SADAGRAD: Strongly Adaptive Stochastic Gradient Methods

Zaiyi Chen [*1 2] **Yi Xu** [*2] **Enhong Chen** [1] **Tianbao Yang** [2]

Abstract

Although the convergence rates of existing variants of ADAGRAD have a better dependence on the number of iterations under the strong convexity condition, their iteration complexities have a explicitly linear dependence on the dimensionality of the problem. To alleviate this bad dependence, we propose a simple yet novel variant of ADAGRAD for stochastic (weakly) strongly convex optimization. Different from existing variants, the proposed variant (referred to as SADAGRAD) uses an adaptive restarting scheme in which (i) ADAGRAD serves as a sub-routine and is restarted periodically; (ii) the number of iterations for restarting ADAGRAD depends on the history of learning that incorporates knowledge of the geometry of the data. In addition to the adaptive proximal functions and adaptive number of iterations for restarting, we also develop a variant that is adaptive to the (implicit) strong convexity from the data, which together makes the proposed algorithm strongly adaptive. In the worst case SADAGRAD has an $O(1/\epsilon)$ iteration complexity for finding an ϵ-optimal solution similar to other variants. However, it could enjoy faster convergence and much better dependence on the problem's dimensionality when stochastic gradients are sparse. Extensive experiments on large-scale data sets demonstrate the efficiency of the proposed algorithms in comparison with several variants of ADAGRAD and stochastic gradient method.

1. Introduction

ADAGRAD is a well-known method for general online and stochastic optimization that adopts a step size adaptive to each feature based on the learning history observed in earlier iterations. It has received tremendous interests for solving big data learning problems (e.g., see (Dean et al., 2012)). A rigorous regret analysis of ADAGRAD for online convex optimization was provided in the original paper (Duchi et al., 2011), which can be easily translated into a convergence result in expectation for stochastic convex optimization.

In spite of the claimed/observed advantage of ADAGRAD over stochastic gradient descent (SGD) method for general stochastic convex optimization (SCO), its benefit for stochastic strongly convex optimization (SSCO) over SGD diminishes due to its linear dependence on the problem's dimensionality and marginal benefit from sparse stochastic gradients. In particular, SGD and its variants can enjoy a convergence rate of $O(G^2/(\lambda T))$ for SSCO (Hazan & Kale, 2011; Rakhlin et al., 2012; Shamir & Zhang, 2013; Lacoste-Julien et al., 2012), where T is the number of iterations, G is a variance bound of stochastic gradient and $O(\cdot)$ only hides a constant factor independent of the problem's dimensionality. In contrast, the convergence result of ADAGRAD for strongly convex function (Duchi et al., 2010) implies an iteration complexity of $O(G_\infty^2 \sum_{i=1}^d \log(\|g_{1:T,i}\|_2^2)/(\lambda T))$, where $g_{1:T,i}$ is a vector of historical stochastic gradients of the i-th dimension, and G_∞ is the upper bound of stochastic gradient's infinity norm. It is notable that $\|g_{1:T,i}\|_2$ enters into the logarithmic function, which gives marginal adaptive benefit from sparse stochastic gradients and also a linear dependence on the dimensionality d even in the presence of sparse stochastic gradients. Such dependence also exists in other variants such as SC-ADAGRAD (Mukkamala & Hein, 2017) and MetaGrad (van Erven & Koolen, 2016) for SSCO, which are two recent variants of ADAGRAD with rigorous regret analysis.

It remains an open problem how to develop a variant of ADAGRAD that can enjoy greater benefit from sparse stochastic gradients and better dependence on the problem's dimensionality while still enjoying $O(1/T)$ convergence rate for SSCO. This paper provides an affirmative solution to address the problem. In particular, we consider the following SCO problem:

$$\min_{\mathbf{w}\in\Omega} F(\mathbf{w}) \tag{1}$$

where $\Omega \subseteq \mathbb{R}^d$ is a closed convex set and $F(\mathbf{w})$ is a proper lower semi-continuous convex function that satisfies (2).

[*]Equal contribution [1]University of Science and Technology of China, China [2]The University of Iowa, USA. Correspondence to: Zaiyi Chen <czy6516@mail.ustc.edu.cn>, Yi Xu <yi-xu@uiowa.edu>, Tianbao Yang <tianbao-yang@uiowa.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

The stochasticity is in the access model (Hazan & Kale, 2011): the only access to $F(\mathbf{w})$ is via a stochastic subgradient oracle, which given any point $\mathbf{w} \in \Omega$, produces a random vector $\mathbf{g}(\mathbf{w})$ whose expectation is a subgradient of $F(\mathbf{w})$ at the point $\mathbf{w}$, i.e., $\mathrm{E}[\mathbf{g}(\mathbf{w})] \in \partial F(\mathbf{w})$, where $\partial F(\mathbf{w})$ denotes the subdifferential set of F at $\mathbf{w}$. The above SCO includes an important family of problems where $F(\mathbf{w}) = \mathrm{E}_\xi[f(\mathbf{w}; \xi)]$, and $f(\mathbf{w}, \xi)$ is a proper lower semi-continuous convex function w.r.t $\mathbf{w}$ and depends on a random variable ξ.

If the function $F(\mathbf{w})$ is strongly convex, i.e., there exists $\lambda > 0$ such that for any $\mathbf{u}, \mathbf{v} \in \Omega$ we have $F(\mathbf{u}) - F(\mathbf{v}) \geq \partial F(\mathbf{v})^\top(\mathbf{w} - \mathbf{v}) + \frac{\lambda}{2}\|\mathbf{u} - \mathbf{v}\|_2^2$, then (1) becomes an instance of SSCO. In this paper, we consider the weaker strong convexity condition such that $F(\mathbf{w})$ satisfies the following inequality with $\lambda > 0$ for any $\mathbf{w} \in \Omega$:

$$\frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}_*\|_2^2 \leq F(\mathbf{w}) - F(\mathbf{w}_*), \tag{2}$$

where $\mathbf{w}_* \in \Omega$ is the closest optimal solution to $\mathbf{w}$. The above condition is also known as second-order growth condition in literature, which is implied by the strong convexity condition (Necoara et al., 2016). Nevertheless, many interesting problems in machine learning may not satisfy the strong convexity condition but can satisfy the above condition (please refer to Xu et al. (2017) for discussions and examples). Our major contributions are summarized below:

- We propose a novel variant of ADAGRAD, named SADAGRAD, for solving SSCO and more generally SCO that satisfies (2), which employs ADAGRAD as a sub-routine and adaptively restart it periodically.
- We provide a convergence analysis of SADAGRAD for achieving an ϵ-optimal solution, and demonstrate that it could enjoy greater benefit from sparse stochastic gradients and better dependence on the problem's dimensionality than ADAGRAD and its variants for SSCO.
- We also develop a proximal variant of SADAGRAD for stochastic composite optimization to reduce the effect of non-stochastic regularizer on the iteration complexity, and a practical variant that can be run without knowing the strong convexity parameter λ and hence can adapt to strong convexity from the data.

