Mixed Optimization for Smooth Functions

Authored by:

Rong Jin
Mehrdad Mahdavi
Lijun Zhang

Abstract

It is well known that the optimal convergence rate for stochastic optimization of smooth functions is $[O(1/sqrt{T})]$, which is same as stochastic optimization of Lipschitz continuous convex functions. This is in contrast to optimizing smooth functions using full gradients, which yields a convergence rate of $[O(1/T2)]$. In this work, we consider a new setup for optimizing smooth functions, termed as {bf Mixed Optimization}, which allows to access both a stochastic oracle and a full gradient oracle. Our goal is to significantly improve the convergence rate of stochastic optimization of smooth functions by having an additional small number of accesses to the full gradient oracle. We show that, with an $[O(ln T)]$ calls to the full gradient oracle and an $O(T)$ calls to the stochastic oracle, the proposed mixed optimization algorithm is able to achieve an optimization error of $[O(1/T)]$.

1 Paper Body

Many machine learning algorithms follow the framework of empirical risk minimization, which often can be cast into the following generic optimization problem n 1? (1) gi (w), min G(w) := w?W n i=1 where n is the number of training examples, gi (w) encodes the loss function related to the ith training example (xi , yi), and W is a bounded convex domain that is introduced to regularize the solution w ? W (i.e., the smaller the size of W, the stronger the regularization is). In this study, we focus on the learning problems for which the loss function gi (w) is smooth. Examples of smooth loss functions include least square with gi (w) = (yi ??w, xi ?)2 and logistic regression with gi (w) = log (1 + exp(?yi ?w, xi ?)). Since the regularization is enforced through the restricted domain W, we did not introduce a ?2 regularizer ??w?2 /2 into the optimization problem and as a result, we do not assume the loss function to be strongly convex. We note that a small ?2 regularizer does NOT improve the convergence rate of stochastic optimization. More specifically, the ? convergence rate?for stochastically optimizing a ?2 regularized loss function remains as O(1/ T) when ?

= O(1/ T) [11, Theorem 1], a scenario that is often encountered in real-world applications. A preliminary approach for solving the optimization problem in (1) is the batch gradient descent (GD) algorithm [16]. It starts with some initial point, and iteratively updates the solution using the equation wt+1 = ?W (wt ? ??G(wt)), where ?W (?) is the orthogonal projection onto the convex domain W. It has been shown that for smooth objective functions, the convergence rate of standard GD is O(1/T) [16], and can be improved to O(1/T 2) by an accelerated GD algorithm [15, 16, 18]. The main shortcoming of GD method is its high cost in computing the full gradient ?G(wt) when the number of training examples is large. Stochastic gradient descent (SGD) [3, 13, 21] alleviates this limitation of GD by sampling one (or a small set of) examples and computing a stochastic (sub)gradient at each iteration based on the sampled examples. Since the computational cost of SGD per iteration is independent of the size of the data (i.e., n), it is usually appealing for largescale learning and optimization. While SGD enjoys a high computational efficiency per iteration, it suffers from a slow convergence rate for optimizing smooth functions. It has been shown in [14] that the effect of the stochastic noise 1

Full (GD) Os ?1 1 0 T
Setting Lipschitz
Convergence
Smooth
1 T2
0
Stochastic (SGD) Os Of ?1 T 0 T
Of T
Convergence
T
?1 T
T
0
Mixed Optimization Os Of ? ? ?
Convergence 1 T
T
log T

Table 1: The convergence rate (O), number of calls to stochastic oracle (Os), and number of calls to full gradient oracle (Of) for optimizing Lipschitz continuous and smooth convex functions, using full GD, SGD, and mixed optimization methods, measured in the number of iterations T . ? cannot be decreased with a better rate than O(1/ T) which is significantly worse than GD that uses the full gradients for updating the solutions and this limitation is also valid when the target function is smooth. In addition, as we can see from Table 1, for general Lipschitz functions, SGD exhibits the same convergence rate as that for the smooth functions, implying that smoothness is essentially not very useful and can not be exploited in stochastic optimization. The slow convergence rate for stochastically optimizing smooth loss functions is mostly due to the variance in stochastic gradients: unlike the full gradient case where the norm of a gradient

