\ &\leq \frac{(1+c_0)^2 c_4^2}{t}\left((1+2c_2)(c_5\|\Delta_0\|_A^4 + c_6\gamma_1^t) + 2c_3\sum_{k=2}^t \gamma_k(c_2\|\Delta_0\|_A^2 + c_3\gamma_1^{k-1})\right) \\ &\leq \frac{(1+c_0)^2 c_4^2}{t}\left((1+2c_2)(c_5\|\Delta_0\|_A^4 + c_6\gamma_1^t) + 2c_2c_3\|\Delta_0\|_A^2\gamma_1^t + c_3^2(\gamma_1^t)^2\right) \\ &\leq \frac{(1+c_0)^2 c_4^2}{t}\left((1+2c_2)c_5\|\Delta_0\|_A^4 + (2c_2c_3\|\Delta_0\|_A^2 + (1+2c_2)c_6)\gamma_1^t + c_3^2(\gamma_1^t)^2\right) \end{aligned}$$

□