2. Related Work

SGD method has received a lot of attentions in the areas of machine learning and optimization, and many variants of SGD have been proposed and analyzed (Nemirovski et al., 2009; Rakhlin et al., 2012; Shamir & Zhang, 2013; Lacoste-Julien et al., 2012). It is well-known that SGD (with appropriate step sizes and averaging scheme) suffers from an $O(1/\epsilon^2)$ iteration complexity for solving a general SCO problem and enjoys an improved $O(1/\epsilon)$ iteration complexity for SSCO. When a non-stochastic regularizer is present in SCO, different proximal variants of SGD have been developed (Duchi et al., 2010; Xiao, 2010). In contrast with ADAGRAD, these algorithms use the same step size across all features, which could slow down the learning for rare features (with smaller gradients).

Epoch-GD (Hazan & Kale, 2011) is one variant of SGD that runs SGD in a stage-wise manner with an increasing number of iterations and a decreasing step size stage by stage. It was shown to achieve the optimal convergence rate for SSCO or more generally SCO that satisfies (2). Recently, Xu et al. (2017) developed a new variant of SGD (named ASSG) to leverage a local growth condition of the problem, which also runs SGD with multiple stages by halving the step size after each stage. Different from Epoch-GD, the number of iterations of each stage in ASSG is chosen based on the problem's local growth condition. For weakly strongly convex problems, a variant of ASSG (named RASSG) can be run without knowing the parameter λ in (2), and enjoys a similar iteration complexity to SGD for SSCO. A similar variant to ASSG also appears in (Chee & Toulis, 2018) but with no convergence analysis. The key differences between the proposed SADAGRAD and these variants of SGD are (i) ADAGRAD is used as a sub-routine of SADAGRAD; (ii) the number of iterations for each stage of SADAGRAD is adaptive to the history of learning instead of being a fixed sequence as in (Hazan & Kale, 2011; Xu et al., 2017). As a result, SADAGRAD enjoys more informative update and convergence.

Due to its adaptive step sizes for different features, ADAGRAD has recently witnessed great potential for solving deep learning problems (Dean et al., 2012) where there exists a large variation in terms of the magnitude of gradients across different layers. Several descendants of ADAGRAD have been developed and found to be effective for deep learning, e.g., ADAM (Kingma & Ba, 2015; Reddi et al., 2018). For general SCO, ADAM enjoys a similar convergence guarantee as ADAGRAD, while there is no formal theoretical convergence analysis for SSCO. We will compare with ADAM empirically for solving SCO problems.

The main competitors of SADAGRAD are variants of ADAGRAD for strongly convex functions (Duchi et al., 2010; Mukkamala & Hein, 2017; van Erven & Koolen, 2016). The advantage of SADAGRAD has been made clear in Introduction and will be explained more in next Section. In addition, MetaGrad (van Erven & Koolen, 2016) employs multiple copies of online Newton method to update the solution, which make it inefficient for high-dimensional data. Nevertheless, MetaGrad can also enjoy $O(d \log T/T)$ convergence for problems satisfying a Bernstein condition. It is not clear whether the analysis in (Duchi et al., 2010;

Algorithm 1 ADAGRAD($\mathbf{w}_0, \eta, \lambda, \epsilon, \epsilon_0$)

Input: $\eta > 0$, and $\mathbf{w}_0 \in \mathbb{R}^d$
Initialize: $\mathbf{w}_1 = \mathbf{w}_0$, $\mathbf{g}_{1:0} = []$, $H_0 \in \mathbb{R}^{d\times d}$
while T does not satisfy the condition in Proposition 1 **do**
  Compute a stochastic subgradient $\mathbf{g}_t$
  Update $g_{1:t} = [g_{1:t-1}, \mathbf{g}_t]$, $s_{t,i} = \|g_{1:t,i}\|_2$
  Set $H_t = H_0 + \text{diag}(\mathbf{s}_t)$ and $\psi_t(\mathbf{w}) = \frac{1}{2}(\mathbf{w} - \mathbf{w}_1)^\top H_t(\mathbf{w} - \mathbf{w}_1)$
  Let $\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\Omega} \eta\mathbf{w}^\top \left(\frac{1}{t}\sum_{\tau=1}^t \mathbf{g}_\tau\right) + \frac{1}{t}\psi_t(\mathbf{w})$
end while
Output: $\widehat{\mathbf{w}}_T = \sum_{t=1}^T \mathbf{w}_t/T$

Mukkamala & Hein, 2017) can be extended for problems with weak strong convexity condition (2).

3. Preliminaries and ADAGRAD

In this sequel, we let $\mathbf{g}_t = \mathbf{g}(\mathbf{w}_t)$ denote a stochastic subgradient of $F(\mathbf{w})$ at $\mathbf{w}_t$, i.e., $\mathrm{E}[\mathbf{g}_t] \in \partial F(\mathbf{w}_t)$. Given a vector $\mathbf{s}_t \in \mathbb{R}^d$, $\text{diag}(\mathbf{s}_t)$ denotes a diagonal matrix with entries equal to the corresponding elements in $\mathbf{s}_t$. Denote by I an identity matrix of an appropriate size. Denote by $g_{1:t} = [\mathbf{g}_1, \ldots, \mathbf{g}_t]$ a cumulative stochastic subgradient matrix of size $d \times t$, and by $g_{1:t,i}$ the i-th row of $g_{1:t}$.

Let $\mathcal{B}(\mathbf{x}_0, D)$ denote an Euclidean ball centered around $\mathbf{x}_0$ with a radius D. Let $\|\mathbf{w}\|_H = \sqrt{\mathbf{w}^T H\mathbf{w}}$ be a general norm where $H \succ 0$ is a positive definite matrix, and $\|\mathbf{w}\|_{H^{-1}} = \sqrt{\mathbf{w}^\top H^{-1}\mathbf{w}}$ be the dual norm. Let $\psi(\mathbf{x}; \mathbf{x}_0) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^\top H(\mathbf{x} - \mathbf{x}_0)$. It is straightforward to show that $\psi(\mathbf{x}; \mathbf{x}_0)$ is a 1-strongly convex function of $\mathbf{x}$ w.r.t. the norm $\|\mathbf{x}\|_H$. Similar to (Duchi et al., 2010; Mukkamala & Hein, 2017), we assume that the stochastic subgradient $\mathbf{g}(\mathbf{w})$ has a bounded infinity norm on Ω, i.e., $\|\mathbf{g}(\mathbf{w})\|_\infty \leq \gamma, \forall \mathbf{w} \in \Omega$. Given an initial solution $\mathbf{w}_0$, we assume that there exists $\epsilon_0 > 0$ such that $F(\mathbf{w}_0) - F(\mathbf{w}_*) \leq \epsilon_0$. For machine learning problems, $F(\mathbf{w}) \geq 0$ and hence an upper bound ϵ_0 of $F(\mathbf{w}_0)$ satisfies $F(\mathbf{w}_0) - F(\mathbf{w}_*) \leq \epsilon_0$.