approaches to zero when the solution is approaching to the optimal solution, in stochastic optimization, the norm of a stochastic gradient is constant even when the solution is close to the ? optimal solution. It is the variance in stochastic gradients that makes the convergence rate O(1/ T) unimprovable in smooth setting [14, 1]. In this study, we are interested in designing an efficient algorithm that is in the same spirit of SGD but can effectively leverage the smoothness of the loss function to achieve a significantly faster convergence rate. To this end, we consider a new setup for optimization that allows us to interplay between stochastic and deterministic gradient descent methods. In particular, we assume that the optimization algorithm has an access to two oracles: ? A stochastic oracle Os that returns the loss function gi (w) and its gradient based on the sampled training example (xi , yi) 2 , and ? A full gradient oracle Of that returns the gradient ?G(w) for any given solution w ? W. We refer to this new setting as mixed optimization in order to distinguish it from both stochastic and full gradient optimization models. The key question we examined in this study is: Is it possible to improve the convergence rate for stochastic optimization of smooth functions by having a small number of calls to the full gradient oracle Of ? We give an affirmative answer to this question. We show that with an additional O(ln T) accesses to the full gradient oracle Of , the proposed algorithm, referred to as M IXED G RAD, can improve the convergence rate for stochastic optimization of smooth functions to O(1/T), the same rate for stochastically optimizing a strongly convex function [11, 19, 23]. M IXED G RAD builds off on multistage methods [11] and operates in epochs, but involves novel ingredients so as to obtain an O(1/T) rate for smooth losses. In particular, we form a sequence of strongly convex objective functions to be optimized at each epoch and decrease the amount of regularization and shrink the domain as the algorithm proceeds. The full gradient oracle Of is only called at the beginning of each epoch. Finally, we would like to distinguish mixed optimization from hybrid methods that use growing sample-sizes as optimization method proceeds to gradually transform the iterates into the full gradient method [9] and batch gradient with varying sample sizes [6], which unfortunately make the iterations to be dependent to the sample size n as opposed to SGD. In contrast, M IXED G RAD is as an alternation of deterministic and stochastic gradient steps, with different of frequencies for each type of steps. Our result for mixed optimization is useful for the scenario when the full gradient of the objective function can be computed relatively efficient although it is still significantly more expensive than computing a stochastic gradient. An example of such a scenario is distributed computing where the computation of full gradients can be speeded up by having it run in parallel on many machines with each machine containing a relatively small subset of the entire training data. Of course, the latency due to the communication between machines will result in an additional cost for computing the full gradient in a distributed fashion. Outline The rest of this paper is organized as follows. We begin in Section 2 by briefly reviewing the literature on deterministic and stochastic optimization. In Section 3, we introduce the necessary definitions and discuss the assumptions that underlie our analysis. Section 4 describes the M IXED G RAD algorithm and states the main result

on its convergence rate. The proof of main result is given in Section 5. Finally, Section 6 concludes the paper and discusses few open questions. 1

The convergence rate can be improved to O(1/T) when the structure of the objective function is provided. We note that the stochastic oracle assumed in our study is slightly stronger than the stochastic gradient oracle as it returns the sampled function instead of the stochastic gradient. 2