Below, we first present a basic variant of ADAGRAD for solving (1). We note that the original paper developed and analyzed two basic variants of ADAGRAD with one based on the mirror descent update and the other one based on the primal-dual update. For sake of saving space, we here only present the primal-dual variant. However, our development can be extended to the mirror descent update. In addition, the original paper of ADAGRAD explicitly deals with a simple non-smooth component in the objective function by a proximal mapping. We leave this extension to Section 5. The detailed steps of the primal-dual variant of ADAGRAD with two modifications are presented in Algorithm 1.

The modifications in Algorithm 1 lies at that (i) a non-zero initial solution $\mathbf{w}_0$ is allowed and used for constructing the proximal function ψ_t; (ii) the algorithm is terminated by comparing the number of iterations T to a quantity given in the following Proposition. These two modifications are mainly for our development of SADAGRAD. The proposition below establishes iteration complexity of ADAGRAD for achieving an ϵ-optimal solution.

Algorithm 2 SADAGRAD($\mathbf{w}_0, \theta, \lambda, \epsilon, \epsilon_0$)

Input: $\theta > 0$, $\mathbf{w}_0 \in \mathbb{R}^d$ and $K = \log_2(\epsilon_0/\epsilon)$
for $k = 1, \ldots, K$ **do**
  Let $\eta_k = \theta\sqrt{\epsilon_k/\lambda}$, where $\epsilon_k = \epsilon_{k-1}/2$
  Let $\mathbf{w}_k = \text{ADAGRAD}(\mathbf{w}_{k-1}, \eta_k, \lambda, \epsilon_k, \epsilon_{k-1})$
end for
Output: $\mathbf{w}_K$

Proposition 1. *Let $\epsilon > 0$ be fixed, $H_0 = \gamma I$, $\gamma \geq \max_t \|\mathbf{g}_t\|_\infty$, and $\mathrm{E}[F(\mathbf{w}_1) - F(\mathbf{w}_*)] \leq \epsilon_0$. If $T \geq \frac{2}{\epsilon}\max\left\{\frac{\epsilon_0(\gamma+\max_i \|g_{1:T,i}\|_2)}{\eta\lambda}, \eta\sum_{i=1}^d \|g_{1:T,i}\|_2\right\}$, then Algorithm 1 gives a solution $\widehat{\mathbf{w}}_T$ such that $\mathrm{E}[F(\widehat{\mathbf{w}}_T) - F_*] \leq \epsilon$.*

Remark: The above result serves the foundation for our analysis, which is different from Eqn. (16) in (Duchi et al., 2011, Theorem 5). Note that T is now a random variable instead of a fixed value. In the proof presented in the supplement, we have to exploit the tool of stopping time of a matingale sequence.

4. Strongly Adaptive Stochastic Gradient Method (SADAGRAD)

In this section, we present SADAGRAD and its convergence analysis. The SADAGRAD shown in Algorithm 2 is built upon ADAGRAD in Algorithm 1. It runs with multiple stages and in each stage it employs ADAGRAD using the solution returned from the previous stage as the starting point and also as the reference point in the proximal function $\psi_t^k(\mathbf{w})$ during the k-th stage. In the following presentation, the notations with superscript k refer to the corresponding ones in the k-th call of Algorithm 1. For example, $\mathbf{g}_\tau^k$ refers to the stochastic subgradient at the τ-th iteration during the k-th call of ADAGRAD.

It is worth mentioning that SADAGRAD is similar to Epoch-GD (Hazan & Kale, 2011) in terms of multi-stage scheme. However, there still exist several key differences that are highlighted below: (i) Epoch-GD is developed with a fixed total number of iterations while SADAGRAD is developed for a fixed precision ϵ. (ii) the initial step size of both algorithms are different. The one in Epoch-GD is $1/\lambda$ and that in SADAGRAD is $\theta\sqrt{\epsilon_0/(2\lambda)}$. For problems with a very small strong convexity parameter, the initial step size of Epoch-GD is very large which usually leads to unstable performance at the beginning. (iii) The number of iterations

per-stage of both algorithms are different. In Epoch-GD, the number of iterations per-stage is geometrically increased by a fixed factor, while that in SADAGRAD is adaptive to the history of learning depending on the data as exhibited in the following theorem that states the convergence of SADAGRAD.

Theorem 1. *Consider SCO (1) with property (2) and a given $\epsilon > 0$. Assume $H_0 = \gamma I$ in Algorithm 1 and $\gamma \geq \max_{k,\tau} \|\mathbf{g}_\tau^k\|_\infty$, $F(\mathbf{w}_0) - F_* \leq \epsilon_0$ and t_k is the minimum number such that $t_k \geq \frac{2}{\sqrt{\lambda\epsilon_k}} \max\left\{\frac{2(\gamma+\max_i \|g^k_{1:t_k,i}\|_2)}{\theta}, \theta\sum_{i=1}^d \|g^k_{1:t_k,i}\|_2\right\}$, where $\theta > 0$ is a step size parameter of the algorithm. With $K = \lceil \log_2(\epsilon_0/\epsilon)\rceil$, we have $\mathrm{E}[F(\mathbf{w}_K) - F_*] \leq \epsilon$.*

Different from the convergence results of SGD and its variants, the above convergence result of SADAGRAD is adaptive to history of learning similar to that of ADAGRAD, therefore it deserves more explanation and understanding.

We first show that in the worst-case for dense stochastic gradients, SADAGRAD can enjoy an optimal iteration complexity of $O(1/(\lambda\epsilon))$. To see this, we can bound $\|g^k_{1:t_k,i}\|_2 \leq \sqrt{t_k}\gamma$, by choosing $\theta \propto 1/\sqrt{d}$, we have $\max\left\{\frac{2(\gamma+\max_i \|g^k_{1:t_k,i}\|_2)}{\theta}, \theta\sum_{i=1}^d \|g^k_{1:t_k,i}\|_2\right\} \leq O(\gamma\sqrt{dt_k})$, then $t_k = O(\frac{d\gamma^2}{\lambda\epsilon_k})$ satisfies the condition in Theorem 1, yielding a total iteration complexity of $O(d\gamma^2/(\lambda\epsilon))$. In comparison, the iteration complexity of Epoch-GD for SCO with property (2) is $O(\frac{G^2}{\lambda\epsilon})$, where G is the Euclidean norm bound of stochastic gradient, which is $\gamma\sqrt{d}$ under the assumption that $\|\mathbf{g}(\mathbf{w})\|_\infty \leq \gamma$. Therefore, in the worst-case for dense stochastic gradients SADAGRAD can enjoy the same iteration complexity of Epoch-GD.

Next, we compare SADAGRAD with ADAGRAD and SC-ADAGRAD for solving SSCO due to they have comparable computational costs per-iteration. Let us recall the convergence results of ADAGRAD and SC-ADAGRAD. For ADAGRAD, (Duchi et al., 2010)'s regret bound for strongly convex functions imply a convergence rate of

$$\frac{2\gamma^2\delta\|\mathbf{w}_0 - \mathbf{w}_*\|_2^2}{\lambda T} + \frac{\gamma^2}{\lambda T}\sum_{i=1}^d \log\left(\frac{\|g_{1:T,i}\|_2^2}{\delta} + 1\right). \quad (3)$$

For SC-ADAGRAD, (Mukkamala & Hein, 2017)'s regret bound implies a convergence rate of

$$\frac{\delta d D_\infty^2}{2\alpha T} + \frac{\alpha}{2T}\sum_{i=1}^d \log\left(\frac{\|g_{1:T,i}\|_2^2}{\delta} + 1\right), \quad (4)$$

where D_∞ is the upper bound of $\|\mathbf{w}_t - \mathbf{w}_*\|_\infty$ and $\alpha \geq \frac{\gamma^2}{2\lambda}$. It is notable that the above two convergence rates are in a similar order.

First, let us consider a scenario in which the growth of historical stochastic gradient vector $g_{1:t,i}$'s norm is slower than $\sqrt{t}$. Specially, let us assume $\|g^k_{1:t_k,i}\|_2 = O(\gamma t_k^\alpha)$ with $\alpha \leq 1/2$. Thus, with a proper value of θ [1] we have $t_k = O\left(\frac{d\gamma^2}{(\lambda\epsilon_k)^{1/2(1-\alpha)}}\right)$ satisfying the condition in Theorem 1. It is not difficult to show that $\sum_{k=1}^K t_k = O\left(\frac{d\gamma^2}{(\lambda\epsilon)^{1/2(1-\alpha)}}\right)$. Thus, when $\alpha < 1/2$, the iteration complexity of SADAGRAD is $o(d\gamma^2/(\lambda\epsilon))$ growing slower than $O(1/\epsilon)$ in terms of ϵ. In contrast, in this scenario the iteration complexity of both ADAGRAD and SC-ADAGRAD is $\widetilde{O}\left(\frac{d\gamma^2}{\lambda\epsilon}\right)$, which justifies the advantage of SADAGRAD. As an example to support this scenario, let us consider support vector machine where $F(\mathbf{w}) = 1/n\sum_i \ell_c(y_i\mathbf{w}^\top\mathbf{a}_i) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$, where $\ell_c(z) = \max(0, c - z)$, $y_i \in \{1, -1\}$ and $\mathbf{a}_i \in \mathbb{R}^d$ and $c > 0$ is a margin parameter. To consider a stochastic gradient, we ignore the regularizer for a moment [2] and have $\mathbf{g}_\tau^k = -\ell_c'(c - y^k_{i_\tau}(\mathbf{w}^k_\tau)^\top\mathbf{a}^k_{i_\tau})y^k_{i_\tau}\mathbf{a}^k_{i_\tau}$. Considering a linearly separable data set with a margin c, i.e., $y_i\mathbf{w}_*^\top\mathbf{a}_i \geq c, \forall i$, when $\mathbf{w}^k_\tau \to \mathbf{w}_*$, we have $\mathbf{g}^k_\tau \to 0$ and consequentially $\|g^k_{1:t_k,i}\|_2 \ll O(\sqrt{t_k})$ when t_k is large. For non-linearly separable data, we could expect that many components in $g_{1:t_k,i}$ will be zeros as $\mathbf{w}^k_\tau \to \mathbf{w}_*$ for those easily classified training examples, which could also render $\|g^k_{1:t_k,i}\|_2$ much smaller than $\sqrt{t_k}$.

Second, we use a synthetic example from (Duchi et al., 2011) to demonstrate that the iteration complexity of SADAGRAD could have a better dependence on d than that of ADAGRAD and SC-ADAGRAD in the presence of sparse stochastic gradients. In particular, Duchi et al. (2011) considered a sparse random data, at each iteration t feature i appears with probability $p_i = \min\{1, ci^{-\alpha}\}$ for some $\alpha \geq 2$ and a constant c. It was shown that $\mathrm{E}[\sum_{i=1}^d \|g^k_{1:t,i}\|_2] \leq O(\sqrt{t}\log d)$. Thus, SADAGRAD could enjoy an iteration complexity of $O(\frac{\gamma^2\log^2 d}{\lambda\epsilon})$ in expectation with an proper value of θ. In contrast, both ADAGRAD and SC-ADAGRAD still have an iteration complexity of $\widetilde{O}\left(\frac{\gamma^2 d}{\lambda\epsilon}\right)$.

Finally, we note that in practice a proper value of θ in SADAGRAD can be tuned by running a number of iterations of ADAGRAD to control the balance between $\frac{2(\delta+\max_i \|g^k_{1:t_k,i}\|_2)}{\theta}$ and $\theta\sum_{i=1}^d \|g^k_{1:t_k,i}\|_2$.

5. A Proximal Variant of SADAGRAD

In this section, we present a variant of SADAGRAD to handle a non-stochastic regularizer term by proximal mapping. To

[1] by which we mean that minimizes the lower bound of t_k in Theorem 1.

[2] it will be handled by a proximal mapping in next section, where $\mathbf{g}^k_\tau$ is still a stochastic gradient of the loss function.

Algorithm 3 ADAGRAD-PROX($\mathbf{w}_0, \eta, \lambda, \epsilon, \epsilon_0$)

1: **Input**: $\eta > 0$, and $\mathbf{w}_0 \in \mathbb{R}^d$
2: **Initialize:** $\mathbf{w}_1 = \mathbf{w}_0$, $\mathbf{g}_{1:0} = []$, H_0
3: **while** T does not satisfy the condition in Theorem 2 **do**
4: Compute a stochastic subgradient $\mathbf{g}_t$
5: Update $g_{1:t} = [g_{1:t-1}, \mathbf{g}_t]$, $s_{t,i} = \|g_{1:t,i}\|_2$
6: Set $H_t = H_0 + \text{diag}(\mathbf{s}_t)$ and $\psi_t(\mathbf{w}) = \frac{1}{2}(\mathbf{w} - \mathbf{w}_1)^\top H_t(\mathbf{w} - \mathbf{w}_1)$
7: Let $\mathbf{w}_{t+1} = \min_{\mathbf{w}\in\Omega} \eta \mathbf{w}^\top \sum_{\tau=1}^t \mathbf{g}_\tau/t + \eta\phi(\mathbf{w}) + \frac{1}{t}\psi_t(\mathbf{w})$
8: **end while**
9: **Output**: $\widetilde{\mathbf{w}}_T = \sum_{t=2}^{T+1} \mathbf{w}_t/T$

Algorithm 4 SADAGRAD-PROX($\mathbf{w}_0, \theta, \lambda, \epsilon, \epsilon_0$)

1: **Input**: $\theta > 0$, $\mathbf{w}_0 \in \mathbb{R}^d$ and $K = \log_2(\epsilon_0/\epsilon)$
2: **for** $k = 1, \ldots, K$ **do**
3: Let $\epsilon_k = \epsilon_{k-1}/2$
4: Let $\eta_k = \theta\sqrt{\epsilon_k/\lambda}$
5: $\mathbf{w}_k = $ ADAGRAD-PROX($\mathbf{w}_{k-1}, \eta_k, \lambda, \epsilon_k, \epsilon_{k-1}$).
6: **end for**
7: **Output**: $\mathbf{w}_K$

be formal, we consider the following SCO

$$\min_{\mathbf{w}\in\Omega} F(\mathbf{w}) \triangleq f(\mathbf{w}) + \phi(\mathbf{w}), \tag{5}$$

where the stochasticity lies in the access model of $f(\mathbf{w})$ and $\phi(\mathbf{w})$ is a non-stochastic regularizer. In this section, we abuse the notation $\mathbf{g}(\mathbf{w})$ to refer to the stochastic gradient of $f(\mathbf{w})$, and assume that $\|\partial f(\mathbf{w})\|_2 \leq G, \forall \mathbf{w} \in \Omega$.