2

2

More Related Work

Deterministic Smooth Optimization The convergence rate of gradient based methods usually depends on the analytical properties of the objective function to be optimized. When the objective function is strongly convex and smooth, it is well known that a simple GD method can achieve a linear convergence rate [5]. For a ? non-smooth Lipschitz-continuous function, the optimal rate for ? the first order method is only O(1/ T) [16]. Although O(1/ T) rate is not improvable in general, several recent studies are able to improve this rate to O(1/T) by exploiting the special structure of the objective function [18, 17]. In the full gradient based convex optimization, smoothness is a highly desirable property. It has been shown that a simple GD achieves a convergence rate of O(1/T) when the objective function is smooth, which is further can be improved to O(1/T 2) by using the accelerated gradient methods [15, 18, 16]. Stochastic Smooth Optimization Unlike the optimization methods based on full gradients, the smoothness assumption was ? not exploited by most stochastic optimization methods. In fact, it was shown in [14] that the O(1/ T) convergence rate for stochastic optimization cannot be improved even when the objective function is smooth. This classical result is further confirmed by the recent studies of composite bounds for the first order optimization methods [2, 12]. The smoothness of the objective function is exploited extensively in mini-batch stochastic optimization [7, 8], where the goal is not to improve the convergence rate but to reduce the variance in stochastic gradients and consequentially the number of times for updating the solutions [24]. We finally note that the smoothness assumption coupled with the strong convexity of function is beneficial in stochastic setting and yields a geometric convergence in expectation using Stochastic Average Gradient (SAG) and Stochastic Dual Coordinate Ascent (SDCA) algorithms proposed in [20] and [22], respectively.

3 Preliminaries We use bold-face letters to denote vectors. For any two vectors w, w? ? W, we denote by ?w, w? ? the inner product between w and w? . Throughout this paper, we only consider the ?2 -norm. We assume the objective function G(w) defined in (1) to be the average of n convex loss functions. The same assumption was made in [20, 22]. We assume that G(w) is minimized at some w? ? W. Without loss of generality, we assume that W ? BR , a ball of radius R. Besides convexity of individual functions, we will also assume that each gi (w) is ?-smooth as formally defined below [16]. Definition 1 (Smoothness). A differentiable loss function f (w) is said to be ?-smooth with respect to a norm ? ? ?, if it holds that ? f (w) ? f (w?) + ??f (w?), w ? w? ? + ?w ? w? ?2 , ? w, w? ? W, 2 The smoothness assumption also implies that

??f (w) ? ?f (w?), w ? w? ? ? ??w ? w? ?2 which is equivalent to ?f (w) being ?-Lipschitz continuous. In stochastic first-order optimization setting, instead of having direct access to G(w), we only have access to a stochastic gradient oracle, which given a solution w ? W, returns the gradient ?gi (w) where i is sampled uniformly at random from {1, 2, ? ? ? , n}. The goal of stochastic optimization ? ? W such that the optimization to use a bounded number T of oracle calls, and compute some w ? ? G(w?), is as small as possible. error, G(w) In the mixed optimization model considered in this study, we first relax the stochastic oracle Os by assuming that it will return a randomly sampled loss function gi (w), instead of the gradient ?gi (w) for a given solution w 3 . Second, we assume that the learner also has an access to the full gradient oracle Of . Our goal is to significantly improve the convergence rate of stochastic gradient descent (SGD) by making a small number of calls to the full gradient oracle Of . In particular, we show that by having only O(log T) accesses to the full gradient oracle and O(T) accesses to the stochastic oracle, we can tolerate the noise in stochastic gradients and attain an O(1/T) convergence rate for optimizing smooth functions. 3 The audience may feel that this relaxation of stochastic oracle could provide significantly more information, and second order methods such as Online Newton [10] may be applied to achieve O(1/T) convergence. We note (i) the proposed algorithm is a first order method, and (ii) although the Online ?Newton method yields a regret bound of O(1/T), its convergence rate for optimization can be as low as O(1/ T) due to the concentration bound for Martingales. In addition, the Online Newton method is only applicable to exponential concave function, not any smooth loss function.

3

Algorithm 1 M IXED G RAD Input: step size ?1 , domain size ?1 , the number of iterations T1 for the first epoch, the number of epoches m, regularization parameter ?1 , and shrinking parameter ? ¿ 1 ?1 = 0 1: Initialize w 2: for k = 1, . . . , m do 3: Construct the domain Wk = {w : w + wk ? W, ?w? ? ?k } ? k) 4: Call the full gradient oracle Of for ?G(w ? ? k + ?G(w ? k) = ?k w ? k + n1 ni=1 ?gi (w ? k) 5: Compute gk = ?k w 6: Initialize wk1 = 0 7: for t = 1, . . . , Tk do 8: Call stochastic oracle Os to return a randomly selected loss function gitk (w) ?kt = gk + ?gitk (wkt + w ? k) ? ?gitk (w ? k) 9: Compute the stochastic gradient as g 10: Update the solution by 1 ?kt + ?k wkt ? + ?w ? wkt ?2 wkt+1 = arg max ?k ?w ? wkt , g 2 w?Wk 11: end for ? k +1 ? k+1 = w ?k + w e k+1 e k+1 = Tk1+1 Tt=1 12: Set w wkt and w 13: Set ?k+1 = ?k /?, ?k+1 = ?k /?, ?k+1 = ?k /?, and Tk+1 = ? 2 Tk 14: end for ? m+1 Return w