Duchi et al. (2011) handled the $\phi(\mathbf{w})$ by a proximal mapping, i.e., replacing step 7 in Algorithm 1 by

$$\min_{\mathbf{w}\in\Omega} \eta \mathbf{w}^\top \sum_{\tau=1}^t \mathbf{g}_\tau/t + \eta\phi(\mathbf{w}) + \psi_t(\mathbf{w})/t. \tag{6}$$

Then they derived a similar convergence to their non-proximal setting (see Theorem 5 (Duchi et al., 2011)), where $\|g_{1:t,i}\|_2$ only captures the stochastic gradient of $f(\mathbf{w})$.

However, the proximal mapping will bring a new challenge when it is employed in the proposed SADAGRAD. To highlight the challenge, we first give a similar convergence to Proposition 1 for using the proximal mapping (6).

Proposition 2. *Let $H_0 = \gamma I$ and $\gamma \geq \max_t \|\mathbf{g}_t\|_\infty$. For any $T \geq 1$ and $\mathbf{w} \in \Omega$, we have*

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^T (f(\mathbf{w}_t) + \phi(\mathbf{w}_{t+1}) - f(\mathbf{w}) - \phi(\mathbf{w})) \\
\leq &\frac{\eta\sum_{i=1}^d \|g_{1:T,i}\|_2}{T} + \frac{\gamma + \max_i \|g_{1:T,i}\|_2}{2\eta T}\|\mathbf{w} - \mathbf{w}_1\|_2^2 \\
&+ \frac{1}{T}\sum_{t=1}^T (\mathrm{E}[\mathbf{g}_t] - \mathbf{g}_t)^\top(\mathbf{w}_t - \mathbf{w}).
\end{aligned}$$

Remark: Note that there is one solution shift between the stochastic component $f(\mathbf{w}_t)$ and the non-stochastic component $\phi(\mathbf{w}_{t+1})$. To handle such an issue, Duchi et al. (2011) used a trick by adding $[\phi(\mathbf{w}_1) - \phi(\mathbf{w}_{t+1})]/T$ on both sides, ad assume that $\phi(\mathbf{w}) \geq 0$ and $\phi(\mathbf{w}_1) = 0$ such that the added term on the R.H.S is less than zero which does not affect the analysis. Nonetheless, when we utilize the above result in the analysis of SADAGRAD, for epochs $k = 2, \ldots, K$ the initial solution $\mathbf{w}_{k-1}$ does not give a zero value of $\phi(\mathbf{w}_{k-1})$, which will cause the challenge in the analysis. A simple remedy is that we assume the non-stochastic component $\phi(\mathbf{w})$ is uniformly bounded over the domain Ω, i.e., $0 \leq \phi(\mathbf{w}) \leq B$. However, such an assumption may impose a strong restriction to the problem. For example if $\Omega = \mathbb{R}^d$ and $\phi(\mathbf{w}) = \|\mathbf{w}\|_2^2/2$, it does not satisfy the uniform boundness assumption.

To avoid introducing the uniform boundness assumption on $\phi(\mathbf{w})$, we propose to add $[f(\mathbf{w}_{T+1}) - f(\mathbf{w}_1)]/T$ on both sides, then we have the following inequality for running ADAGRAD with proximal mapping, i.e., Algorithm 3.

$$\begin{aligned}
&F(\widetilde{\mathbf{w}}_T) - F(\mathbf{w}) \qquad (7)\\
&\leq \frac{G\|\mathbf{w}_1 - \mathbf{w}_{T+1}\|_2}{T} + \frac{1}{T}\sum_{t=1}^T (\mathrm{E}[\mathbf{g}_t] - \mathbf{g}_t)^\top(\mathbf{w}_t - \mathbf{w}) \\
&+ \frac{\eta\sum_{i=1}^d \|g_{1:T,i}\|_2}{T} + \frac{\gamma + \max_i \|g_{1:T,i}\|_2}{2\eta T}\|\mathbf{w} - \mathbf{w}_1\|_2^2,
\end{aligned}$$

where $\widetilde{\mathbf{w}}_T = \sum_{t=2}^{T+1} \mathbf{w}_t/T$, and the first term in the R.H.S is due to G-Lipschitz continuity of $f(\mathbf{w})$.

We present the modified algorithm in Algorithm 4, where the name PROX refers to proximal mapping. The theorem below establishes the convergence guarantee of Algorithm 4.

Theorem 2. *For a given $\epsilon > 0$, let $K = \lceil \log_2(\epsilon_0/\epsilon) \rceil$. Assume $H_0 = \gamma I$ and $\gamma \geq \max_{k,\tau} \|\mathbf{g}_\tau^k\|_\infty$, $F(\mathbf{w}_0) - F_* \leq \epsilon_0$ and t_k is the minimum number such that $t_k \geq \frac{3}{\sqrt{\lambda\epsilon_k}} \max\left\{A_k, \frac{\sqrt{\lambda}G\|\mathbf{w}_1^k - \mathbf{w}_{t_k+1}^k\|_2}{\sqrt{\epsilon_k}}\right\}$, where $A_k = \max\left\{\frac{2(\gamma + \max_i \|g_{1:t_k,i}^k\|_2)}{\theta}, \theta\sum_{i=1}^d \|g_{1:t_k,i}^k\|_2\right\}$. Algorithm 4 guarantees that $\mathrm{E}[F(\mathbf{w}_K) - F_*] \leq \epsilon$.*

Remark: We note that there is an additional term in the lower bound of t_k, which comes from the first term in (7) compared to that in Theorem 1. The convergence of SADAGRAD-PROX is still adaptive to the history of updates, though the analysis of improvement compared with ADAGRAD becomes difficult due to the additional term dependent on $\|\mathbf{w}_1^k - \mathbf{w}_{t_k+1}^k\|_2$. Nevertheless, using the worst-case analysis, we can bound $\|\mathbf{w}_1^k - \mathbf{w}_{t_k+1}^k\|_2 \leq O(\frac{1}{\lambda})$ for SSCO according to (Hazan & Kale, 2011) (in their Lemma 4), which implies a worse-case iteration complexity of $O(\frac{1}{\lambda\epsilon})$ similar to that of Epoch-GD. However, in practice $\|\mathbf{w}_1^k - \mathbf{w}_{t_k+1}^k\|_2$ will decrease as the epoch number k