The analysis of the proposed algorithm relies on the strong convexity of intermediate loss functions introduced to facilitate the optimization as given below. Definition 2 (Strong convexity). A function f (w) is said to be ?-strongly convex w.r.t a norm ? ? ?, if there exists a constant ? ¿ 0 (often called the modulus of strong convexity) such that it holds ? f (w) ? f (w?) + ??f (w?), w ? w? ? + ?w ? w? ?2 , ? w, w? ? W 2

4

Mixed Stochastic/Deterministic Gradient Descent

We now turn to describe the proposed mixed optimization algorithm and state its convergence rate. The detailed steps of M IXED G RAD algorithm are shown in Algorithm 1. It follows the epoch gradient descent algorithm proposed in [11] for stochastically minimizing strongly convex functions and divides the optimization process into m epochs, but involves novel ingredients so as to obtain an O(1/T) convergence rate. The key idea is to introduce a ?2 regularizer into the objective function to make it strongly convex, and gradually reduce the amount of regularization over the epochs. We also shrink the domain as the algorithm proceeds. We note that reducing the amount of regularization over time is closely-related to the classic proximal-point algorithms. Throughout the paper, we will use the subscript for the index of each epoch, and the superscript for the index of iterations within each epoch. Below, we describe the key idea behind M IXED G RAD. ? k be the solution obtained before the kth epoch, which is initialized to be 0 for the first epoch. Let w ? k , resulting in the following Instead of searching for w? at the kth epoch, our goal is to find w? ? w optimization problem for the kth epoch n 1? ?k ? k ?2 + ? k), ?w + w (2) gi (w + w min 2 n i=1 w + wk ? W ?w? ? ?k

where ?k specifies the domain size of w and ?k is the regularization parameter introduced at the kth epoch. By introducing the ?2 regularizer, the objective function in (2) becomes strongly convex, making it possible to exploit the technique for stochastic optimization of strongly convex function in order to improve the convergence rate. The domain size ?k and the regularization parameter ?k are initialized to be ?1 ¿ 0 and ?1 ¿ 0, respectively, and are reduced by a constant factor ? ¿ 1 ? k ?2 /2 every epoch, i.e., ?k = ?1 /? k?1 and ?k = ?1 /? k?1 . By removing the constant term ?k ?w from the objective function in (2), we obtain the following optimization problem for the kth epoch [ ] n 1? ?k 2 ? k? + ? k) , ?w? + ?k ?w, w gi (w + w (3) min Fk (w) = w?Wk 2 n i=1 4

where Wk = {w : w + wk ? W, ?w? ? ?k }. We rewrite the objective function Fk (w) as n ?k 1? ? k? + ? k) Fk (w) = ?w?2 + ?k ?w, w gi (w + w 2 n i=1 ? ? n n ?k 1? 1? 2 ?k + ? k) + ? k) ? ?w, ?gi (w ? k)? = ?w? + w, ?k w ?gi (w gi (w + w 2 n i=1 n i=1 ?k 1? k ?w?2 + ?w, gk ? + gb (w) 2 n i=1 i n

= where

(4)

1? ? k) and gbik (w) = gi (w + w ? k) ? ?w, ?gi (w ? k)?. ?gi (w n i=1 n ?k + gk = ?k w