Algorithm 5 rSADAGRAD($\mathbf{w}_0, \theta, \lambda_1, \epsilon, \epsilon_0, \tau$)

1: **Input**: $\theta > 0$, $\mathbf{w}^{(0)} \in \mathbb{R}^d$, $\lambda_1 \geq \lambda$, $\tau \in (0, 1]$
2: **for** $s = 1, \ldots, S$ **do**
3: $\quad \mathbf{w}^{(s)} = \text{SADAGRAD}\ (\mathbf{w}^{(s-1)}, \theta, \lambda_s, \epsilon, \epsilon_{s-1})$
4: $\quad \lambda_{s+1} = \lambda_s/2$, $\epsilon_s = \tau\epsilon_{s-1}$
5: **end for**
6: **Output**: $\mathbf{w}^{(S)}$

increases, which can give much better performance than Epoch-GD and also faster convergence than SADAGRAD as observed in our experiments.

6. A Practical Variant of SADAGRAD

In this section, we provide a practical variant of SADAGRAD, which usually converges much faster. The issue that we aim to address is related to the value of λ. In practice, the strong convexity parameter λ is usually underestimated, yielding slower convergence. For example, for ℓ_2^2 norm regularized problems, the strong convexity parameter λ is usually set to the regularization parameter before the ℓ_2^2 norm. However, such an approach ignores the curvature in the loss functions defined over the data, which is difficult to estimate. In addition, for some problems that satisfy (2) the value of λ is difficult to estimate. For example, ℓ_1 regularized square loss minimization satisfy (2), where the value of λ is difficult to compute (Necoara et al., 2016, Theorem 10). To address this issue, we develop a restarting variant of SADAGRAD starting with a relatively large value of λ inspired by the technique in (Xu et al., 2017), which is presented in Algorithm 5 and its formal convergence guarantee is presented in the following theorem. Similar extension can be made to SADAGRAD-PROX and is omitted.

Theorem 3. *Under the same assumptions as Theorem 1 and* $F(\mathbf{w}_0) - F_* \leq \epsilon_0$*, where* $\mathbf{w}_0$ *is an initial solution. Let* $\epsilon \leq \frac{\epsilon_0}{2}$, $\tau = 1$, $K = \log_2 \frac{\epsilon_0}{\epsilon}$ *and* $t_k^{(s)} \geq \frac{2}{\sqrt{\lambda_s \epsilon_k}} \max\left\{ \frac{2(\gamma + \max_i \|g^k_{1:t_k,i}\|_2)}{\theta}, \theta \sum_{i=1}^d \|g^k_{1:t_k,i}\|_2 \right\}$. *Then with at most a total number of* $S = \left\lceil \log_2 \left(\frac{\lambda_1}{\lambda}\right) \right\rceil + 1$ *calls of* SADAGRAD *and a worse-cast iteration complexity of* $O(1/(\lambda\epsilon))$*, Algorithm 5 finds a solution* $\mathbf{w}^{(S)}$ *such that* $\mathrm{E}[F(\mathbf{w}^{(S)}) - F_*] \leq \epsilon$.

Remark: $t_k^{(s)}$ is the number of iterations in k-th stage of the s-th call of SADAGRAD. Note that $\lambda_s \geq \lambda$ for all $s \leq S - 1$, so the iteration complexity can be similarly understood as that in Theorem 1. However, since the algorithm rSADAGRAD starts with a relative large value of λ_1, we expect that the number of iterations in the first several calls of SADAGRAD can be much smaller than that of Algorithm 2, especially when the underlying strong convexity parameter λ is small. In practice, the parameter τ can be also tuned for better performance.

Table 1. Statistics of real data sets

data	#of instance	#of feature	feature density
covtype	581,012	54	22.12%
epsilon	400,000	2,000	100%
rcv1	697,641	47,236	0.15%
news20	19,996	1,355,191	0.0336%

7. Experiments

In this section, we present some experiments to show the effectiveness of the proposed algorithms, SADAGRAD and SADAGRAD-PROX. For all our experiments, we implement their practical variants referred to as rSADAGRAD and rSADAGRAD-PROX, respectively.

We will consider binary classification problems with two different formulations. The first formulation consists of smoothed hinge loss and an ℓ_1 norm regularization, i.e.,

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{w}) + \zeta\|\mathbf{w}\|_1, \tag{8}$$

where $f_i(\mathbf{w}) = \begin{cases} \frac{1}{2} - y_i\mathbf{w}^\top\mathbf{x}_i, & \text{if } y_i\mathbf{w}^\top\mathbf{x}_i \leq 0 \\ \frac{1}{2}(1 - y_i\mathbf{w}^\top\mathbf{x}_i)^2, & \text{if } 0 < y_i\mathbf{w}^\top\mathbf{x}_i \leq 1 \\ 0, & \text{otherwise} \end{cases}$,

$(\mathbf{x}_i, y_i), i = 1, \ldots, n$, is a set of training data examples, and ζ is the regularization parameter. The second formulation is the classical support vector machine (SVM) problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n \max\{0, c - y_i\mathbf{w}^\top\mathbf{x}_i\} + \lambda\|\mathbf{w}\|_2^2, \tag{9}$$

where c is the margin parameter and λ is the regularization parameter. It is notable that (8) is a piecewise convex quadratic problem, which satisfy the weak strong convexity condition (2) on a compact set according to (Li, 2013). However, the value of the strong convexity modulus λ is unknown for (8). (9) is a strongly convex problem, which satisfies (2) with λ being the regularization parameter. The experiments are performed on four data sets from libsvm (Chang & Lin, 2011) website with different scale of instances and features, namely covtype, epsilon, rcv1,and news20. The statistics of these data sets are shown in Table 1.