The main reason for using gbik (w) instead of gi (w) is to tolerate the variance in the stochastic gradients. To see this, from the smoothness assumption of gi (w) we obtain the following inequality for the norm of gbik (w) as:

k

?b ? k) ? ?gi (w ? k)? ? ??w?. gi (w) = ??gi (w + w As a result, since ?w? ? ?k and ?k shrinks over epochs, then ?w? will approach to zero over epochs and consequentially ??b gik (w)? approaches to zero, which allows us to effectively control the variance in stochastic gradients, a key to improving the convergence of stochastic optimization for smooth functions to O(1/T). Using Fk (w) in

(4), at the tth iteration of the kth epoch, we call the stochastic oracle Os to randomly select a loss function gikt (w) and update the solution by following the standard paradigm of SGD by wkt+1

= =

) (gikt (wkt)) ?w?Wk wkt ? ?k (?k wkt + gk + ?b k ( ) t t ? k) ? ?gitk (w ? k)) , ?w?Wk wk ? ?k (?k wk + gk + ?gitk (wkt + w

(5)

where ?w?Wk (w) projects the solution w into the domain Wk that shrinks over epochs. e k , and update the solution from w ? k to At the end of each epoch, we compute the average solution w ? k+1 = w ?k +w e k . Similar to the epoch gradient descent algorithm [11], we increase the number of w iterations by a constant ? 2 for every epoch, i.e. Tk = T1 ? 2(k?1) . In order to perform stochastic gradient updating given in (5), we need to compute vector gk at the beginning of the kth epoch, which requires an access to the full gradient oracle Of . It is easy to count that the number of accesses to the full gradient oracle Of is m, and the number of accesses to the stochastic oracle Os is m ? ? 2m ? 1 T1 . T = T1 ? 2(i?1) = 2 ? ?1 i=1 Thus, if the total number of accesses to the stochastic gradient oracle is T , the number of access to the full gradient oracle required by Algorithm 1 is O(ln T), consistent with our goal of making a small number of calls to the full gradient oracle. The theorem below shows that for smooth objective functions, by having O(ln T) access to the full gradient oracle Of and O(T) access to the stochastic oracle Os , by running M IXED G RAD algorithm, we achieve an optimization error of O(1/T). Theorem 1. Let ? ? e?9/2 be the failure probability. Set ? = 2, ?1 = 16? and 1 m ? T1 = 300 ln , ?1 = , and ?1 = R. ? 2? 3T1 (2m) ? m+1 be the solution returned by Algorithm 1 after m epochs Define T = T1 2 ? 1 /3. Let w with m = O(ln T) calls to the full gradient oracle Of and T calls to the stochastic oracle Os . Then, with a probability 1 ? 2?, we have ( ) ? 80?R2 ? m+1) ? min G(w) ? 2m?2 = O . G(w w?W 2 T 5

5

Convergence Analysis

Now we turn to proving the main theorem. The proof will be given in a series of lemmas and theorems where the proof of few are given in the Appendix. The proof of main theorem is based b ?k be the optimal solution that minimizes Fk (w) defined in (3). on induction. To this end, let w b ?k ? ? ?k , with a high probability, it holds that The key to our analysis is show that when ?w b ?k+1 ? ? ?k /?, where w b ?k+1 is the optimal solution that minimizes Fk+1 (w), as revealed by the ?w following theorem. b ?k and w b ?k+1 be the optimal solutions that minimize Fk (w) and Fk+1 (w), reTheorem 2. Let w e k+1 be the average solution obtained at the end of (kth?epoch) of M IXED G RAD spectively, and w b ?k ? ? ?k . By setting the step size ?k = 1/ 2? 3Tk , we have, with a algorithm. Suppose ?w probability 1 ? 2?, ?k ?k ?2k b ?k+1 ? ? e k+1) ? min Fk (w) ? ?w and Fk (w w ? 2? 4 ?9/2 provided that ? ? e and 300? 8 ? 2 1 Tk ? ln . ?2k ? Taking this statement as given for the moment, we proceed with the proof of Theorem 1, returning later to establish the claim stated in Theorem 2. Proof of Theorem 1. It is easy to check that for the first

epoch, using the fact W ? BR , we have ?w?1 ? = ?w? ? ? R := ?1 . m b ?m+1 be the optimal solution obLet w? be the optimal solution that minimizes Fm (w) and let w tained in the last epoch. Using Theorem 2, with a probability 1 ? 2m?, we have ?1 ?m ?2m ?1 ?21 b ?m ? ? m?1 , Fm (w e m+1) ? Fm (w b ?m) ? ?w = 3m+1 4 ? 2? 2? Hence by expanding the left hand side and utilizing the smoothness of individual loss functions we get n 1? ?1 ?21 ?1 ? m+1) ? Fm (w b ?m) + 3m+1 e m+1 , w ? m? gi (w ? m?1 ?w n i=1 2? ? ? m ??1 ?1 ?21 ?1 ?w + 3m+1 2m?2 2? ? b ?m+1 ? ? ?m = ?1 ? 1?m . Since where the last step uses the fact ?w m m ? ? ??1 ? m? ? e i— ? ?w —w ?i ? ? 2?1 ??1 i=1 i=1 b ?m) + ? Fm (w

where in the last step holds under the condition ? ? 2. By combining above inequalities, we obtain n ?1 ?21 1? 2?1 ?2 ? m+1) ? Fm (w b ?m) + 3m+1 gi (w + 2m?21 . n i=1 2? ? b ?m minimizes Fm (w), for any w? ? Our final goal is to relate Fm (w) to minw G(w). Since w arg min G(w), we have n) 1? ?1 (? m ?2 + 2?w? ? w ? m, w ? m? . Fm (w?m) ? Fm (w?) = (6) gi (w?) + m?1 ?w? ? w n i=1 2? ? m ?. To this end, after the first m Thus, the key to bound —F(w?m) ? G(w?)— is to bound ?w? ? w ? m+1 , w ? m+2 , . . . be the sequence of solutions epoches, we run Algorithm 1 with full gradients. Let w generated by Algorithm 1 after the first m epochs. For this sequence of solutions, Theorem 2 e k ? ? ?k for any will hold deterministically as we deploy the full gradient for updating, i.e., ?w k ? m + 1. Since we reduce ?k exponentially, ?k will approach to zero and therefore the sequence ? k }? {w k=m+1 will converge to w? , one of the optimal solutions that minimize G(w). Since w? is the ? k }? ? k ? ? ?k for any k ? m + 1, we have limit of sequence {w k=m+1 and ?w ? ? ? ? 2?1 ?1 ? m? ? e i— ? ? m ?w? ? w —w ?k ? m ?1) ? (1 ? ? ? i=m+1 k=m+1

6

where the last step follows from the condition ? ? 2. Thus, (2) n 1? 4?1 8?21 ?1 m Fm (w?) ? gi (w?) + m?1 + m n i=1 2? ? 2m ? =

n n) 1? 2?1 ?2 (1? 5?1 ?2 gi (w?) + 2m?11 2 + ? ?m ? gi (w?) + 2m?11 n i=1 ? n i=1 ?

(7)

By combining the bounds in (6) and (7), we have, with a probability 1 ? 2m?, n n 1? 1? 5?1 ?2 ? m+1) ? gi (w gi (w?) ? 2m?21 = O(1/T) n i=1 n i=1 ? where T = T1

m?1 ? k=0

?

2k

( ) T1 ? 2m ? 1 T1 2m ? ? . = 2 ? ?1 3

We complete the proof by plugging in the stated values for ?, ?1 and ?1 . 5.1

Proof of Theorem 2

For the convenience of discussion, we drop the subscript k for epoch just to simplify our notation. ? =w ? k be the solution obtained before the start of Let ? = ?k , T = Tk , ? = ?k , g = gk . Let w ?? = w ? k+1 be the solution obtained after running through the kth epoch. We the epoch k, and let

w denote by F(w) and F ? (w) the objective functions Fk (w) and Fk+1 (w). They are given by n 1? ? ? + ? ?w?2 + ??w, w? gi (w + w) (8) F(w) = 2 n i=1 ? ? 1? ? ?) ? ?? + gi (w + w ?w?2 + ?w, w 2? ? n i=1 n

F ? (w)