For solving (8), we compare SADAGRAD and SADAGRAD-PROX with ADAGRAD (Duchi et al., 2011), ADAM (Kingma & Ba, 2015), RASSG (Xu et al., 2017). Since the strong convexity modulus λ is unknown, we do not compare with Epoch-GD and SC-ADAGRAD (Mukkamala & Hein, 2017) for solving (8), which require knowing the value of λ. For solving (9), we compare SADAGRAD with ADAGRAD, ADAM, SC-ADAGRAD, RASSG and Epoch-

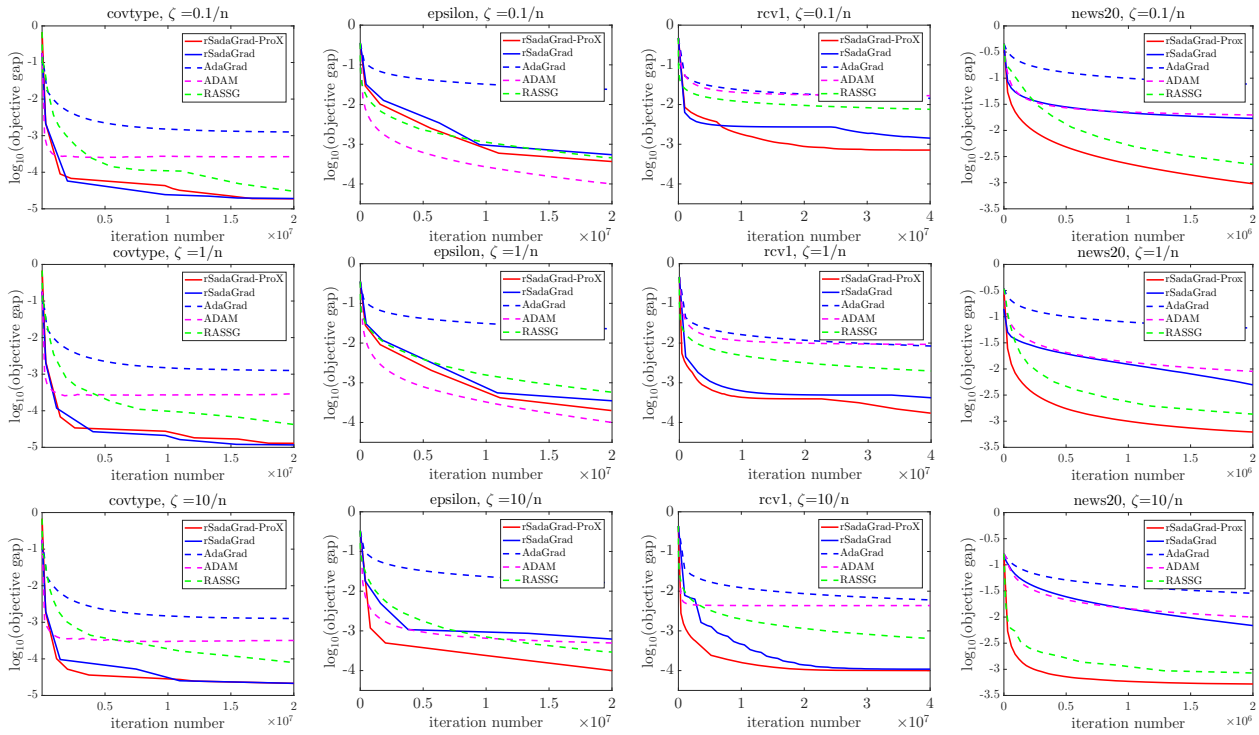

Figure 1. Results for smoothed hinge loss + ℓ_1 norm with varying ζ.

GD. Please note that RASSG can be considered an adaptive version of SGD that is adaptive to the problem's implicit strong convexity from the data, which is also observed to perform better than SGD in (Xu et al., 2017) and in our experiments. Hence SGD is omitted in our comparison. The parameters of Epoch-GD are chosen as recommended in the cited papers except for initial iteration number, which is tuned to achieve a faster convergence. The parameters for RASSG are tuned following the guidance in (Xu et al., 2017). The step size of ADAM is tuned in $10^{[-2:2]}$, and other parameters are chosen as recommended in the paper. For SC-ADAGRAD, the parameters α and ξ_1 in their papers are tuned in $10^{[-4:2]}$ and $[0.1, 1]$ respectively. Based on the analysis in the previous sections, the step size parameter θ would influence the convergence speed of both ADAGRAD and SADAGRAD. So we tuned this parameter for both ADAGRAD and SADAGRAD on each data set. We run ADAGRAD a number of iterations (i.e., 5,000) on each dataset and set $\theta = \sqrt{\frac{2(\gamma+\max_i \|\mathbf{g}^k_{1:5000,i}\|_2)}{\sum_{i=1}^d \|\mathbf{g}^k_{1:5000,i}\|_2}}$. Besides, we set $\lambda_1 = 100\lambda$ for solving (9) and $\lambda_1 = 100\zeta$ for solving (8) and $\tau = 1$ for rSADAGRAD and rSADAGRAD-PROX.

We first present and discuss the results for solving the ℓ_1 regularized smoothed hinge loss minimization problem (8) with varying regularization parameter ζ. The results are shown in Figure 1, where the y-axis is log-scale of the gap between the objective value of the obtained solution and that of the optimal solution. We have several observations from the results (i) rSADAGRAD-PROX performs consistently better than rSADAGRAD on high-dimensional data, especially on the extremely high-dimensional data news20. This is due to the presence of ℓ_1 norm regularization and the proximal mapping of rSADAGRAD-PROX that reduces the effect of the regularizer; (ii) in most cases rSADAGRAD-PROX has the best performance except on epsilon with two settings of $\zeta = 0.1/n, 1/n$, in which ADAM is better.

Next, we present the results for solving the SVM problem with varying λ and varying c. By varying c, we can control the growth of the stochastic gradient vector. The results of varying λ with fixed $c = 1$ on the four data sets are reported in Figure 2, and the results of varying c with fixed $\lambda = 1/n$ for the two data sets epsilon and rcv1 are reported in Figure 3. Here, we only report the results of r-SADAGRAD-PROX. From the results, we can see that SADAGRAD performs considerably faster than other baselines in most cases. In addition, we have several interesting observations that are consistent with our analysis and theory: (i) for smaller strong convexity parameter (corresponding to smaller values of λ), Epoch-GD perform poorly at earlier iterations due to very large step size; in contrast SADAGRAD doesn't suffer from this issue because of its unique step size scheme (c.f. the discussion above Theorem 1); (ii) ADAGRAD and SC-ADAGRAD perform poorly on the extremely high-dimensional data news20, especially when