=

(9)

b? = w b ?k and w b ?? = w b ?k+1 be the optimal solutions that minimize F(w) and F ? (w) over the Let w b ? ? ? ?, our goal is to show domain Wk and Wk+1 , respectively. Under the assumption that ?w ? ??2 b ?? ? ? , F(w ? ?) ? F (w b ?) ? ?w ? 2? 4 b ?) where the proof is deferred to Appendix. The following lemma bounds F(wt) ? F (w Lemma 1.

b ? ?2 b ? ?2 ?wt ? w ?wt+1 ? w ? 2 b ?) ? F(wt) ? F (w ? + ??b git (wt) + ?wt ? + ?g, wt ? wt+1 ? 2? 2? 2 ? ? ? ? b w b w b t), wt ? w b ?) ? ?b b ?), wt ? w b ? + ??b b ?) ? ?F(b ?) + ?F(w b? + ?F(git (w git (wt) + ?b git (w ? 1 = 0, we have By adding the inequality in Lemma 1 over all iterations, using the fact w T ? b ? ?2 b ? ?2 ?w ?wT +1 ? w b ?) ? F(wt) ? F(w ? ? ?g, wT +1 ? 2? 2? t=1 +

T T ? ?? b w b ?) ? ?b b ?), wt ? w b ?? ??b git (wt) + ?wt ?2 + ??F(git (w 2 t=1 t=1 {z } — {z } — :=AT

+

T ? ?

:=BT

? b w b t), wt ? w b ?) ? ?F(b ?) + ?F(w b? . ??b git (wt) + ?b git (w t=1

—

{z

}

:=CT

Since g = ?F(0) and F(wT +1) ? F (0) ? ??F(0), wT +1 ? +

? ? ?wT +1 ?2 = ?g, wT +1 ? + ?wT +1 ?2 2 2 7

using the fact F(0) ? F(w?) + ?2 ?w? ?2 and max(?w? ?, ?wT +1 ?) ? ?, we have ? b ?)) ??g, wT +1 ? ? F(0) ? F (wT +1) + ?2 ? ??2 ? (F(wT +1) ? F(w 2 and therefore ( ) T +1 ? 1 ? 2 b ?) ? ? F(wt) ? F (w + ? + AT + BT + CT . 2? 2 t=1

(10)

The following lemmas bound AT , BT and CT . Lemma 2. For AT defined above we have AT ? 6? 2 ?2 T . The following lemma upper bounds BT and CT . The proof is based on the Bernstein?s inequality for Martingales [4] and is given in the Appendix. Lemma 3. With a probability 1 ? 2?, we have)) ((? ? 1 1 1 1 BT ? ??2 ln + 2T ln and CT ? 2??2 ln + 2T ln . ? ? ? ? Using Lemmas 2 and 3, by substituting the uppers bounds for AT , BT , and CT in (10), with a probability 1 ? 2?, we obtain ( ) ? T +1 ? 1 1 1 2 2 b ?) ? ? F(wt) ? F(w + ? + 6? ?T + 3? ln + 3? 2T ln 2? ? ? t=1 ? By choosing ? = 1/[2? 3T], we have) (? T +1 ? ? 1 1 2 b ?) ? ? 2? 3T + ? + 3? ln + 3? 2T ln F(wt) ? F(w ? ? t=1 ?T +1 e = i=1 wt /(T + 1), we have and using the fact w ? ? 3 ln[1/?] 3 ln[1/?] 2 5? 2 2 2 5? b ? ? e ? F (w b ?) ? ? e ?w b ?? ?

? F(w) , and ? = ?w . T +1 ? T +1 Thus, when T ? [300? 8 ? 2 ln 1?]/?2 , we have, with a probability 1 ? 2?, 2 ? b 2 ? ? , and —F(w) e ? F (w b ?)— ? 4 ?2 . ? (11) 4 ? 2? e ?w b ? ?. b ?? ? to ?w The next lemma relates ?w ? b ? ? ? ??w e ?w b ? ?. Lemma 4. We have ?w b ?? ? ? ?/?. Combining the bound in (11) with Lemma 4, we have ?w

6

Conclusions and Open Questions

We presented a new paradigm of optimization, termed as mixed optimization, that aims to improve the convergence rate of stochastic optimization by making a small number of calls to the full gradient oracle. We proposed the M IXED G RAD algorithm and showed that it is able to achieve an O(1/T) convergence rate by accessing stochastic and full gradient oracles for O(T) and O(log T) times, respectively. We showed that the M IXED G RAD algorithm is able to exploit the smoothness of the function, which is believed to be not very useful in stochastic optimization. In the future, we would like to examine the optimality of our algorithm, namely if it is possible to achieve a better convergence rate for stochastic optimization of smooth functions using O(ln T) accesses to the full gradient oracle. Furthermore, to alleviate the computational cost caused by O(log T) accesses to the full gradient oracle, it would be interesting to empirically evaluate the proposed algorithm in a distributed framework by distributing the individual functions among processors to parallelize the full gradient computation at the beginning of each epoch which requires O(log T) communications between the processors in total. Lastly, it is very interesting to check whether an O(1/T 2) rate could be achieved by an accelerated method in the mixed optimization scenario, and whether linear convergence rates could be achieved in the strongly-convex case. Acknowledgments. The authors would like to thank the anonymous reviewers for their helpful and insightful comments. This work was supported in part by ONR Award N000141210431 and NSF (IIS-1251031).

8

2 References

[1] A. Agarwal, P. L. Bartlett, P. D. Ravikumar, and M. J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. IEEE Transactions on Information Theory, 58(5):3235?3249, 2012. [2] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. Oper. Res. Lett., 31(3):167?175, 2003. [3] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In NIPS, pages 161?168, 2008. [4] S. Boucheron, G. Lugosi, and O. Bousquet. Concentration inequalities. In Advanced Lectures on Machine Learning, pages 208?240, 2003. [5] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, 2004. [6] R. H. Byrd, G. M. Chin, J. Nocedal, and Y. Wu. Sample size selection in optimization methods for machine learning. Mathematical programming, 134(1):127?155, 2012. [7] A. Cotter, O. Shamir, N. Srebro,

and K. Sridharan. Better mini-batch algorithms via accelerated gradient methods. In NIPS, pages 1647?1655, 2011. [8] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao. Optimal distributed online prediction using mini-batches. The Journal of Machine Learning Research, 13:165?202, 2012. [9] M. P. Friedlander and M. Schmidt. Hybrid deterministic-stochastic methods for data fitting. SIAM Journal on Scientific Computing, 34(3):A1380?A1405, 2012. [10] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. Machine Learning, 69(2-3):169?192, 2007. [11] E. Hazan and S. Kale. Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. Journal of Machine Learning Research - Proceedings Track, 19:421?436, 2011. [12] Q. Lin, X. Chen, and J. Pena. A smoothing stochastic gradient method for composite optimization. arXiv preprint arXiv:1008.5204, 2010. [13] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. SIAM J. on Optimization, 19:1574?1609, 2009. [14] A. S. Nemirovsky and D. B. Yudin. Problem complexity and method efficiency in optimization. 1983. [15] Y. Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). In Soviet Mathematics Doklady, volume 27, pages 372?376, 1983. [16] Y. Nesterov. Introductory Lectures on Convex Optimization: A Basic Course. Kluwer Academic Publishers, 2004. [17] Y. Nesterov. Excessive gap technique in nonsmooth convex minimization. SIAM Journal on Optimization, 16(1):235?249, 2005. [18] Y. Nesterov. Smooth minimization of non-smooth functions. Math. Program., 103(1):127? 152, 2005. [19] A. Rakhlin, O. Shamir, and K. Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In ICML, 2012. [20] N. L. Roux, M. W. Schmidt, and F. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In NIPS, pages 2672?2680, 2012. [21] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In ICML, pages 807?814, 2007. [22] S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. JMLR, 14:567599, 2013. [23] O. Shamir and T. Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. ICML, 2013. [24] L. Zhang, T. Yang, R. Jin, and X. He. O(logt) projections for stochastic optimization of smooth and strongly convex functions. ICML, 2013.

9