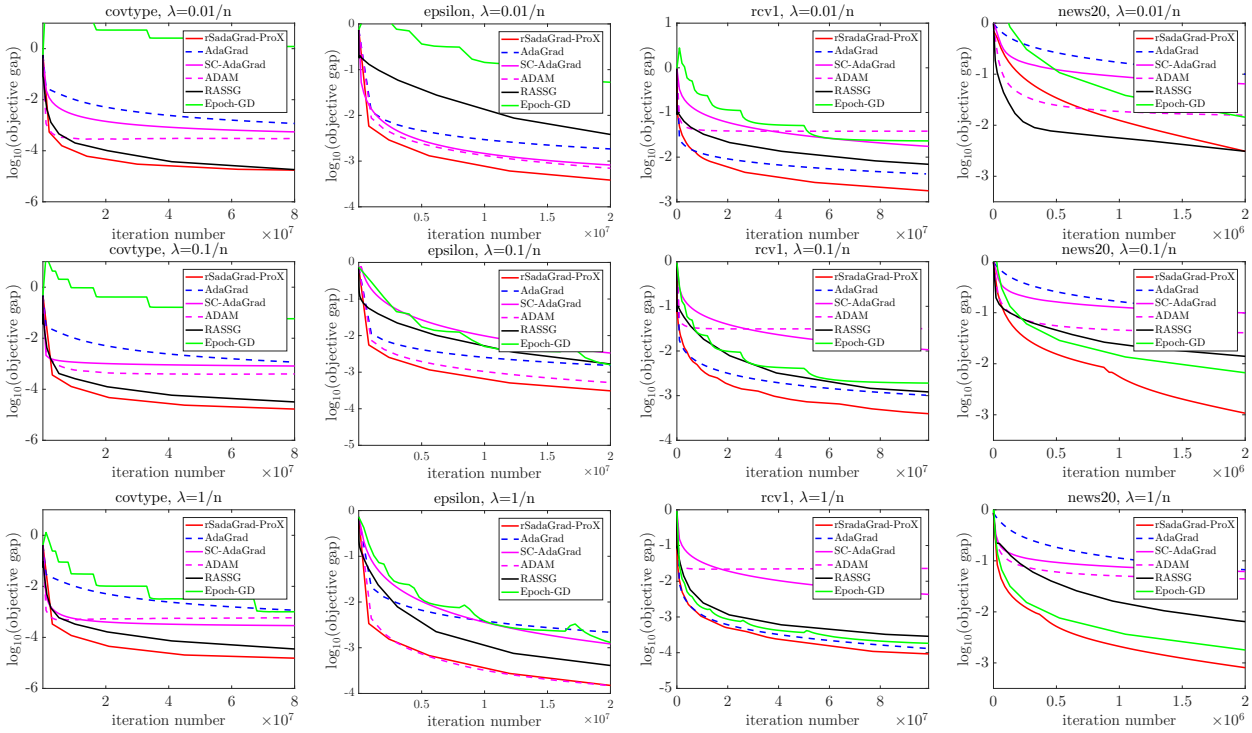

Figure 2. Results with varying λ for solving SVM.

λ is small. This is consistent with our prediction that their bad dependence on the dimensionality. In contrast, SADAGRAD is more robust to the high dimensionality and also enjoy adaptiveness to history of learning, making it better than Epoch-GD; (iii) for smaller margin parameter c, the improvement of SADAGRAD over ADAGRAD and SC-ADAGRAD becomes larger, which is consistent with our analysis (c.f., the discussion below Theorem 1). Overall, we can see that the proposed algorithms SADAGRAD achieve very promising results compared with existing adaptive and non-adaptive stochastic algorithms.

8. Conclusion

In this paper, we have proposed a simple yet novel variant of ADAGRAD, namely SADAGRAD, for solving stochastic strongly convex optimization and more generally stochastic convex optimization that satisfies the second order growth condition. We analyzed the iteration complexity of the proposed variant and demonstrated that it not only achieves the optimal iteration complexity but also enjoys faster convergence and better dependence on the problem's dimensionality when the stochastic gradients are sparse. We have also developed a proximal variant to reduce the effect of the non-stochastic regularizer. Experiments on large-scale real data sets demonstrate the effectiveness of the proposed SADAGRAD for solving ℓ_2 norm regularized hinge loss minimization problem and ℓ_1 regularized smoothed hinge loss minimization problem.

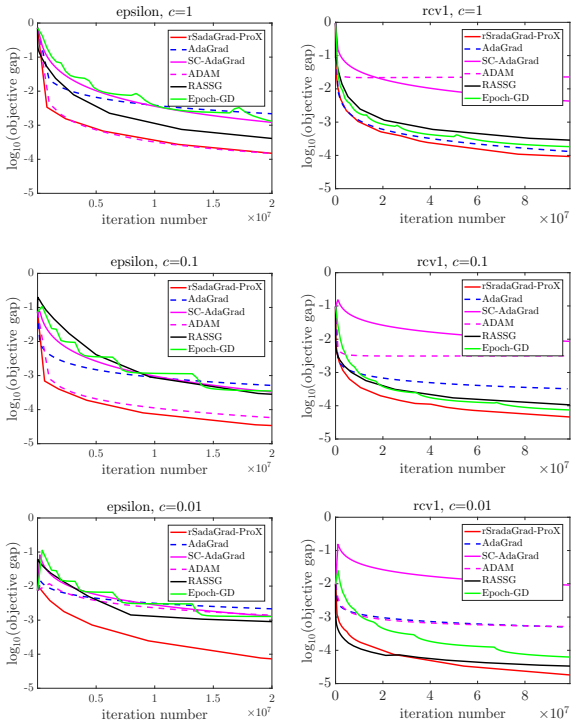

Figure 3. Results with varying c for solving SVM.

Acknowledgement

We thank the anonymous reviewers for their helpful comments. Most work of Z. Chen was done when he was visiting T. Yang's group at the University of Iowa. Y. Xu and T. Yang are partially supported by National Science Foundation (IIS-1545995). Z. Chen and E. Chen are partially supported by National Natural Science Foundation of China (Grants No. U1605251 and 61727809).

References

Chang, C.-C. and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

Chee, J. and Toulis, P. Convergence diagnostics for stochastic gradient descent with constant learning rate. In *International Conference on Artificial Intelligence and Statistics*, pp. 1476–1485, 2018.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Senior, A., Tucker, P., Yang, K., Le, Q. V., et al. Large scale distributed deep networks. In *Advances in neural information processing systems*, pp. 1223–1231, 2012.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Duchi, J. C., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. In *The 23rd Conference on Learning Theory (COLT)*, pp. 257–269, 2010.

Hazan, E. and Kale, S. Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. In *Proceedings of the 24th Annual Conference on Learning Theory (COLT)*, pp. 421–436, 2011.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Lacoste-Julien, S., Schmidt, M., and Bach, F. A simpler approach to obtaining an O(1/t) convergence rate for the projected stochastic subgradient method. *arXiv preprint arXiv:1212.2002*, 2012.

Li, G. Global error bounds for piecewise convex polynomials. *Mathematical Programming*, 137(1-2):37–64, 2013.

Mukkamala, M. C. and Hein, M. Variants of rmsprop and adagrad with logarithmic regret bounds. In *International Conference on Machine Learning*, pp. 2545–2553, 2017.

Necoara, I., Nesterov, Y., and Glineur, F. Linear convergence of first order methods for non-strongly convex optimization. *Mathematical Programming*, pp. 1–39, 2016.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

Rakhlin, A., Shamir, O., Sridharan, K., et al. Making gradient descent optimal for strongly convex stochastic optimization. In *International Conference on Machine Learning*, pp. 1571–1578, 2012.

Reddi, S. J., Kale, S., and Kumar, S. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

Shamir, O. and Zhang, T. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International Conference on Machine Learning*, pp. 71–79, 2013.

van Erven, T. and Koolen, W. M. Metagrad: Multiple learning rates in online learning. In *Advances in Neural Information Processing Systems*, pp. 3666–3674, 2016.

Xiao, L. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.

Xu, Y., Lin, Q., and Yang, T. Stochastic convex optimization: Faster local growth implies faster global convergence. In *International Conference on Machine Learning*, pp. 3821–3830, 2017.